Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Department of Computer Science & Applications
Minutes of 14th Board of Studies Virtual Meeting

The Department of Computer Science & Applications called for the 14th Board of Studies (BOS) meeting, scheduled to take place on the 31st July, 2026 starting from 10.00A.M onwards.

Members Attended:

1. Dr. K. Bhagavan – HoD, Department of Computer Science and Applications
2. Dr. E. Suresh Babu – Faculty, National Institute of Technology Warangal
3. Smt. Siva Leela – Hexaware, Chennai
4. Dr. V. N. Sailaja – Associate Dean (Academics)
5. Dr. P. Kasiviswanath – Associate Dean (Academics)
6. Dr. Radhika Rani Chintala – HoD, MCA
7. Dr. Y. Rama Krishna – Alternate HoD, BCA
8. Dr. V. S. Bhagavan – Professor, Department of Mathematics
9. Mr. Rajesh Nagineni,Mr. Rajeev Kuduri – CDOE
10. Dr. P. Srinivasa Reddy – HoD, KLH Borampet
11. Dr. D. S. Rao – HoD, KLH Aziz Nagar
12. Dr. K. Kiran Kumar – Professor, Department of BCA
13. Dr. V. Valli Mayil – Professor, Department of BCA
14. Dr. B. Mouleswara Rao – Professor, Department Of MCA
15. Dr. Ch. K. Kiran Kumar – Associate Professor, Department of MCA
16. Dr. V. S. Narayana Tinnaluri – Associate Professor, Department of BCA
17. Dr. Ch. Madhava Rao – Associate Professor, Department of MCA
18. Dr. M. Kalimuthu – Associate Professor, Department of MCA
19. Dr. N. Venkata Ramana – Assistant Professor, Department of MCA.
20. Dr. G. Venkateswarlu – Assistant Professor, Department of MCA
21. Dr. ACP. Ranjani – Assistant Professor, Department of MCA
22. Mrs. T. Hema Latha – Assistant Professor, Department of BCA
23. P. Vasu Dev – II Year BCA (Student Representative)
24. M. Sivanaga Ganesh – III Year BCA (Student Representative)
25. B. Suhasini – II Year MCA (Student Representative)
26. Ujjwal Raj – II Year MCA (Student Representative)

31/07/26.

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Opening Remarks

Dr. K. Bhagavan, Chairperson and HOD-CSA welcomed all attendees and opened the meeting online.

Agenda 1: Review and Approval of Previous Meeting Minutes

Discussion:

- Confirmed accuracy of documented action items from the last meeting.
- Allowed participants to share additional comments or concerns.
- Encouraged suggestions for improving the documentation and approval process.

Recommendations:

- **Dr. E. Suresh Babu** suggested incorporating a tracking mechanism for action items in future meetings.
- **Dr. V.S. Bhagavan** recommended including a summary of key decisions in each meeting minute.

Action Taken:

- The minutes of the previous meeting were reviewed and approved unanimously.

Agenda 2: To consider and approve Key Decisions and Recommendations by the DAC on Y26 BCA Structure.

Discussion:

- Reviewed key decisions and recommendations made by the **Department Academic Committee (DAC)**.
- Evaluated suggestions for curriculum enhancement, faculty development, and industry collaborations.
- Discussed strategies for improving student engagement, research opportunities, and employability.

Recommendations:

- Faculties **Smt T.Hemalatha** suggested to include few AI related courses in the curriculum of BCA from Y26 regulation
- **Smt. Sri Leela**, Industrial peer suggested to include Blockchain and Fintech specialization to give the knowledge on the concepts of block chain and Fintech.
- **Dr. E. Suresh Babu**, Academic Peer and faculties **Dr.Valli Mayil** & Dr. **T.V.S.Narayana** proposed to run BCA in Trimester Mode instead of Regular Model
- Faculty **Smt.Priya Ranjani** and **Dr. Ch. Madhava Rao** suggested to include more hours for skilling and practical to acquire the knowledge
- **Dr. Y.RamaKrishna** and **Dr.B.Krishnaprasad** suggested to introduce Term Paper and Capstone Project from Y26 Regulation.

Action Taken:

- The committee decided to include few AI related courses from Y26 Regulation onwards.

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

- The committee decided to include "Blockchain & Fintech" Specialization from Y26 BCA Regulation students.
- The committee decided to run the BCA program in Trimester mode from Y26 Regulation.
- The committee decided to increase hours in skill and in lab component for all the courses to acquire more knowledge.
- The committee decided to introduce Termpaper and Capstone Project for Y26 BCA admitted students.
- **All the above initiatives are recommended for Academic Council approval [Annexure I]**

Agenda 3: To consider and approve Key Decisions and Recommendations on the Y26 MCA Curriculum.

Discussion:

Need to design and implement a **new MCA curriculum** aligned with industry needs, emerging technologies, and employability skills. Emphasis on industry relevance, technological integration, skill-based learning and flexibility in electives

Recommendations:

- DAC Committee recommended developing the new MCA curriculum entirely from scratch, ensuring industry alignment and academic rigor.
- Faculties **Smt T.Hemalatha** suggested to include few AI related courses in the curriculum of MCA from Y26 regulation
- **Smt. Sri Leela**, Industrial peer suggested to include Blockchain and Fintech specialization to give the knowledge on the concepts of block chain and Fintech.
- **Dr. E. Suresh Babu**, Academic Peer and faculties **Dr.Valli Mayil** & Dr. **T.V.S.Narayana** proposed to run MCA in Trimester Mode instead of Regular Model
- Faculty **Smt.Priya Ranjani** and **Dr. Ch. Madhava Rao** suggested to include more hours for skilling and practical to acquire the knowledge
- **Dr. Y.RamaKrishna** and **Dr.B.Krishnaprasad** suggested to introduce Term Paper and Capstone Project from Y26 Regulation

Action Taken:

- Development of the new MCA curriculum initiated, with faculty task forces assigned specific responsibilities.
- Syllabus and course structures under preparation for submission to the BoS.
- AI-related and foundational courses integrated into the MCA Y26 regulation to ensure smooth academic progression.
- Recommendations on new electives, frameworks, and specialization structure approved for inclusion in the curriculum design.
- **All the above initiatives are recommended for Academic Council approval. [Annexure II]**

31/12/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

ANNEXURE I

Y26 BCA Program Structure

Detailed structure of the program highlighting all the courses and their credits

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
26CA1101E	Java Programming	PCC	3	0	6	0	4	NIL	Newly Introduced Course	Faculty	Skill Development	Builds core object-oriented programming proficiency, a foundational skill sought by employers.
26CA1102E	Mathematics for Computer Applications	BSC	2	1	3	0	3	NIL	Newly Introduced Course	Faculty	Employability	Strengthens the quantitative and analytical reasoning required for computing roles.
26CA1103E	Linux Systems Programming	PCC	3	0	0	6	3	NIL	Newly Introduced Course	Faculty	Skill Development	Provides hands-on systems-level skills valued in DevOps and infrastructure roles.
26UC1108E	English Communication	HAS	2	1	0	0	2	NIL	Newly Introduced Course	Faculty	Employability	Improves workplace communication skills essential for professional readiness.
26CA1204E	Data Structures (Java)	PCC	3	0	6	0	4	JAPR	Newly Introduced Course	Faculty	Employability	Develops problem-solving skills central to technical interviews and

30/7/25
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
												software roles.
26CA 1205E	Python Programming	PCC	3	0	3	0	3	JAPR	Newly Introduced Course	Faculty	Skill Development	Equips students with a versatile scripting language widely used across industry.
26CA 1206E	Frontend Development	PCC	0	0	9	0	3	NIL	Newly Introduced Course	Industry peers	Skill Development	Builds practical UI development skills for full-stack roles.
26UC 1209E	Technical & Professional Communication	HAS	2	1	0	0	2	ECM1	Newly Introduced Course	Faculty	Employability	Enhances technical writing and presentation skills for the workplace.
26CA 1307E	Applied AI & Machine Learning	PCC	3	0	6	0	4	DSJV	Newly Introduced Course	Faculty	Employability	Introduces applied AI/ML techniques in high demand across industry.
26CA 1308E	DBMS & Backend Development	PCC	3	0	6	0	4	PYPR	Newly Introduced Course	Industry peers	Employability	Builds database and backend engineering skills central to software careers.
26CA 1309E	Operating Systems & Concurrent	PC	2	1	3	0	3	LXSP	Newly Introduced	Faculty	Employability	Deepens understanding of system internals and

HoD-CSA 31/7/26
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
	Programming	C							Course			concurrency for systems roles.
26UC0039E	Innovation & Entrepreneurship	HAS	2	1	3	0	3	NIL	Newly Introduced Course	Faculty	Entrepreneurship	Cultivates entrepreneurial thinking and innovation management skills.
26IE2040E	Social Internship	PRI	0	0	6	0	0	NIL	Newly Introduced Course	Faculty	Employability	Provides real-world exposure that builds professional and social competencies.
26UC0009E	Ecology & Environment	HAS	3	0	0	0	0	NIL	Newly Introduced Course	Faculty	Employability	Builds awareness of sustainability considerations relevant to responsible practice.
26CA2110E	Algorithm Design & Analysis	PCC	3	0	6	0	4	DSJV	Newly Introduced Course	Industry peers	Employability	Strengthens algorithmic problem-solving skills valued in technical roles.
26CA2111E	Computer Networks & Applications	PCC	3	0	6	0	4	OSCP	Newly Introduced Course	Faculty	Employability	Builds networking fundamentals required for infrastructure and security roles.

31/7/24
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K L Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
26CA 2112E	Applied Networks & Virtualization	PCC	3	0	3	0	3	CNAP	Newly Introduced Course	Faculty	Skill Development	Provides hands-on virtualization skills relevant to cloud and IT operations.
FLP	Foreign Language Proficiency	HAS	3	0	0	0	2	NIL	Newly Introduced Course	Faculty	Employability	Expands global communication skills for international career opportunities.
26CA 2213E	Advanced Data Structures & Algorithms	PCC	3	0	3	0	3	ALDA	Newly Introduced Course	Faculty	Employability	Builds advanced problem-solving skills for competitive technical roles.
26CA 2214E	Cloud Computing & DevOps	PCC	3	0	6	0	4	CNAP	Newly Introduced Course	Industry peers	Skill Development	Develops practical cloud and DevOps skills in high industry demand.
26CA 2215E	Deep Learning Foundations	PCC	3	0	3	0	3	AAML	Newly Introduced Course	Faculty	Employability	Introduces foundational deep learning concepts applicable across AI roles.
26CA 2216E	Cybersecurity Foundation	PCC	3	0	3	0	3	CNAP	Newly Introduced Course	Faculty	Employability	Builds foundational cybersecurity skills addressing growing

31/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
	s											industry demand.
26CA2317E	Generative AI Foundations	PCC	3	0	3	0	3	DLFO	Newly Introduced Course	Industry peers	Employability	Builds knowledge of generative AI techniques increasingly used across industry.
26CA2318E	Machine Learning Operations (MLOps)	PCC	3	0	6	0	4	DLFO	Newly Introduced Course	Faculty	Skill Development	Develops practical MLOps skills for deploying and maintaining ML systems.
26CA2319E	AI Platform Engineering	PCC	3	0	3	0	3	CCDO	Newly Introduced Course	Industry peers	Skill Development	Builds platform engineering skills for scaling AI systems in production.
26CA2320E	Software Engineering & Agile	PCC	3	0	3	0	3	NIL	Newly Introduced Course	Faculty	Employability	Builds software engineering and agile project management skills.
26IE3041E	Technical Internship	PRI	0	0	6	0	0	SICE	Newly Introduced Course	Faculty	Employability	Provides industry exposure that builds practical, job-ready skills.
26UC0032E	Insights to Sustainable Developme	HA	3	0	0	0	0	NIL	Newly Introduced	Faculty	Employability	Builds awareness of sustainability principles relevant

31/1/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
	nt Goals	S							Course			to responsible professional practice.
26CA3121E	High Performance Computing	PCC	3	0	0	6	3	CCDO	Newly Introduced Course	Academic Peers	Employability	Builds specialized computing skills for performance-critical applications.
26CA3122E	Offensive Security (Red Team)	PCC	3	0	3	0	3	CYBF	Newly Introduced Course	Faculty	Employability	Builds offensive security skills addressing growing demand for security professionals.
26CA3123E	Agentic AI & Applications	PCC	3	0	3	0	3	GAIF	Newly Introduced Course	Academic Peers	Employability	Builds knowledge of agentic AI systems, an emerging and in-demand skill area.
26CA3224E	Defensive Security (SOC, SIEM, DFIR)	PCC	3	0	6	0	4	OFSE	Newly Introduced Course	Academic Peers	Employability	Builds defensive security operations skills in high industry demand.
26IE3201E	Term Paper	PRI	0	0	9	0	3	NIL	Newly Introduced Course	Faculty	Employability	Builds independent research and technical writing skills.

3/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
26CA3325E	DevSecOps	PCC	2	1	3	0	3	DFSE	Newly Introduced Course	Faculty	Skill Development	Builds practical DevSecOps skills integrating security into delivery pipelines.
26IE3030E	Capstone Project	PRI	0	0	18	0	6	TRPO	Newly Introduced Course	Faculty	Employability	Builds the ability to apply learning to solve real-world problems end-to-end.
List of Foreign Language Electives												
26FL1001E	German Language Proficiency	HAS	3	0	0	0	2	NIL	Newly Introduced Course	Faculty	Employability	Enhances employability by developing German language proficiency for global career opportunities.
26FL1002E	Japanese Language Proficiency	HAS	3	0	0	0	2	NIL	Newly Introduced Course	Faculty	Employability	Enhances employability by developing Japanese language proficiency for global career opportunities.
26FL1003E	Korean Language	HAS	3	0	0	0	2	NIL	Newly Introduced	Faculty	Employability	Enhances employability by developing Korean

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
	Proficiency								Course			language proficiency for global career opportunities.
List of Professional Electives												
Stream A: Applied AI &GenAI												
26CA31A1E	Conversational AI & Speech Systems	PEC	3	0	3	0	3	NIL	Newly Introduced Course	Academic Peers	Employability	Builds specialized skills in conversational AI, increasingly used in industry applications.
26CA32A2E	Computer Vision & Visual Intelligence	PEC	3	0	3	0	3	NIL	Newly Introduced Course	Faculty	Employability	Builds computer vision skills applicable to AI-driven products.
26CA32A3E	Deep Learning Architectures	PEC	3	0	3	0	3	NIL	Newly Introduced Course	Faculty	Employability	Deepens expertise in advanced deep learning architectures.
26CA33A4E	Natural Langua	PEC	3	0	3	0	3	NIL	Newly Introduced	Faculty	Employability	Builds NLP and LLM skills central

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
	ge Processing & LLM Foundations								Course			to current AI industry roles.
Stream B: Cloud Native & SRE												
26CA31B1E	Cloud-Native Infrastructure &GitOps	PEC	3	0	3	0	3	NIL	Newly Introduced Course	Faculty	Skill Development	Builds hands-on cloud-native infrastructure and GitOps skills.
26CA32B2E	Event-Driven & Stream Processing Systems	PEC	3	0	3	0	3	CNIG	Newly Introduced Course	Academic Peers	Skill Development	Builds skills in event-driven architectures used in scalable systems.
26CA32B3E	Site Reliability Engineering & Observability	PEC	3	0	3	0	3	CNIG	Newly Introduced Course	Faculty	Employability	Builds SRE skills in high demand for maintaining production systems.

31/4/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
26CA33B4E	Serverless& Edge Computing	PEC	3	0	3	0	3	CNIG	Newly Introduced Course	Faculty	Skill Development	Builds serverless and edge computing skills relevant to modern cloud architectures.
Stream C: Applied Cybersecurity												
26CA31C1E	Web & API Security	PEC	3	0	3	0	3	NIL	Newly Introduced Course	Faculty	Employability	Builds web and API security skills addressing common attack surfaces.
26CA32C2E	Privacy Engineering & Data Protection	PEC	3	0	3	0	3	WAPS	Newly Introduced Course	Faculty	Employability	Builds data protection skills increasingly required by regulation and industry.
26CA32C3E	Cloud Security & Zero Trust	PEC	3	0	3	0	3	WAPS	Newly Introduced Course	Academic Peers	Employability	Builds cloud security skills aligned with modern zero-trust architectures.
26CA33C4E	Security Governance, Risk & Compliance	PEC	3	0	3	0	3	WAPS	Newly Introduced Course	Faculty	Entrepreneurship	Builds governance and compliance knowledge valuable for security leadership roles.

3/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
Stream D: Blockchain& Fintech												
26CA31D1E	Smart Contract Development & Security	PEC	3	0	3	0	3	NIL	Newly Introduced Course	Faculty	Employability	Builds smart contract development skills for blockchain careers.
26CA32D2E	Decentralized Finance &Tokenomics	PEC	3	0	3	0	3	SCDS	Newly Introduced Course	Academic Peers	Entrepreneurship	Builds knowledge of decentralized finance relevant to fintech innovation.
26CA32D3E	Financial Markets & Quantitative Methods	PEC	3	0	3	0	3	SCDS	Newly Introduced Course	Faculty	Employability	Builds quantitative finance skills applicable to fintech roles.
26CA33D4E	RegTech& Compliance Automation	PEC	3	0	3	0	3	SCDS	Newly Introduced Course	Faculty	Employability	Builds regulatory technology skills addressing compliance automation needs.

31/7/24
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
Stream E: Data Science												
26CA31E1E	Data Engineering at Scale	PEC	3	0	3	0	3	NIL	Newly Introduced Course	Faculty	Skill Development	Builds data engineering skills for handling large-scale data systems.
26CA32E2E	Customer & Growth Analytics	PEC	3	0	3	0	3	DEAS	Newly Introduced Course	Faculty	Employability	Builds analytics skills applicable to business growth functions.
26CA32E3E	Big Data & Distributed Analytics	PEC	3	0	3	0	3	DEAS	Newly Introduced Course	Faculty	Skill Development	Builds big data skills for distributed analytics environments.
26CA33E4E	Experimentation & Causal Analytics	PEC	3	0	3	0	3	DEAS	Newly Introduced Course	Faculty	Employability	Builds experimentation and causal inference skills for data-driven decision making.
List of Courses for Honors Degree												
26IE4060E	Research Method	HON	3	0	3	0	3	NIL	Newly Introduced Course	Faculty	Employability	Builds research methodology skills for advanced

3/1/26
HoD-CSA

Dr. K. SRAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Course Code	Course Name	Course Category	L	T	P	S	CR	Pre-Requisite	New/Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
	s											academic and industry work.
26IE4061E	Industry Immersion-I	HON	0	0	27	12	11	NIL	Newly Introduced Course	Faculty	Employability	Provides extended industry exposure building practical, job-ready skills.
26IE4062E	Industry Capstone-II / Honors Research Thesis	HON	0	0	39	6	14	NIL	Newly Introduced Course	Faculty	Employability	Builds the ability to conduct independent, in-depth technical research or projects.

Percentage of Syllabus Revision = **100% (62 out of 62)**
Percentage of Courses focusing on Employability= **72.5% (45 out of 62)**
Percentage of Courses focusing on Entrepreneurship= **4.83% (3 out of 62)**
Percentage of Courses focusing on Skill Development = **22.5% (14 out of 62)**

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2578129

Bachelor of Computer Applications

Y26 Batch

(DEEMED TO BE UNIVERSITY)

A.Y. 2026-27 (I Year)

Syllabus

HoD-CSA 31/7/26

DEPUTY REGISTRAR
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Y26 BCA - Trimester (PBL) Structure												
Sl No	TRI	CAT	COURSE CODE	COURSE TITLE	ACRO	L	T	P	S	Cr	CH	PRE-REQ
1	T01	PCC	26CA1101E	Java Programming	JAPR	3	0	6	0	4	9	NIL
2	T01	BSC	26CA1102E	Mathematics for Computer Applications	MACA	2	1	3	0	3	6	NIL
3	T01	PCC	26CA1103E	Linux Systems Programming	LXSP	3	0	0	6	3	9	NIL
4	T01	HAS	26UC1108E	English Communication	ECM1	2	1	0	0	2	3	NIL
5	T02	PCC	26CA1204E	Data Structures (Java)	DSJV	3	0	6	0	4	9	JAPR
6	T02	PCC	26CA1205E	Python Programming	PYPR	3	0	3	0	3	6	JAPR
7	T02	PCC	26CA1206E	Frontend Development	FRDE	0	0	9	0	3	9	NIL
8	T02	HAS	26UC1209E	Technical & Professional Communication	TPCO	2	1	0	0	2	3	ECM1
9	T03	PCC	26CA1307E	Applied AI & Machine Learning	AAML	3	0	6	0	4	9	DSJV
10	T03	PCC	26CA1308E	DBMS & Backend Development	DBBD	3	0	6	0	4	9	PYPR
11	T03	PCC	26CA1309E	Operating Systems & Concurrent Programming	OSCP	2	1	3	0	3	6	LXSP
12	T03	HAS	26UC0039E	Innovation & Entrepreneurship	INEN	2	1	3	0	3	6	NIL
13	1S	PR1	26IE2040E	Social Internship	SICE	0	0	6	0	0	6	NIL
14	1S	HAS	26UC0009E	Ecology & Environment	ECEV	3	0	0	0	0	3	NIL
15	T04	PCC	26CA2110E	Algorithm Design & Analysis	ALDA	3	0	6	0	4	9	DSJV
16	T04	PCC	26CA2111E	Computer Networks &	CNAP	3	0	6	0	4	9	OSCP

31/7/24
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

				Applications								
17	T04	PCC	26CA2112E	Applied Networks & Virtualization	ANVT	3	0	3	0	3	6	CNAP
18	T04	HAS	FLP	Foreign Language Proficiency	FOLP	3	0	0	0	2	3	NIL
19	T05	PCC	26CA2213E	Advanced Data Structures & Algorithms	ADSA	3	0	3	0	3	6	ALDA
20	T05	PCC	26CA2214E	Cloud Computing & DevOps	CCDO	3	0	6	0	4	9	CNAP
21	T05	PCC	26CA2215E	Deep Learning Foundations	DLFO	3	0	3	0	3	6	AAML
22	T05	PCC	26CA2216E	Cybersecurity Foundations	CYBF	3	0	3	0	3	6	CNAP
23	T06	PCC	26CA2317E	Generative AI Foundations	GAIF	3	0	3	0	3	6	DLFO
24	T06	PCC	26CA2318E	Machine Learning Operations (MLOps)	MLOP	3	0	6	0	4	9	DLFO
25	T06	PCC	26CA2319E	AI Platform Engineering	AIPE	3	0	3	0	3	6	CCDO
26	T06	PCC	26CA2320E	Software Engineering & Agile	SEAG	3	0	3	0	3	6	NIL
27	2S	PRI	26IE3041E	Technical Internship	TIIP	0	0	6	0	0	6	SICE
28	2S	HAS	26UC0032E	Insights to Sustainable Development Goals	SDGL	3	0	0	0	0	3	NIL
29	T07	PCC	26CA3121E	High Performance Computing	HPCO	3	0	0	6	3	9	CCDO
30	T07	PCC	26CA3122E	Offensive Security (Red Team)	OFSE	3	0	3	0	3	6	CYBF
31	T07	PCC	26CA3123E	Agentic AI & Applications	AGAI	3	0	3	0	3	6	GAIF
32	T07	PEC	PE-1	Professional Elective - 1	PEC1	3	0	3	0	3	6	NIL
33	T08	PCC	26CA3224E	Defensive Security (SOC, SIEM, DFIR)	DFSE	3	0	6	0	4	9	OFSE

31/1/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

34	T08	PEC	PE-2	Professional Elective - 2	PEC2	3	0	3	0	3	6	PEC1
35	T08	PEC	PE-3	Professional Elective - 3	PEC3	3	0	3	0	3	6	PEC1
36	T08	OEC	OE-1	Open Elective - 1	OEC1	3	0	3	0	3	6	NIL
37	T08	PRI	26IE3201E	Term Paper	TRPO	0	0	9	0	3	9	NIL
38	T09	PCC	26CA3325E	DevSecOps	DSOP	2	1	3	0	3	6	DFSE
39	T09	PEC	PE-4	Professional Elective - 4	PEC4	3	0	3	0	3	6	PEC1
40	T09	OEC	OE-2	Open Elective - 2	OEC2	3	0	3	0	3	6	OE01
41	T09	PRI	26IE3030E	Capstone Project	CPST	0	0	18	0	6	18	TRPO
				Total Credits						**120**		

List of Foreign Language Electives												
1		HAS	26FL1001E	German Language Proficiency	GLP	3	0	0	0	2	3	
2		HAS	26FL1002E	Japanese Language Proficiency	JLP	3	0	0	0	2	3	
3		HAS	26FL1003E	Korean Language Proficiency	KLP	3	0	0	0	2	3	

List of Professional Electives												
Sl No	**TRI**	**CAT**	**COURSE CODE**	**COURSE TITLE**	**ACRO**	**L**	**T**	**P**	**S**	**Cr**	**CH**	**PRE-REQ**
Stream A: Applied AI & GenAI												
1	T07	PEC	26CA31A1E	Conversational AI & Speech Systems	CASS	3	0	3	0	3	6	NIL
2	T08	PEC	26CA32A2E	Computer Vision & Visual Intelligence	CVVI	3	0	3	0	3	6	NIL
3	T08	PEC	26CA32A3E	Deep Learning Architectures	DLA	3	0	3	0	3	6	NIL
4	T09	PEC	26CA33A4E	Natural Language Processing & LLM Foundations	NLPLF	3	0	3	0	3	6	NIL
Stream B: Cloud Native & SRE												
5	T07	PEC	26CA31B1E	Cloud-Native Infrastructure &GitOps	CNIG	3	0	3	0	3	6	NIL
6	T08	PEC	26CA32B2E	Event-Driven & Stream	EDSP	3	0	3	0	3	6	CNIG

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

				Processing Systems								
7	T08	PEC	26CA32B3E	Site Reliability Engineering & Observability	SREO	3	0	3	0	3	6	CNIG
8	T09	PEC	26CA33B4E	Serverless & Edge Computing	SECG	3	0	3	0	3	6	CNIG
Stream C: Applied Cybersecurity												
9	T07	PEC	26CA31C1E	Web & API Security	WAPS	3	0	3	0	3	6	NIL
10	T08	PEC	26CA32C2E	Privacy Engineering & Data Protection	PEDP	3	0	3	0	3	6	WAPS
11	T08	PEC	26CA32C3E	Cloud Security & Zero Trust Architecture	CSZT	3	0	3	0	3	6	WAPS
12	T09	PEC	26CA33C4E	Security Governance, Risk & Compliance	SGRC	3	0	3	0	3	6	WAPS
Stream D: Blockchain & Fintech												
13	T07	PEC	26CA31D1E	Smart Contract Development & Security	SCDS	3	0	3	0	3	6	NIL
14	T08	PEC	26CA32D2E	Decentralized Finance &Tokenomics	DFTK	3	0	3	0	3	6	SCDS
15	T08	PEC	26CA32D3E	Fintech Systems Engineering	FSEG	3	0	3	0	3	6	SCDS
16	T09	PEC	26CA33D4E	RegTech& Compliance Automation	RTCA	3	0	3	0	3	6	SCDS
Stream E: Data Science												
17	T07	PEC	26CA31E1E	Data Engineering at Scale	DEAS	3	0	3	0	3	6	NIL
18	T08	PEC	26CA32E2E	Customer & Growth Analytics	CAGA	3	0	3	0	3	6	DEAS
19	T08	PEC	26CA32E3E	Business Forecasting & Demand Analytics	BFDA	3	0	3	0	3	6	DEAS
20	T09	PEC	26CA33E4E	Experimentation & Causal Analytics	EACA	3	0	3	0	3	6	DEAS

31/7/26

HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

COURSE CODE: 26CA1101E
COURSE NAME :JAVA PROGRAMMING
L-T-P-S : 3-0-6-0 **PRE-REQ : NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply object-oriented programming principles — encapsulation, inheritance, polymorphism, abstraction — to design Java classes and class hierarchies for application-domain entities.	PO1,PO3	3
CO2	Apply the Java Collections Framework (List, Map, Set, Queue), generics, and File I/O APIs to build type-safe data manipulation modules and persistence layers in real applications.	PO1,PO5	3
CO3	Analyze concurrent execution scenarios using threads, ExecutorService, and synchronization primitives to identify race conditions and design correct multi-threaded application components.	PO2,PO4	4
CO4	Analyze JVM internals (memory model, class loading, garbage collection) and exception handling pathways to diagnose runtime issues, memory leaks, and reliability gaps in deployed Java applications.	PO2,PO4	4
CO5	Evaluate code quality, design pattern choices, and modular architecture decisions to refactor Java applications using modern features (Stream API, lambdas, sealed classes) for maintainability and extensibility.	PO3,PO5	5
CO6	Evaluate the effectiveness of build, test, and version-control workflows (Maven, JUnit 5, Git, GitHub Actions) integrated with a complete project deliverable, including AI-assisted coding tools (GitHub Copilot).	PO5,PO10	5

Syllabus:

Module No.	Syllabus
M1	**Object-Oriented Foundations & Class Design** Mapped: **CO1 (BTL-3)** · 12 hrs (8L + 4P) · Project Stage 1 — Domain Model • Java syntax essentials: primitives, control flow, arrays, strings

HoD-CSA 21/7/26

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• Class as a first-class abstraction: fields, methods, constructors, access modifiers • Encapsulation in practice: getter/setter discipline, immutable objects • Inheritance & polymorphism: method override, dynamic dispatch • Abstraction: abstract classes vs interfaces; when to use each • Packages, imports, project structure (Maven src/main/java) • equals(), hashCode(), toString() — the contract & why it matters • Java 17+ features at the foundation: records, sealed types, var *Concepts are introduced through the lens of modeling the project domain. Students design User, Asset (abstract), Stock, and MutualFund classes by Module 1 end. Lectures alternate with hands-on coding using GitHub Copilot suggestions reviewed for correctness.*
M2	**Collections, Generics & I/O** Mapped: **CO2 (BTL-3)** · 12 hrs (8L + 4P) · Project Stage 2 — Data Layer • Collections Framework hierarchy: List, Set, Map, Queue, Deque • Implementations: ArrayList vs LinkedList, HashMap vs TreeMap, HashSet vs LinkedHashSet • Generics: type parameters, bounded wildcards (? extends, ? super), type erasure • Comparable & Comparator: sorting custom objects • File I/O: java.io vs java.nio.file • JSON parsing with Jackson; Properties files *Students extend the project to maintain a Portfolio using HashMap, persist it to/from JSON files, and demonstrate complexity awareness. Library integration introduces dependency management.*
M3	**Concurrency &Multi-threading** Mapped: **CO3 (BTL-4)** · 12 hrs (8L + 4P) · Project Stage 3 — Concurrent Price Engine • Thread fundamentals: Thread, Runnable, Callable, lifecycle, daemon threads • ExecutorService and thread pools (newFixedThreadPool, ForkJoinPool) • Synchronization: synchronized blocks, intrinsic locks, ReentrantLock • Race conditions, deadlock, livelock, starvation — diagnosis & prevention • Concurrent collections: ConcurrentHashMap, CopyOnWriteArrayList, BlockingQueue • Atomic variables: AtomicInteger, AtomicReference, CAS operations • CompletableFuture for asynchronous composition; Virtual threads (Java 21) *The project gains a price-feed simulator that updates prices concurrently. Students must identify and fix race conditions when multiple threads update the same Holding. Performance is measured before and after concurrent refactoring.*

31/1/26

HoD-CSA

D. [illegible]AGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

M4	**JVM Internals, Exception Handling & AI Tooling** Mapped: **CO4 (BTL-4)** · 12 hrs (8L + 4P) · Project Stage 4 — Reliability Hardening • JVM architecture: classloaders, runtime data areas (heap, stack, metaspace) • Memory model: object lifetime, references (strong, weak, soft, phantom) • Garbage collection: G1, ZGC, Shenandoah — algorithmic intuition • Profiling tools: VisualVM, JFR, JConsole, jstack, jmap • Exception hierarchy: checked vs unchecked, custom exceptions, try-with-resources • Logging: SLF4J + Logback configuration • AI coding tools: GitHub Copilot prompt patterns, Claude Code for review *Students profile the project under load, identify a deliberately seeded memory leak, and fix it. AI tools are introduced as productivity multipliers — students learn when to trust suggestions and when to override them.*
M5	**Modern Java, Functional Patterns & Design Refactoring** Mapped: **CO5 (BTL-5)** · 12 hrs (8L + 4P) · Project Stage 5 — Architecture Refinement • Lambda expressions and functional interfaces (Function, Predicate, Consumer, Supplier) • Stream API: map, filter, reduce, collect; lazy evaluation; parallel streams • Optional: avoiding null, idiomatic chaining; pattern matching for instanceof and switch • Modern data: records, sealed interfaces for ADT-like modeling • GoF design patterns: Singleton, Factory, Strategy, Observer, Builder, Decorator • SOLID principles, clean code conventions, code smells *Students refactor the project to use Strategy (different valuation methods), Observer (price-change notifications), Builder (Trade construction), and the Stream API for portfolio analytics. Code quality is reviewed through peer code-review sessions.*
M6	**Build, Test, Version Control & Deployment** Mapped: **CO6 (BTL-5)** · 12 hrs (6L + 6P) · Project Stage 6 — Production Readiness • Maven: pom.xml structure, dependencies, plugins, build lifecycle, profiles • Unit testing with JUnit 5: assertions, parameterized tests, lifecycle • Mocking with Mockito; Test coverage with JaCoCo (target ≥ 70%) • Git workflows: branching strategies, pull requests, code review • GitHub Actions CI: build → test → coverage on every push • Packaging: executable JAR, fat-JAR with Maven Shade, Docker basics *The project is fully Mavenized. Students write 30+ unit tests, push to GitHub with a clean commit history, configure GitHub Actions for CI, and produce a deployable JAR. The course closes with a 5-minute demo + Q&A per student.*

3/7/26
HoD-CSA

Dr. K. Srujan Raju
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	*Core Java, Volume I — Fundamentals*	Cay S. Horstmann	Pearson	12th Ed, 2022	978-0137673629
2	*Effective Java*	Joshua Bloch	Addison-Wesley	3rd Ed, 2018	978-0134685991
3	*Head First Java*	Sierra, Bates & Gee	O'Reilly Media	3rd Ed, 2022	978-1491910771
4	*Java Concurrency in Practice*	Brian Goetz et al.	Addison-Wesley	1st Ed, 2006	978-0321349606
5	*Clean Code*	Robert C. Martin	Prentice Hall	1st Ed, 2008	978-0132350884
6	*Java — The Complete Reference*	Herbert Schildt	McGraw Hill	12th Ed, 2021	978-1260463415
7	*Modern Java in Action*	Urma, Fusco, Mycroft	Manning	1st Ed, 2018	978-1617293566

COURSE CODE: 26CA1102E
COURSE NAME :MATHEMATICS FOR COMPUTER APPLICATIONS
L-T-P-S : 3-0-6-0 **PRE-REQ : NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply discrete mathematics — sets, logic, relations, functions — to model computational problems involving structured reasoning.	PO1,PO2	3
CO2	Apply linear algebra — vectors, matrices, transformations, eigenvalues — using NumPy to solve systems and analyze data	PO1,PO5	3

31/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	geometrically.		
CO3	Analyze derivative-based optimization (gradient descent) and integration to understand the math underlying machine learning training.	PO2, PO4	4
CO4	Analyze probability distributions, expectation, variance, and Bayes' theorem to reason about uncertainty in data and predictions.	PO2, PO4	4
CO5	Evaluate combinatorial counting and graph-theoretic constructions to model dependencies, networks, and computational complexity.	PO2, PO3	5
CO6	Evaluate numerical methods for solving equations, integration, and optimization to choose appropriate techniques for engineering problems.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Discrete Mathematics & Logic** Mapped: **CO1 (BTL-3)** · 8 hrs · Project Stage 1 — Logic notebook • Sets, set operations, Cartesian products; sets in Python • Propositional logic: AND, OR, NOT, XOR, implication, equivalence • Predicate logic and quantifiers (∀, ∃) • Truth tables and tautologies; SAT solver intro • Relations: equivalence, partial order • Functions: injective, surjective, bijective • Boolean algebra and digital circuits • Proof techniques: direct, contradiction, induction *Concepts implemented as Python set operations and SymPy logical formulas. Every theorem proved is also verified by code.*
M2	**Linear Algebra for ML** Mapped: **CO2 (BTL-3)** · 12 hrs · Project Stage 2 — Linear algebra notebook • Vectors: addition, scalar multiplication, dot product, magnitude, projection • Vector spaces, linear independence, span, basis (geometric intuition) • Matrices: addition, multiplication, transpose, identity, inverse • Systems of linear equations: Ax = b; Gaussian elimination • Determinants and geometric meaning (volume scaling) • Eigenvalues and eigenvectors: computation and intuition

HoD-CSA 31/7/26

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• Linear transformations: rotation, scaling, projection • Singular Value Decomposition (SVD) — preview, used in ML *Every concept is implemented in NumPy and visualized. Students compute eigenvectors of small matrices by hand and verify with NumPy.*
M3	**Calculus & Optimization** Mapped: **CO3 (BTL-4)** · 10 hrs · Project Stage 3 — Calculus notebook • Limits and continuity (intuitive) • Derivatives: rules, chain rule, product rule, quotient rule • Partial derivatives and gradients • Geometric meaning of gradient (steepest ascent) • Gradient descent algorithm and variants (intro) • Integration: definite and indefinite, fundamental theorem • Numerical integration: trapezoidal, Simpson's rule • Convex functions and why ML loves them *Students implement gradient descent from scratch on a 2-variable function and visualize the descent path.*
M4	**Probability & Statistics** Mapped: **CO4 (BTL-4)** · 12 hrs · Project Stage 4 — Probability notebook • Sample space, events, probability axioms • Conditional probability and Bayes' theorem • Random variables: discrete and continuous • Distributions: Bernoulli, Binomial, Poisson, Normal, Uniform • Expectation, variance, standard deviation; LOTUS • Joint, marginal, conditional distributions • Central Limit Theorem (intuition + simulation) • Hypothesis testing basics; p-values *Each distribution is sampled in Python and the empirical histogram converges to the theoretical PDF. CLT shown by simulation, not just stated.*
M5	**Combinatorics & Graph Theory** Mapped: **CO5 (BTL-5)** · 8 hrs · Project Stage 5 — Combinatorics notebook • Counting: permutations, combinations, multinomial • Pigeonhole principle and applications • Inclusion-exclusion principle • Generating functions (intro) • Graphs: vertices, edges, directed/undirected, weighted • Graph properties: connectivity, cycles, planarity • Trees as a special class of graphs • Big graphs in practice: social networks, web, knowledge graphs *Cross-link with DSTR: same graphs PathFinder uses are analyzed mathematically*

3/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K L Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	here. Theory-practice connection direct.
M6	**Numerical Methods** Mapped: **CO6 (BTL-5)** · 10 hrs · Project Stage 6 — Numerical methods notebook • Floating-point arithmetic: precision, error, why $0.1 + 0.2 \neq 0.3$ • Root-finding: bisection, Newton-Raphson, secant • Solving ODEs (numerically): Euler, Runge-Kutta (intro) • Numerical linear algebra: LU decomposition, QR (computational view) • Optimization: gradient descent variants (SGD, momentum, Adam — intro) • Monte Carlo methods (intro) • When numerical methods fail and how to detect it • scipy.optimize and scipy.integrate as production tools *Students implement bisection and Newton's method from scratch, then compare with scipy. Learn to trust black-box libraries — and to know when not to.*

Reference Books:

Sno	Title	Author(s)	Publisher	Edition / Year	ISBN
1	Mathematics for Machine Learning	Deisenroth, Faisal, Ong	Cambridge	1st Ed, 2020	978-1108455145
2	Discrete Mathematics and Its Applications	Kenneth H. Rosen	McGraw Hill	8th Ed, 2018	978-1259676512
3	Introduction to Linear Algebra	Gilbert Strang	Wellesley-Cambridge Press	5th Ed, 2016	978-0980232776
4	A First Course in Probability	Sheldon Ross	Pearson	10th Ed, 2018	978-9353433062
5	Numerical Methods for Engineers	Chapra, Canale	McGraw Hill	8th Ed, 2020	978-1260232073
6	Calculus: Early Transcendentals	James Stewart	Cengage	9th Ed, 2020	978-1337613927

3/7/24
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

7	Think Stats	Allen B. Downey	O'Reilly	2nd Ed. 2014	978-1491907337

COURSE CODE: 26CA1103E
COURSE NAME :LINUX SYSTEMS PROGRAMMING
L-T-P-S : 3-0-0-6

PRE-REQ : NIL

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply Linux command-line tools — file manipulation, text processing, pipes, redirection — to perform common system administration and developer tasks fluently.	PO1, PO5	3
CO2	Apply Bash scripting (variables, conditionals, loops, functions, parameter handling) to automate routine workflows and build reusable command-line tools.	PO3, PO5	3
CO3	Analyze process management — fork, exec, signals, job control — to design programs that interact correctly with the operating system.	PO2, PO4	4
CO4	Analyze file descriptors, system calls, and the C-level interface to OS services to write low-level system programs.	PO2, PO4	4
CO5	Evaluate inter-process communication mechanisms (pipes, FIFOs, sockets) and choose appropriate IPC for specific application architectures.	PO3, PO5	5
CO6	Evaluate Linux-based DevOps fundamentals — package management, services, networking, security — to provision and maintain a working Linux server.	PO5, PO10	5

Syllabus:

Module No.	Syllabus
M1	**Linux Fundamentals & Filesystem** Mapped: **CO1 (BTL-3)** · 10 hrs · Project Stage 1 — Dev environment • Linux philosophy: everything is a file; small tools that compose • Filesystem layout: /etc, /var, /home, /usr, /tmp, /proc • File permissions: chmod, chown, chgrp; SUID/SGID/sticky bit

3/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• Text processing: grep, sed, awk, sort, uniq, cut, head, tail • Pipes, redirection, command substitution • find and locate for filesystem search • Compression and archives: tar, gzip, bzip2, zip • Vim or Nano fluency *Students set up their personal Linux dev box (Ubuntu in WSL2 or VirtualBox) in Week 1. All subsequent work happens on it.*
M2	**Bash Scripting** Mapped: **CO2 (BTL-3)** · 10 hrs · Project Stage 2 — CLI tool • Shell variables, environment variables, parameter expansion • Quoting: single, double, backtick • Conditionals: if-else, test, [[ ]], case • Loops: for, while, until; break and continue • Functions and recursive functions in Bash • Parameter handling: $1, $@, getopts • Exit codes and error handling: set -euopipefail • Bash debugging: set -x, shellcheck *Students start building their personal CLI tool — a productivity helper they actually want to use (note-taker, todo, time-tracker).*
M3	**Process Management & Signals** Mapped: **CO3 (BTL-4)** · 10 hrs · Project Stage 3 — Multi-process tool • Process model: PID, PPID, parent-child relationships • ps, top, htop, kill, killall, nice, renice • Job control: &, fg, bg, jobs, nohup, disown, screen, tmux • Signals: SIGTERM, SIGKILL, SIGINT, SIGHUP, SIGCHLD • Cron and at for scheduling • Systemd basics: services, units, timers • fork() and exec() at C level (preview, deep in M4) • Resource limits: ulimit, /etc/security/limits.conf *The CLI tool gains a daemon mode that runs as a background service and handles signals gracefully.*
M4	**System Programming in C** Mapped: **CO4 (BTL-4)** · 10 hrs · Project Stage 4 — C extension • Why systems programming matters; Bash limits • C basics for systems: pointers, structs, function pointers (rapid review) • File descriptors: open, read, write, close • fork(), exec(), wait() — the process-creation triad • Standard library vs system calls: when each is right • errno, perror, strerror • Memory: malloc/free; valgrind for leak detection

21/7/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• Compilation: gcc, make, basic Makefiles *Students write a C program that does something Bash cannot do efficiently and integrate it into their CLI tool.*
M5	**IPC & Networking Basics** Mapped: **CO5 (BTL-5)** · 10 hrs · Project Stage 5 — IPC integration • Pipes: anonymous and named (FIFO) • Shared memory and semaphores (System V intro) • Message queues • Sockets: TCP and UDP basics; socket(), bind(), listen(), accept() • A simple client-server in C • netstat, ss, lsof, tcpdump for network diagnostics • SSH: key-based authentication, port forwarding, tunneling • Network namespaces (intro to container internals) *Students extend the CLI tool to communicate over a Unix socket — separating daemon from client.*
M6	**DevOps Foundations & Server Administration** Mapped: **CO6 (BTL-5)** · 10 hrs · Project Stage 6 — Deployed • Package management: apt (Debian/Ubuntu), dnf (RHEL/Fedora) • Service management with systemd (deep dive) • Log management: journalctl, log rotation, /var/log • Networking config: /etc/hosts, DNS resolution, iptables/nftables intro • Firewall basics: ufw, ports • SSH server hardening: keys-only, fail2ban, port change • Cloud server provisioning: AWS EC2 free tier • Monitoring: top, sar, vmstat, iotop; healthcheck script *Students provision a real Linux server (AWS EC2 free tier) and deploy their CLI tool as a systemd service. Tool runs 24/7 on a real internet-accessible server.*

Reference Books:

Sno	Title	Author(s)	Publisher	Edition / Year	ISBN
1	The Linux Programming Interface	Michael Kerrisk	No Starch Press	1st Ed, 2010	978-1593272203
2	Linux System Programming	Robert Love	O'Reilly	2nd Ed, 2013	978-1449339531

31/3/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

3	Advanced Bash-Scripting Guide	Mendel Cooper	Linux Documentation Project (free)	2014	—
4	Linux Command Line and Shell Scripting Bible	Blum, Bresnahan	Wiley	4th Ed, 2020	978-1119700913
5	How Linux Works	Brian Ward	No Starch Press	3rd Ed, 2021	978-1718500402
6	UNIX and Linux System Administration Handbook	Nemeth et al.	Addison-Wesley	5th Ed, 2017	978-0134277554
7	The C Programming Language	Kernighan, Ritchie	Prentice Hall	2nd Ed, 1988	978-0131103627

COURSE CODE: 26UC1108E
COURSE NAME :ENGLISH COMMUNICATION
L-T-P-S : 2-1-0-0 **PRE-REQ : NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply listening comprehension strategies to professional contexts including lectures, technical talks, podcasts, and meetings; produce structured notes and summaries.	PO9, PO10	3
CO2	Apply spoken English fluency in professional and academic contexts including team discussions, brief presentations, and technical conversations using clear pronunciation and pacing.	PO8, PO9	3
CO3	Apply reading strategies (skimming, scanning, intensive, critical) to technical articles, news pieces, research abstracts, and literary excerpts.	PO9, PO10	3
CO4	Analyze grammar, syntax, and vocabulary errors in own and peers' writing; produce error-free professional prose using active voice and clear sentence structure.	PO2, PO9	4
CO5	Apply academic and professional writing conventions to compose emails, reports, summaries, and short essays appropriate to the audience and purpose.	PO9, PO10	3

24/7/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO6	Examine cross-cultural communication patterns and apply situational awareness to navigate workplace dialogues, interview interactions, and team feedback.	PO8, PO9	4

Syllabus:

Module No.	Syllabus
M1	**Listening & Speaking Foundations** Mapped: **CO1, CO2** · 10 hrs • Active listening • note-taking • pronunciation drills (vowel/consonant sounds, stress, intonation, contractions) • minimal pairs • word stress and sentence rhythm • British vs American accents • Indian English standardization • podcasts and TED talks for input • debate fundamentals • expressing opinions and disagreements politely • small-talk strategies in professional settings *Daily 30-min listening practice (assigned podcasts); weekly speaking lab (pair discussions, mini-debates, 3-min impromptu talks); peer feedback rubric*
M2	**Reading & Vocabulary** Mapped: **CO3, CO4** · 10 hrs • Skimming and scanning techniques • intensive vs critical reading • identifying main idea, supporting details, inferences • vocabulary building (academic word list, technical terms, collocations, idioms) • context clues • root-prefix-suffix decoding • reading for evidence • identifying author's tone, purpose, bias • speed reading and comprehension trade-offs *Weekly reading assignments (news articles, technical pieces, short stories); vocabulary journal (5 new words/day with usage); reading comprehension quizzes; one critical reading exercise per week*

31/7/26
HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M3	**Grammar & Sentence Structure** Mapped: **CO4, CO5** · 8 hrs • Tenses and aspect (simple, continuous, perfect, perfect-continuous) • modal verbs (ability, obligation, possibility, advice) • voice (active vs passive — when each is appropriate) • conditionals (zero, first, second, third, mixed) • reported speech • subject-verb agreement • common Indian English errors (article use, prepositions, tense shifts) • sentence types (simple, compound, complex) • punctuation discipline *Grammar workshop format with diagnostic tests; targeted exercises per error type; peer-edit sessions; one writing piece per module corrected and re-submitted*
M4	**Professional Writing** Mapped: **CO5, CO6** · 10 hrs • Email writing (subject lines, greetings, structure, tone, sign-offs, formal vs informal) • technical reports • meeting minutes • project descriptions • status updates • LinkedIn profile writing • resume and cover letter fundamentals • summary writing (compression, paraphrasing, attribution) • citing sources • avoiding plagiarism in academic writing *Weekly writing assignment (300-500 words) with structured feedback; portfolio of 6 polished pieces by end of term; peer review pairs*
M5	**Communication in Context** Mapped: **CO2, CO6** · 7 hrs • Cross-cultural communication awareness • Indian English in global context • interviews (HR vs technical, behavioral questions, STAR method) • workplace conversations (feedback, escalation, disagreement) • team meetings and standups • presentation skills (structure, slides, delivery, Q&A) • body language and non-verbal cues • managing nervousness • reading the room *Mock interviews; team meeting role-plays; one 5-minute presentation per student with*

31/7/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	peer + instructor critique; final integrated communication exercise

Reference Books:

1. Murphy, R. — English Grammar in Use (5th ed.), Cambridge 2019, ISBN 978-1108457651
2. Sethi, J., Dhamija, P. — A Course in Phonetics and Spoken English (2nd ed.), PHI 2009, ISBN 978-8120311466
3. Sweeney, S. — English for Business Communication (2nd ed.), Cambridge 2012, ISBN 978-1107618367
4. Kumar, S., Lata, P. — Communication Skills (2nd ed.), Oxford India 2015, ISBN 978-0199457069
5. Strunk, W., White, E.B. — The Elements of Style (4th ed.), Pearson 2000, ISBN 978-0205309023
6. Day, R., Gastel, B. — How to Write and Publish a Scientific Paper (8th ed.), Cambridge 2017, ISBN 978-1316640432

COURSE CODE: 26CA1204E
COURSE NAME :DATA STRUCTURES(Java)
L-T-P-S : 3-0-6-0 **PRE-REQ : JAPR**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply linear data structures (arrays, linked lists) and asymptotic analysis (Big-O) to evaluate the cost of operations on application data.	PO1, PO2	3
CO2	Apply abstract data types (stacks, queues, deques) and recursive algorithms to solve problems with natural LIFO/FIFO/recursive structures.	PO1, PO3	3
CO3	Analyze tree-based data structures (BST, AVL, heap) to choose appropriate structures for ordered or priority-based access patterns.	PO2, PO3	4
CO4	Analyze hash-based data structures and collision handling to design efficient lookup-heavy data layers.	PO2, PO4	4
CO5	Evaluate graph data structures and traversal algorithms (BFS, DFS, Dijkstra) to solve real-world routing and connectivity problems.	PO3, PO4	5
CO6	Evaluate the performance characteristics of competing data-structure choices and refactor application code to use the optimal structure for each access pattern.	PO3, PO5	5

3/17/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Syllabus:

Module No.	Syllabus
M1	**Arrays, Linked Lists & Asymptotic Analysis** Mapped: **CO1 (BTL-3)** · 12 hrs · Project Stage 1 — Place catalog • Asymptotic notation: Big-O, Big-Ω, Big-Θ; intuitive analysis • Time vs space complexity; amortized analysis (intro) • Arrays: fixed vs dynamic (ArrayList revisited) • Singly, doubly, circular linked list implementations • When to choose array vs linked list (cache locality, random access) • Skip lists (preview of probabilistic structures) *Students implement their own linked list and benchmark it against ArrayList on the project's place-catalog dataset.*
M2	**Stacks, Queues & Recursion** Mapped: **CO2 (BTL-3)** · 12 hrs · Project Stage 2 — Path history • Stack ADT: array-based and linked-list-based implementations • Queue ADT: FIFO semantics, circular array implementation • Deque (double-ended queue) and priority queue • Recursion: base case, recursive case, call stack visualization • Tail recursion and recursion-to-iteration conversion • Classic problems: balanced parentheses, infix-to-postfix • Memoization basics (preview of DP) *The project gains an undo/history stack and a queue for batch path requests. Recursion is introduced via a visit-all-rooms problem.*
M3	**Trees & Binary Search Trees** Mapped: **CO3 (BTL-4)** · 12 hrs · Project Stage 3 — Indexed search • Tree terminology: root, leaf, height, depth, balanced/unbalanced • Binary trees: traversals (in-order, pre-order, post-order, level-order) • BST: search, insert, delete; worst-case degeneration • AVL tree: rotations to maintain balance • Heap: max-heap, min-heap, priority queue applications • B-tree (intro for awareness) • Trie for string lookup (autocomplete) *A BST is added to support fast lookup of named places. A min-heap powers the priority queue used by Dijkstra in Module 5.*

31/1/26

HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M4	**Hashing & Hash Tables** Mapped: **CO4 (BTL-4)** · 12 hrs · Project Stage 4 — Fast lookup • Hash function design: distribution, determinism, low collision rate • Java's hashCode() contract revisited • Collision resolution: separate chaining, open addressing (linear, quadratic, double hashing) • Load factor and rehashing • HashMap internals (Java) and Python dict • Bloom filter (probabilistic existence test) • Consistent hashing (preview) • When NOT to use hash tables *Students implement HashMap from scratch with both collision strategies, then benchmark against java.util.HashMap.*
M5	**Graphs & Graph Algorithms** Mapped: **CO5 (BTL-5)** · 12 hrs · Project Stage 5 — Routing engine • Graph representations: adjacency matrix vs adjacency list • Directed vs undirected, weighted vs unweighted • BFS for shortest path in unweighted graphs • DFS for connectivity, cycle detection, topological sort • Dijkstra's algorithm for weighted shortest path • A* search (heuristic-driven pathfinding) • Minimum spanning tree: Prim's and Kruskal's • Real-world graph problems: maps, social networks, dependencies *The project becomes a working campus navigation system. Students implement Dijkstra and A* and compare paths visually.*
M6	**Performance Engineering & DS Selection** Mapped: **CO6 (BTL-5)** · 12 hrs · Project Stage 6 — Production hardening • Choosing the right data structure: a decision-tree heuristic • JMH (Java Microbenchmark Harness) basics • Cache locality and CPU cache lines • Memory layout: object headers, padding, primitives vs boxed • Concurrent data structures: ConcurrentHashMap, CopyOnWriteArrayList • External-memory data structures (intro) • Property-based testing with jqwik • The art of refactoring brute-force to appropriate DS *Students profile the project's hottest paths, identify suboptimal DS choices, and refactor with measured improvements.*

Reference Books:

31/7/26
HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Sno	Title	Author(s)	Publisher	Edition / Year	ISBN
1	Introduction to Algorithms (CLRS)	Cormen, Leiserson, Rivest, Stein	MIT Press	4th Ed, 2022	978-0262046305
2	Algorithms	Sedgewick, Wayne	Addison-Wesley	4th Ed, 2011	978-0321573513
3	Data Structures and Algorithms in Java	Goodrich, Tamassia, Goldwasser	Wiley	6th Ed, 2014	978-1118771334
4	Cracking the Coding Interview	Gayle Laakmann McDowell	CareerCup	6th Ed, 2015	978-0984782857
5	Grokking Algorithms	Aditya Bhargava	Manning	1st Ed, 2016	978-1617292231
6	The Algorithm Design Manual	Steven S. Skiena	Springer	3rd Ed, 2020	978-3030542559

COURSE CODE:26CA1205E
COURSE NAME :PYTHON PROGRAMMING
L-T-P-S : 3-0-3-0 **PRE-REQ : NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply Python's idiomatic features — comprehensions, generators, decorators, dynamic typing — to write clean, expressive Python programs that contrast with Java's static style.	PO1, PO5	3
CO2	Apply Python data structures (lists, dicts, sets, tuples) and File/JSON/CSV I/O to ingest, transform, and persist real-world structured data.	PO1, PO3	3
CO3	Analyze data with NumPy and pandas — vectorized operations, DataFrames, groupby, joins — to extract insights from realistic datasets.	PO2, PO5	4
CO4	Analyze visualization choices using matplotlib, seaborn, and Plotly to design effective charts that communicate data narratives.	PO2, PO5	4

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

CO5	Evaluate strategies for consuming external data via HTTP APIs, web scraping, and authentication patterns to build data-integrated Python applications.	PO3, PO5	5
CO6	Evaluate the deployment readiness of a Python data application — testing with pytest, packaging, virtual environments, and Streamlit for shareable dashboards.	PO5, PO10	5

Syllabus:

Module No.	Syllabus
M1	**Pythonic Foundations & Idioms** Mapped: **CO1 (BTL-3)** · 10 hrs · Project Stage 1 — Pythonic refactor • Python data model: everything-is-an-object, mutability vs immutability • Dynamic typing, duck typing, type hints (PEP 484) • List/dict/set comprehensions, generator expressions • Functions: default args, *args, **kwargs, lambda, closures • Decorators (functools.wraps, parametrized decorators) • Context managers and the with statement • Pythonic style: PEP 8, naming conventions, idiomatic patterns • How Python differs from Java: when each is the right tool *Students migrate a small Java program to Python to viscerally feel the language differences. The project's first stage establishes the Pythonic baseline.*
M2	**Data Structures, File & JSON I/O** Mapped: **CO2 (BTL-3)** · 10 hrs · Project Stage 2 — Data ingestion • Built-in collections: list, dict, set, tuple — when to use which • collections module: namedtuple, defaultdict, Counter, deque • Dataclasses (PEP 557) for structured records • File I/O: open(), with-blocks, encoding awareness • CSV with the csv module and pandas.read_csv • JSON serialization/deserialization; nested structures • Path manipulation with pathlib • Configuration files: TOML, YAML, INI *Students load a CSV/JSON dataset (personal finance, fitness, study time) into typed Python data structures.*

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K L Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-522502

M3	**NumPy & Pandas** Mapped: **CO3 (BTL-4)** · 10 hrs · Project Stage 3 — Analysis core • NumPy: ndarray, broadcasting, vectorization vs Python loops • Pandas: Series, DataFrame, indexing (loc/iloc), boolean masks • Reading data: read_csv, read_json, read_excel, read_sql • Cleaning: missing values, type conversion, deduplication • Transformation: apply, map; aggregation with groupby • Combining data: concat, merge, join (SQL-equivalent toolkit) • Pivot tables and reshaping (melt, pivot, stack, unstack) • Time-series basics: dates, resampling, rolling windows *Students compute meaningful analytics on their dataset (top categories, monthly trends, anomalies). pandas vs naive loop benchmarks on 100K rows.*
M4	**Visualization for Data Stories** Mapped: **CO4 (BTL-4)** · 10 hrs · Project Stage 4 — Visual narratives • matplotlib: Figure/Axes, plot types, styling, subplots • seaborn: statistical visualization, default aesthetics • Plotly Express: interactive charts in 1 line • Choosing the right chart for each insight • Color theory and accessibility (colorblind-safe palettes) • Dashboards: laying out multiple charts coherently • Common visualization mistakes (truncated axes, misleading scales) *Students build 6+ charts that together tell a story about their dataset. Peer review focuses on chart-level clarity.*
M5	**APIs, Web Data & Integration** Mapped: **CO5 (BTL-5)** · 10 hrs · Project Stage 5 — External data • HTTP fundamentals from a Python lens (request/response) • The requests library: GET, POST, query params, JSON payloads • Authentication: API keys, OAuth tokens, headers • Rate limiting, retries, exponential backoff • Web scraping ethics; BeautifulSoup for HTML parsing • Public APIs (OpenWeatherMap, NewsAPI, Yahoo Finance) • Caching with requests-cache • Error handling: graceful degradation when APIs fail *Students augment their dataset with live external data — correlate spending with weather, study time with news cycles. The project becomes networked.*
M6	**Streamlit, Testing & Deployment** Mapped: **CO6 (BTL-5)** · 10 hrs · Project Stage 6 — Shareable app • Virtual environments: venv, pip, requirements.txt

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• Streamlit: turning a Python script into a web app • Streamlit components: widgets, layout, state, @st.cache • pytest fundamentals: writing tests, fixtures, parametrize • Coverage with pytest-cov • Type checking with mypy as a quality gate • Deployment: Streamlit Community Cloud (free). GitHub integration • Code style: black formatter, ruff linter, pre-commit hooks *Students publish their dashboard to Streamlit Community Cloud. URL is shareable. Real audience = real motivation.*

Reference Books:

Sno	Title	Author(s)	Publisher	Edition / Year	ISBN
1	Python Crash Course	Eric Matthes	No Starch Press	3rd Ed, 2023	978-1718502703
2	Fluent Python	Luciano Ramalho	O'Reilly	2nd Ed, 2022	978-1492056355
3	Python for Data Analysis	Wes McKinney	O'Reilly	3rd Ed, 2022	978-1098104030
4	Effective Python	Brett Slatkin	Addison-Wesley	2nd Ed, 2019	978-0134853987
5	Automate the Boring Stuff with Python	Al Sweigart	No Starch Press	2nd Ed, 2019	978-1593279929
6	Python Data Science Handbook	Jake VanderPlas	O'Reilly	2nd Ed, 2023	978-1098121228
7	Architecture Patterns with Python	Percival, Gregory	O'Reilly	1st Ed, 2020	978-1492052203

COURSE CODE:26CA1206E
COURSE NAME :FRONTEND DEVELOPMENT
L-T-P-S : 0-0-9-0 **PRE-REQ : JAPR**

CO.	Course Outcome Description	PO	BTL

31/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applica
Koneru Lakshmaiah Education Found
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-52

No		MAPPING	
CO1	Apply semantic HTML5, modern CSS (Flexbox, Grid, custom properties), and responsive design (mobile-first, media queries) to build accessible, well-structured web pages.	PO1, PO3	3
CO2	Apply modern JavaScript (ES2020+: let/const, arrow functions, modules, async/await, destructuring, spread/rest) and the DOM API to build interactive web pages.	PO1, PO5	3
CO3	Analyze React fundamentals (components, JSX, props, state, hooks — useState, useEffect, useContext) to architect single-page applications.	PO2, PO3	4
CO4	Analyze React Router for client-side routing, plus state management patterns (lifting state, Context API, intro to Zustand/Redux) for non-trivial apps.	PO2, PO5	4
CO5	Evaluate API integration patterns — fetch / axios, React Query (TanStack Query) for caching, optimistic updates, error handling, JWT-protected requests — to consume the TaskMaster backend.	PO3, PO5	5
CO6	Evaluate frontend tooling (Vite, ESLint, Prettier), testing (Jest + React Testing Library), accessibility (WCAG basics), and deployment (Vercel / Netlify / Cloudflare Pages).	PO5, PO10	5

Syllabus:

Module No.	Syllabus
M1	**HTML5 & Modern CSS** Mapped: **CO1 (BTL-3)** · 10 hrs · Project Stage 1 — Static prototype • Semantic HTML5: header, nav, main, section, article, footer, aside • Form elements; ARIA attributes for accessibility (intro) • CSS fundamentals: selectors, specificity, the cascade, box model • Modern layout: Flexbox (1D) and Grid (2D) — when each fits • CSS custom properties (variables) and the modern reset • Responsive design: mobile-first, media queries, fluid typography • CSS architecture: BEM, utility-first (Tailwind preview) • Accessibility basics: contrast, keyboard nav, focus states *Students build a static, fully responsive TaskMaster-Web mockup with mock data — no JavaScript yet. Faculty validate WCAG-AA contrast and keyboard navigation.*

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M2	**Modern JavaScript & the DOM** Mapped: **CO2 (BTL-3)** · 10 hrs · Project Stage 2 — Interactive prototype • ES2020+ essentials: let/const, arrow functions, default params • Destructuring (object & array); spread/rest operators • Modules: import/export • Promises and async/await; the event loop intuition • Array methods: map, filter, reduce, find, some, every • The DOM API: querySelector, addEventListener, classList • Fetch API: GET/POST, headers, error handling • localStorage and sessionStorage; the persistence escape hatch *Students add vanilla JS interactivity to the static mockup: filtering tasks, marking complete, persisting to localStorage. No frameworks. They feel the DOM friction that React solves.*
M3	**React Fundamentals** Mapped: **CO3 (BTL-4)** · 12 hrs · Project Stage 3 — React rewrite • Why React: declarative + component model • Vite scaffold: project structure, dev server, build • JSX: it's not HTML, the gotchas (className, htmlFor) • Function components; props and prop drilling • State with useState; the rules of hooks • Effects with useEffect: dependencies, cleanup, common bugs • Context with useContext for cross-cutting values (theme, user) • Custom hooks: extracting reusable logic • Conditional rendering, list rendering with keys *Students rebuild the Module 2 mockup in React. Same UI, different architecture. The difference in code maintainability is felt.*
M4	**Routing & State Management** Mapped: **CO4 (BTL-4)** · 10 hrs · Project Stage 4 — Multi-page app • Client-side routing concept; React Router v6 • Routes, nested routes, params, query strings • Programmatic navigation; redirects; protected routes • State management spectrum: local, lifted, context, store • The prop drilling problem and Context API solution • Zustand (recommended) or Redux Toolkit (intro) • When to use what: heuristics for choosing • Form state: controlled vs uncontrolled; React Hook Form *TaskMaster-Web becomes a multi-route SPA: /login, /projects, /projects/:id, /tasks/:id, /profile. Login state shared via context.*

31/1/26

HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicatic
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522 50

M5	**API Integration with Real Backend** Mapped: **CO5 (BTL-5)** · 10 hrs · Project Stage 5 — Wired to backend • fetch vs axios; setting up an API client • JWT auth flow: login, store token, attach to requests • Token refresh on 401; auto-retry pattern • React Query (TanStack Query): caching, refetching, mutations • Optimistic updates: marking task complete instantly • Error states: loading, error, empty, success • Pagination + infinite scroll patterns • WebSocket basics for real-time updates (intro) *TaskMaster-Web is wired to the actual DBBE TaskMaster API. Students log in, create projects, manage tasks — full end-to-end. Frontend + backend are now one product.*
M6	**Tooling, Testing, Accessibility & Deploy** Mapped: **CO6 (BTL-5)** · 10 hrs · Project Stage 6 — Production • Vite: dev mode, build mode, env vars • ESLint + Prettier: enforcing consistency in a team • Jest + React Testing Library: testing components, not implementation • Accessibility audit: axe-core, Lighthouse, manual screen reader test • Performance: lazy loading routes, code splitting, image optimization • Build artifacts; static hosting fundamentals • Deploy to Vercel / Netlify / Cloudflare Pages • Environment variables, build-time vs runtime config *TaskMaster-Web is publicly deployed at a real URL with a CI pipeline that runs tests + lint + build on every PR. Lighthouse score > 90 across all four metrics.*

Reference Books:

Sno	Title	Author(s)	Publisher	Edition / Year	ISBN
1	HTML and CSS: Design and Build Websites	Jon Duckett	Wiley	1st Ed, 2011 (still highly recommended)	978-1118008188
2	JavaScript: The Definitive Guide	David Flanagan	O'Reilly	7th Ed, 2020	978-1491952023
3	Eloquent JavaScript	Marijn Haverbeke	No Starch / free online	4th Ed, 2024	978-1593279509

21/7/26
HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

4	Learning React	Alex Banks, Eve Porcello	O'Reilly	2nd Ed, 2020	978-1492051725
5	The Road to React	Robin Wieruch	Self-published	2024 Ed	—
6	CSS in Depth	Keith J. Grant	Manning	2nd Ed, 2024	978-1633437555
7	Refactoring UI	Adam Wathan, Steve Schoger	Self-published	2018	—

COURSE CODE:26UC1209E
COURSE NAME :TECHNICAL & PROFESSIONAL COMMUNICATION
L-T-P-S : 2-1-0-0 **PRE-REQ : ECM**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply technical writing conventions to produce documentation including README files, API docs, runbooks, and design documents using clear, concise prose.	PO9, PO10	3
CO2	Apply scientific and technical presentation norms to deliver short technical talks (Lightning, conference style, demo) with structured slides and confident Q&A handling.	PO8, PO9	3
CO3	Analyze technical documentation patterns (Diátaxis framework: tutorial, how-to, reference, explanation) and select appropriate structure for given communication tasks.	PO2, PO9	4
CO4	Examine cross-functional communication scenarios (developer-to-PM, developer-to-designer, developer-to-business stakeholder) and adapt language to audience expertise level.	PO8, PO9	4
CO5	Apply business communication conventions including meeting facilitation, status reporting, escalation, and cross-team negotiation in professional simulation contexts.	PO8, PO9	3
CO6	Examine ethical communication including disclosure, attribution, dissent, whistleblowing protocols, and Indian workplace communication norms (hierarchy, indirect speech, face-saving).	PO6, PO9	4

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Syllabus:

Module No.	Syllabus
M1	**Technical Writing Foundations** Mapped: **CO1, CO3** · 10 hrs • Diátaxis framework: tutorial vs how-to vs reference vs explanation • README anatomy • API documentation conventions • code comments and docstrings • commit messages • pull request descriptions • technical blog posts • documentation as a product (versioning, search, accessibility) • writing for translation and accessibility (plain language) *Students draft a README, API doc, and 600-word technical blog post on a topic from their JAVA or PYTH course; peer review using a structured rubric*
M2	**Presentations & Public Speaking** Mapped: **CO2, CO4** · 8 hrs • Slide design (Tufte minimalism vs Lessig style • one idea per slide) • narrative arc (problem → context → approach → result → next) • Lightning talks • demo discipline (rehearse, fail-safes, narrate the journey) • voice modulation • pacing • controlling nervousness • handling tough questions • AMA format • conference vs workshop talk styles *Each student delivers 3 short talks across the term: 3-min lightning, 5-min explainer, 8-min demo; peer + faculty critique via rubric*
M3	**Cross-Functional Communication** Mapped: **CO4, CO5** · 8 hrs • Translating technical to non-technical audiences (executive summaries, stakeholder updates) • communicating uncertainty and risk • estimating and pushing back on timelines • design discussions across roles

31/7/26
HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

	• handing-off and onboarding documentation • reading and writing requirements (PRDs, specs, ADRs) *Role-play simulations where students play developer, PM, designer, business stakeholder; documents are written for each role's perspective on the same project*
M4	**Meetings, Reporting & Escalation** Mapped: **CO5, CO6** · 7 hrs • Meeting hygiene (agendas, time-boxing, action items) • standup discipline (yesterday-today-blockers) • sprint planning and retrospectives • status reports for managers vs leadership vs board • written escalation • handling conflicting opinions • disagreeing without being disagreeable *Mock standups, sprint plannings, and retrospectives; weekly status report writing; scenario-based escalation exercises*
M5	**Ethics & Indian Workplace Communication** Mapped: **CO6, CO5** · 7 hrs • Disclosure ethics in technical communication (AI assistance, sponsored content, conflicts of interest) • attribution and plagiarism in code and docs • whistleblowing protocols (Sarbanes-Oxley parallel for India: Whistle Blowers Protection Act 2014) • Indian workplace communication norms (hierarchy, face-saving, indirect speech, when explicitness is required) • virtual communication etiquette *Case-study analysis of real ethical communication failures (Theranos memos, Boeing 737 MAX, Indian PSU whistleblower cases); position-paper assignment*

Reference Books:

1. Procida, D. — Diátaxis: A systematic approach to technical documentation (free, diataxis.fr)
2. Strunk & White — The Elements of Style (4th ed.), Pearson 2000
3. Williams, J., Bizup, J. — Style: Lessons in Clarity and Grace (12th ed.), Pearson 2017
4. Tufte, E. — The Visual Display of Quantitative Information (2nd ed.), Graphics Press 2001
5. Anderson, C. — TED Talks: The Official TED Guide to Public Speaking, HMH 2016
6. Larson, W. — An Elegant Puzzle (free online); Staff Engineer (Stripe Press 2021, free PDF)
7. Newport, C. — Deep Work, Grand Central 2016

COURSE CODE:26CA1307E
COURSE NAME :APPLIED AI & MACHINE LEARNING
L-T-P-S : 3-0-6-0 **RE-REQ : DSJV**

31/3/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522 [illegible]

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply the standard ML workflow — problem framing, data acquisition, EDA, feature engineering, model training, evaluation — to a real-world structured dataset.	PO1, PO3	3
CO2	Apply supervised learning algorithms — linear/logistic regression, k-NN, SVM, naive Bayes — using scikit-learn to solve regression and classification problems.	PO1, PO5	3
CO3	Analyze model performance using cross-validation, confusion matrices, ROC curves, precision/recall trade-offs to diagnose underfitting, overfitting, and bias-variance issues.	PO2, PO4	4
CO4	Analyze unsupervised learning techniques — k-means, hierarchical clustering, PCA, t-SNE — to discover structure in unlabeled data.	PO2, PO4	4
CO5	Evaluate ensemble methods (bagging, random forests, gradient boosting, XGBoost) and a basic neural network (PyTorch) to choose the right algorithm class for a given problem.	PO3, PO5	5
CO6	Evaluate ML production concerns — model serialization, REST API serving, monitoring, drift, ethical considerations — to deploy a trained model as a usable service.	PO5, PO6	5

Syllabus:

Module No.	Syllabus
M1	**ML Workflow & Exploratory Data Analysis** Mapped: **CO1 (BTL-3)** · 12 hrs · Project Stage 1 — Problem framing & baseline • What ML is and isn't; types of learning (supervised, unsupervised, reinforcement) • Standard ML workflow: problem → data → features → model → eval → deploy • Data sources: Kaggle, UCI, scikit-learn datasets, live APIs • Exploratory Data Analysis (EDA): missing values, distributions, correlations • Feature engineering: categorical encoding, scaling, binning, derived features • Train-test-validation split; the danger of data leakage • Baseline models: always-predict-mean, always-predict-majority-class • scikit-learn API: fit, predict, transform, score *Students choose a real dataset in Week 1, run EDA, and build a no-ML baseline*

3/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	before any modeling. The baseline becomes the floor every later model must beat.
M2	**Supervised Learning Foundations** Mapped: **CO2 (BTL-3)** · 12 hrs · Project Stage 2 — First models • Linear regression: closed-form solution and gradient descent (cross-link MTCA M3) • Polynomial regression and the bias-variance trade-off • Regularization: Ridge (L2), Lasso (L1), Elastic Net • Logistic regression for binary classification • k-Nearest Neighbors: simple but powerful • Support Vector Machines: linear, kernel trick (RBF, polynomial) • Naive Bayes: when independence assumption is acceptable • Hyperparameter intuition: what does "C" do in SVM, what does "k" do in k-NN *Students implement linear regression from scratch using NumPy (cross-link MTCA), then re-implement using scikit-learn. Compare to verify correctness.*
M3	**Model Evaluation & Tuning** Mapped: **CO3 (BTL-4)** · 12 hrs · Project Stage 3 — Tuning loop • Train/validation/test methodology; k-fold cross-validation • Regression metrics: MAE, MSE, RMSE, R^2 • Classification metrics: accuracy, precision, recall, F1, ROC-AUC • Confusion matrix interpretation; the cost of false positives vs false negatives • Underfitting vs overfitting: learning curves • Bias-variance decomposition • Hyperparameter tuning: GridSearchCV, RandomizedSearchCV, Optuna intro • Feature selection: forward, backward, mutual information, recursive elimination *Students cross-validate every Module-2 model on the project dataset and produce a leaderboard. The winner so far is identified.*
M4	**Unsupervised Learning** Mapped: **CO4 (BTL-4)** · 12 hrs · Project Stage 4 — Hidden structure • k-means clustering: algorithm, initialization, k selection (elbow, silhouette) • Hierarchical clustering: agglomerative, divisive; dendrograms • DBSCAN for density-based clustering • Gaussian Mixture Models (intro) • Principal Component Analysis (PCA): connection to SVD (cross-link MTCA M2) • t-SNE and UMAP for high-dim visualization

31/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields
Guntur District, Andhra Pradesh

	• Anomaly detection: Isolation Forest, One-Class SVM • When to use unsupervised: customer segmentation, anomaly detection, exploratory data reduction *Students apply clustering to their dataset to find natural segments, then verify whether the segments improve a downstream supervised model.*
M5	**Ensembles & Neural Networks (Intro)** Mapped: **CO5 (BTL-5)** · 12 hrs · Project Stage 5 — Strong models • Decision trees: how splits are chosen (entropy, Gini); pruning • Bagging: random forests; out-of-bag error • Boosting intuition: AdaBoost, gradient boosting • XGBoost, LightGBM in practice — typical Kaggle-winner configurations • Stacking and blending • Multi-Layer Perceptron in PyTorch (very brief intro): forward pass, backprop, loss, optimizer • Train an MLP on the project dataset and compare to XGBoost • When NOT to use a neural network *Students train Random Forest, XGBoost, and a small MLP on their project. Honest comparison: which wins, why, and at what training-time cost.*
M6	**ML Production & Ethics** Mapped: **CO6 (BTL-5)** · 12 hrs · Project Stage 6 — Deployed model • Model serialization: pickle, joblib, ONNX • Serving via Flask or FastAPI: a thin REST wrapper around a model • Streamlit for ML demo apps (cross-link PYTH) • Containerization: Dockerfile for an ML service (preview of CLDO) • Cloud deployment: AWS Lambda or HuggingFace Spaces or Streamlit Cloud • Monitoring: data drift, concept drift, model staleness • Bias and fairness: protected attributes, fairness metrics • Reproducibility: random seeds, environment pinning, MLflow basics *Students publish a public URL where anyone can submit input and get a prediction from their trained model. README documents one concrete fairness concern they identified.*

Reference Books:

Sno	Title	Author(s)	Publisher	Edition / Year	ISBN
1	*Hands-On Machine Learning with Scikit-*	Aurélien Géron	O'Reilly	3rd Ed, 2022	978-1098125974

24/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

	Learn, Keras, and TensorFlow				
2	*An Introduction to Statistical Learning* (ISLR / ISLP)	James, Witten, Hastie, Tibshirani	Springer	2nd Ed, 2021	978-1071614174
3	*Pattern Recognition and Machine Learning*	Christopher M. Bishop	Springer	1st Ed, 2006	978-0387310732
4	*The Elements of Statistical Learning*	Hastie, Tibshirani, Friedman	Springer	2nd Ed, 2009	978-0387848570
5	*Designing Machine Learning Systems*	Chip Huyen	O'Reilly	1st Ed, 2022	978-1098107963
6	*Deep Learning with PyTorch*	Stevens, Antiga, Viehmann	Manning	1st Ed, 2020	978-1617295263
7	*Fairness and Machine Learning*	Barocas, Hardt, Narayanan	fairmlbook.org (free)	2023	—

COURSE CODE:26CA1308E
COURSE NAME :DBMS & BACKEND DEVELOPMENT

L-T-P-S : 3-0-6-0 **PRE-REQ : PYPR**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply relational database design principles (normalization, keys, constraints) and SQL (DDL, DML, joins, subqueries, CTEs) to design and query non-trivial schemas.	PO1, PO3	3
CO2	Apply ORM patterns (JPA / Hibernate) and migration tools (Flyway / Liquibase) to manage schema evolution and persist Java objects.	PO3, PO5	3
CO3	Analyze RESTful API design — resource modeling, HTTP semantics, status codes, versioning, OpenAPI documentation — to build well-structured Spring Boot services.	PO2, PO3	4

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO4	Analyze authentication, authorization, input validation, and error-handling patterns in backend systems to harden services against common vulnerabilities.	PO2, PO6	4
CO5	Evaluate trade-offs between SQL and NoSQL stores (PostgreSQL vs MongoDB vs Redis) and apply caching strategies to optimize backend latency and throughput.	PO3, PO5	5
CO6	Evaluate the production readiness of a backend — testing strategy, observability (logging, metrics), graceful shutdown, deployment via Docker — to ship a public REST service.	PO5, PO10	5

Syllabus:

Module No.	Syllabus
M1	**Relational Modeling& SQL** Mapped: **CO1 (BTL-3)** · 12 hrs · Project Stage 1 — Schema design • Relational model: tables, rows, columns, primary/foreign keys • Normalization: 1NF, 2NF, 3NF, BCNF (intuitive); when to denormalize • ER diagrams and translation to schemas • DDL: CREATE TABLE, ALTER, DROP; constraints (NOT NULL, UNIQUE, CHECK) • DML: SELECT, INSERT, UPDATE, DELETE • Joins: INNER, LEFT, RIGHT, FULL, SELF; on a real schema • Subqueries, common table expressions (CTEs), window functions intro • Indexes: B-tree, hash; the cost of indexes; EXPLAIN basics *Students design TaskMaster's schema with users, projects, tasks, comments, attachments tables. SQL queries are written by hand for every requirement before any ORM appears.*
M2	**ORM & Migrations** Mapped: **CO2 (BTL-3)** · 12 hrs · Project Stage 2 — Java persistence • JDBC review; the verbose-Java problem ORMs solve • JPA annotations: @Entity, @Id, @GeneratedValue, @Column • Relationships: @OneToMany, @ManyToOne, @ManyToMany; cascade and orphan removal • JPQL vs native SQL; when to use each • The N+1 query problem and how to fix it (fetch joins, EntityGraph) • Spring Data JPA: repositories, derived queries, paging and sorting • Flyway / Liquibase migrations; versioned schema evolution • Test data: SQL seeds, @Sql annotation, Testcontainers

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	TaskMaster gains its persistence layer. Students intentionally write the N+1 problem, observe it in the SQL log, then fix it with a fetch join.
M3	**REST API Design with Spring Boot** Mapped: **CO3 (BTL-4)** · 12 hrs · Project Stage 3 — Public API • HTTP fundamentals: methods, status codes, headers, content negotiation • REST principles: resource modeling, statelessness, HATEOAS (briefly) • Spring Boot scaffold: starter, application.yml, profiles • @RestController, @GetMapping/@PostMapping, @PathVariable, @RequestBody • DTO pattern; MapStruct or manual mapping • Pagination & sorting via Spring Data; consistent response envelopes • API versioning strategies: URI, header, content negotiation • OpenAPI 3.0 with springdoc-openapi; auto-generated docs *TaskMaster's public REST API is built. Every endpoint has request/response DTOs, validation, and OpenAPI documentation. Students consume their own API with curl and Postman.*
M4	**Auth, Validation & Errors** Mapped: **CO4 (BTL-4)** · 12 hrs · Project Stage 4 — Hardening • Authentication vs authorization • Password hashing: bcrypt, Argon2; never store plaintext • JWT tokens: structure, signing, validation, refresh tokens • Spring Security 6: filter chain, AuthenticationManager, UserDetailsService • Method-level security: @PreAuthorize, role-based access • Input validation: Bean Validation (@Valid, @NotNull, @Size, custom validators) • Global exception handling: @ControllerAdvice, problem+json (RFC 7807) • OWASP Top 10 awareness for backend developers *TaskMaster gains JWT authentication, role-based access (USER, ADMIN), and a hardened error response format. Each common vulnerability is exploited then fixed.*
M5	**NoSQL, Caching & Search** Mapped: **CO5 (BTL-5)** · 12 hrs · Project Stage 5 — Performance layer • CAP theorem and BASE properties • Document stores (MongoDB): when JSON-shaped data wins over relational • Redis as cache: GET/SET, TTL, eviction policies, cache aside pattern • Spring Cache abstraction: @Cacheable, @CacheEvict • Cache invalidation strategies; the famous "two hard things" problem • Search: PostgreSQL full-text search vs Elasticsearch (intro) • Choosing your data store: a decision-tree heuristic

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• Polyglot persistence: combining stores in one app *Students add Redis caching for hot reads. They measure p50/p95/p99 latency before and after; the improvement is real and documented.*
M6	**Testing, Observability & Deployment** Mapped: **CO6 (BTL-5)** · 12 hrs · Project Stage 6 — Production • Unit tests: JUnit 5 + Mockito (revisit JAVA M6) • Integration tests with @SpringBootTest and Testcontainers (real PostgreSQL in Docker) • Contract tests: REST Assured for API tests • Logging: SLF4J + Logback; structured JSON logs; log levels and correlation IDs • Metrics: Spring Boot Actuator + Micrometer; Prometheus format • Health checks, readiness/liveness probes (preview of CLDO) • Containerizing: multi-stage Dockerfile for Spring Boot fat JAR • Cloud deployment: AWS Elastic Beanstalk, Render, or fly.io free tier *TaskMaster is deployed publicly with a real domain, monitored with metrics, and with a passing CI test suite. Production-grade.*

Reference Books:

Sno	Title	Author(s)	Publisher	Edition / Year	ISBN
1	*Database System Concepts*	Silberschatz, Korth, Sudarshan	McGraw Hill	7th Ed, 2019	978-0078022159
2	*Designing Data-Intensive Applications*	Martin Kleppmann	O'Reilly	1st Ed, 2017	978-1449373320
3	*Spring Boot in Action*	Craig Walls	Manning	1st Ed, 2015 (also 2nd Ed Spring in Action)	978-1617292545
4	*Spring in Action*	Craig Walls	Manning	6th Ed, 2022	978-1617297571
5	*SQL: The Complete Reference*	James Groff, Paul Weinberg	McGraw Hill	3rd Ed, 2010	978-0071592550
6	*Building*	Sam Newman	O'Reilly	2nd Ed, 2021	978-

31/7/26

HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	Microservices				1492034025
7	*RESTful Web APIs*	Leonard Richardson, Mike Amundsen	O'Reilly	1st Ed, 2013	978-1449358068

COURSE CODE:26CA1309E
COURSE NAME :OPERATING SYSTEMS & CONCURRENT PROGRAMMING
L-T-P-S : 2-1-3-0 **PRE-REQ : LXSP**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply process and thread concepts (PCB, TCB, fork/exec, pthreads, Java threads) to write correct multi-process and multi-threaded programs.	PO1, PO3	3
CO2	Apply synchronization primitives (mutexes, semaphores, condition variables, monitors) and detect race conditions in concurrent code.	PO2, PO3	3
CO3	Analyze CPU scheduling algorithms (FCFS, SJF, priority, RR, MLFQ, CFS) by simulating their behavior on workloads and computing turnaround / waiting / response metrics.	PO2, PO4	4
CO4	Analyze memory management techniques (paging, segmentation, virtual memory, page replacement: FIFO, LRU, Optimal) and reason about TLBs, page faults, working sets.	PO2, PO4	4
CO5	Evaluate file system and I/O design (inodes, journaling, buffer cache, async I/O) and detect deadlocks via resource allocation graphs and Banker's algorithm.	PO3, PO4	5
CO6	Evaluate concurrent programming patterns (thread pools, producer-consumer, reader-writer, futures/promises, async/await) for correctness, throughput, and scalability.	PO3, PO5	5

Syllabus:

Module No.	Syllabus

31/7/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M1	**Processes & Threads** Mapped: **CO1 (BTL-3)** · 10 hrs · Project Stage 1 — Process tree • OS overview: kernel vs user mode; system calls (cross-link LINX) • Process: PCB, address space, lifecycle (new, ready, running, waiting, terminated) • fork(), exec(), wait(), exit() — building a process tree • Inter-process communication: pipes, FIFOs, shared memory, signals • Threads: TCB; user vs kernel threads • pthreads in C; Thread class in Java • Stack vs heap memory in threaded programs • The cost of context switches *Students write a tiny shell in C that forks child processes and waits for them. Then they write a Java program that spawns 10 threads competing for shared state.*
M2	**Synchronization & Concurrency Bugs** Mapped: **CO2 (BTL-3)** · 10 hrs · Project Stage 2 — Race conditions • Race conditions: the world's most common concurrency bug • Critical sections; mutual exclusion • Mutexes / locks; reentrant locks • Semaphores: binary and counting • Condition variables and monitors (Java synchronized + wait/notify) • Classic problems: producer-consumer, readers-writers, dining philosophers • Atomic operations and the volatile keyword • Thread-safety of standard library data structures *Students reproduce a real race condition — two threads incrementing a shared counter — see the wrong answer, then fix with a mutex. The bug is felt before the fix.*
M3	**CPU Scheduling** Mapped: **CO3 (BTL-4)** · 10 hrs · Project Stage 3 — Scheduler simulator • Scheduling criteria: throughput, turnaround, waiting time, response time, fairness • Algorithms: FCFS, SJF (preemptive& non-preemptive), priority, Round Robin • Multi-level feedback queues (MLFQ) • Linux's Completely Fair Scheduler (CFS): vruntime intuition • Real-time scheduling: rate monotonic, EDF • Multiprocessor scheduling: load balancing, affinity • Context switch cost; the cache impact • Gantt chart analysis *Students build a scheduler simulator in Java that takes a workload (process burst times, arrival times) and computes metrics for 6 algorithms. Output Gantt charts.*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-522502.

M4	**Memory Management & Virtual Memory** Mapped: **CO4 (BTL-4)** · 10 hrs · Project Stage 4 — VM simulator • Memory hierarchy: registers, cache, RAM, disk • Contiguous allocation: first-fit, best-fit, worst-fit; fragmentation • Paging: page table, frames, address translation • TLB and the cost of misses • Multi-level paging; inverted page tables • Virtual memory: demand paging, page faults • Page replacement: FIFO, LRU, Optimal, Clock (second-chance) • Working set model; thrashing *Students extend the simulator to handle paging. They run the same workload through FIFO, LRU, Optimal and graph fault rates. Reality matches theory.*
M5	**File Systems, I/O & Deadlocks** Mapped: **CO5 (BTL-5)** · 10 hrs · Project Stage 5 — Deadlock detector • File system structure: inodes, directories, links • Allocation: contiguous, linked, indexed; ext4 vs ZFS overview • Journaling and crash consistency • Buffer cache and prefetching • I/O models: blocking, non-blocking, async (epoll, io_uring) • Deadlocks: necessary conditions; resource allocation graphs • Deadlock prevention, avoidance (Banker's algorithm), detection, recovery • Livelock and starvation *Students implement Banker's algorithm and a resource allocation graph cycle detector. They construct deadlock scenarios from the dining philosophers and detect them programmatically.*
M6	**Concurrent Programming Patterns** Mapped: **CO6 (BTL-5)** · 10 hrs · Project Stage 6 — Production server • Thread pools: fixed, cached, scheduled (Java ExecutorService) • Producer-consumer with BlockingQueue • Reader-writer locks: ReentrantReadWriteLock • Futures, CompletableFuture, async chaining • Reactive streams (intro): backpressure • Async/await in Python and JavaScript (cross-link FRNT) • Lock-free data structures (intro to AtomicInteger, ConcurrentHashMap) • Modern alternatives: virtual threads in Java 21+, goroutines (briefly) *Students upgrade their multi-threaded server (from M1) to use a thread pool, BlockingQueue, and CompletableFuture chains. They benchmark throughput improvements.*

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Reference Books:

Sno	Title	Author(s)	Publisher	Edition / Year	ISBN
1	Operating System Concepts (the "Dinosaur" book)	Silberschatz, Galvin, Gagne	Wiley	10th Ed, 2018	978-1119320913
2	Operating Systems: Three Easy Pieces (free online)	Remzi & Andrea Arpaci-Dusseau	self / OSTEP	2018	—
3	Modern Operating Systems	Andrew S. Tanenbaum, Herbert Bos	Pearson	5th Ed, 2022	978-0137618873
4	The Linux Programming Interface	Michael Kerrisk	No Starch Press	1st Ed, 2010	978-1593272203
5	Java Concurrency in Practice	Brian Goetz et al.	Addison-Wesley	1st Ed, 2006	978-0321349606
6	The Art of Multiprocessor Programming	Maurice Herlihy, Nir Shavit	Morgan Kaufmann	2nd Ed, 2020	978-0124159501

COURSE CODE: 26UC0039E
COURSE NAME :INNOVATION & ENTREPRENEURSHIP
L-T-P-S : 2-1-3-0 **PRE-REQ : NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply design-thinking and customer-discovery techniques (problem interviews, empathy mapping, JTBD framework) to identify validated user problems worth solving.	PO2, PO3	3
CO2	Analyzestartup business models using contemporary frameworks (Lean Canvas, Business Model Canvas, value proposition canvas) and assess product-market fit signals.	PO2, PO8	4

3/7/24
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-52250

CO3	Apply lean startup methodology (build-measure-learn) to validate hypotheses through MVPs, A/B tests, and iterative customer development.	PO3, PO8	3
CO4	Examine the Indian startup ecosystem including DPIIT registration, Startup India scheme, angel investing, seed/Series A norms, ESOPs, and exits in 2025-2026 context.	PO8, PO10	4
CO5	Analyze financial fundamentals for technology ventures: unit economics (CAC, LTV, payback), runway, burn rate, capital efficiency, valuation methods (DCF, comps, VC method).	PO2, PO8	4
CO6	Architect a minimum-viable venture pitch including problem statement, solution, market sizing, go-to-market, traction, ask, and team narrative for investor and judge audiences.	PO8, PO9	5

Syllabus:

Module No.	Syllabus
M1	**Design Thinking & Customer Discovery** Mapped: **CO1** · 8 hrs • Problem-first thinking vs solution-first thinking • jobs-to-be-done (JTBD) framework • problem interviews (Steve Blank, Rob Fitzpatrick's 'Mom Test') • empathy mapping • customer journey mapping • defining the ideal customer profile (ICP) • the 'why now' question • cognitive biases in customer discovery • building a hypothesis backlog *Each team identifies a problem space and conducts ≥10 problem interviews; weekly synthesis sessions; iteration on the problem hypothesis*
M2	**Business Models & Product-Market Fit** Mapped: **CO2** · 8 hrs • Lean Canvas vs Business Model Canvas • value proposition canvas • the 9 types of revenue model (SaaS, marketplace, transactional, subscription, freemium, ad-supported, etc.) • product-market fit indicators (Sean Ellis test, retention curves) • Steve Blank's 'Four Steps to the Epiphany'

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522 302

	• competitor analysis • defensibility (network effects, switching costs, brand) *Each team produces Lean Canvas, value proposition canvas, and competitor map; peer review across teams*
M3	**Lean Startup& MVP Building** Mapped: **CO3** · 8 hrs • Build-measure-learn loop • pivot vs persevere decisions • types of MVP (concierge, Wizard of Oz, landing-page, smoke test, single-feature, pre-order) • cohort analysis • A/B testing fundamentals • growth experiments • vanity vs actionable metrics • AARRR funnel (Acquisition, Activation, Retention, Referral, Revenue) *Each team builds a no-code or low-code MVP using NocoDB, Bubble, Glide, Notion, Tally, Airtable, etc.; runs at least one validation experiment with 20+ real users*
M4	**Indian Startup Ecosystem** Mapped: **CO4** · 7 hrs • DPIIT registration and Startup India benefits • Section 80-IAC tax holiday • ANGEL Tax exemption • Mudra/Stand-Up loans • SAMRIDH and TIDE 2.0 programs • angel network landscape (Indian Angel Network, Mumbai Angels, LetsVenture, Inflection Point Ventures) • seed funds and accelerators (Y Combinator, 100x.VC, Sequoia Surge, Accel Atoms) • seed-to-series-A norms • ESOP structures and the 4-year vesting cliff • exits (acquihire, strategic acquisition, IPO) • current Indian unicorn and decacorn snapshot *Case-study analysis of 3 recent Indian startup journeys (across sectors); guest lecture from a founder or investor*
M5	**Startup Finance & Pitching** Mapped: **CO5, CO6** · 7 hrs • Unit economics (CAC, LTV, payback period, contribution margin) • MRR vs ARR • runway and burn rate • capital efficiency ratio

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applic
Koneru Lakshmaiah Education Foun
(K.L. Deemed to be University
Green Fields, VADDESWARA
Guntur District, Andhra Pradesh-52

	• gross margin and net retention • valuation methods (Berkus, Scorecard, Risk-Factor, DCF, comparable companies, VC method) • cap tables and dilution • convertible notes vs SAFEs vs priced rounds • pitch deck anatomy (Sequoia template, YC template) • pitching to angels vs VCs vs strategic investors *Each team builds a 10-slide pitch deck and delivers it to a mock investor panel including outside guests*

Reference Books:

1. Ries, E. — The Lean Startup, Crown Business 2011, ISBN 978-0307887894
2. Blank, S. — The Four Steps to the Epiphany (5th ed.), Wiley 2020, ISBN 978-1119690351
3. Fitzpatrick, R. — The Mom Test (free PDF available)
4. Osterwalder, A., Pigneur, Y. — Business Model Generation, Wiley 2010, ISBN 978-0470876411
5. Christensen, C. — The Innovator's Dilemma, Harvard Business Review 2011, ISBN 978-1422196021
6. Thiel, P. — Zero to One, Crown Business 2014, ISBN 978-0804139298
7. Mehta, R. — Beyond the Hammer (Indian context), Penguin 2022

COURSE CODE:26IE2040E
COURSE NAME :Social Internship — Community Engagement
L-T-P-S :0-0-6-0 **PRE-REQ : NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply ethnographic observation and active listening to understand a host community's lived experience, social structure, and stated needs over a 4-week immersion.	PO 2, PO9	3
CO2	Apply needs-assessment frameworks (asset mapping, root-cause analysis, stakeholder mapping) to identify community priorities and existing resources or gaps.	PO2, PO4	3
CO3	Examine the structural drivers of community challenges including socioeconomic, policy, infrastructure, and cultural dimensions specific to the host context.	PO2, PO7	4
CO4	Apply participatory engagement methods (community meetings, focus groups, surveys, intergenerational dialogue) to co-design intervention concepts with community members..	PO3, PO9	3

31/1/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur Dist., Andhra Pradesh-522502.

CO5	Examine the role of NGOs, government schemes (PMGKAY, MGNREGA, NRLM, ASHA, anganwadi), Panchayati Raj institutions, and CSR programs operating in the host community.	PO6, PO7	4
CO6	Architect a sustainability-aware community impact report with verifiable observations, evidence-based recommendations, and reflective self-assessment.	PO4, PO10	5

Syllabus:

Module No.	Syllabus
M1	**Pre-Immersion Orientation** Mapped: **CO1, CO5** · 8 hrs • Internship objectives and ethical contracts • learning vs helping mindset • the 'voluntourism' critique • cultural humility • safeguarding and child-protection norms • informed consent • data and privacy in community work • physical safety planning • mental wellbeing during fieldwork • debrief journaling • emergency contacts • the host NGO / government / PRI partner introduction • understanding the funded program landscape (MGNREGA, NRLM, NHM, MDM, NSAP, etc.)
M2	**Field Immersion Week 1: Listening** Mapped: **CO1, CO2** · 24 hrs • Structured observation and field notes • daily reflective journal • meeting community elders, women's groups, youth groups • learning daily routines • mapping the geographic and social space • identifying key institutions (schools, health center, anganwadi, panchayat

21/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	office, water source) • identifying informal networks and trust holders(briefly)
M3	**Field Immersion Week 2-3: Co-Design** Mapped: **CO3, CO4** · 48 hrs • Asset mapping (what does the community already have?) • identifying problems vs identifying needs (different concepts) • root-cause analysis with the 5 Whys and fishbone methods • stakeholder mapping • convening focus group discussions • intergenerational dialogue • small intervention pilots (literacy support, health awareness, livelihood skill, environmental restoration, digital-literacy mini-camps)
M4	**Field Immersion Week 4: Hand-off** Mapped: CO5, CO6 · 16 hrs • Building local ownership of the intervention • identifying continuation pathways (linking to existing schemes, NGO follow-on, panchayat support) • knowledge transfer • closure conversations • honoring contributions • commitment statement to community • preparing impact-report data
M5	**Post-Immersion Synthesis** Mapped: **CO6** · 8 hrs • Writing the community impact report (background, methodology, observations, intervention, results, lessons, recommendations) • critical self-reflection on positionality and bias • presenting to peers and faculty • identifying what would be done differently • alignment with relevant SDGs • preparing for public/parents'-day showcase

Reference Books:

31/7/26
HoD-CSA

Dr. R. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

S.No	Title	Author(s)	Publisher	Edition / Year	ISBN
1	Whose Reality Counts? Putting the First Last	Robert Chambers	ITDG Publishing	1997	978-1853393860
2	Poor Economics	Abhijit V. Banerjee, Esther Duflo	PublicAffairs	2011	978-1610390934
3	Development as Freedom	Amartya Sen	Anchor Books	2000	978-0385720274
4	Whose Development Is It Anyway?	Nidhi Sharma	Routledge	2018	978-1138483989
5	Rural Development: Putting the Last First	Robert Chambers	Routledge	2014 (Reprint)	978-1138869974
6	ASER Annual Status of Education Report(Free Annual Report)	Pratham	ASER Centre	Annual Edition	—
7	Aspirational Districts Programme Reports(Free Reports)	NITI Aayog	Government of India	Latest Annual Reports	—

COURSE CODE: 26UC0009E
COURSE NAME :Ecology& Environment
L-T-P-S : 3-0-0-0 **PRE-REQ : NIL**

CO.	Course Outcome Description	PO	BTL

31/7/26
HoD-CSA
DR. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

No		MAPPING	
CO1	Apply ecological concepts (biotic-abiotic interactions, energy flow, biogeochemical cycles, food webs, succession) to analyze local and global ecosystems.	PO1, PO2	3
CO2	Analyze biodiversity at gene, species, and ecosystem levels and examine causes of biodiversity loss including habitat fragmentation, invasive species, climate change, and pollution.	PO2, PO7	4
CO3	Examine major environmental challenges (climate change, air and water pollution, soil degradation, plastic waste, e-waste) and the policy/technology responses to each.	PO2, PO5	4
CO4	Examine Indian environmental law (Environment Protection Act 1986, Air Act, Water Act, Wildlife Protection Act, NGT, NEPRA, EIA Notification 2020) and the role of CPCB/SPCBs.	PO6, PO7	4
CO5	Apply sustainability frameworks (UN SDG-13, ESG, circular economy, carbon footprint) to evaluate and reduce environmental impact of computing infrastructure and consumer choices.	PO5, PO7	3
CO6	Build a green audit / sustainability report for the campus or a chosen organization including baseline measurement, impact analysis, and intervention recommendations.	PO4, PO10	5

Syllabus:

Module No.	Syllabus
M1	**Ecology Foundations** Mapped: CO1 · 7 hrs • Ecosystem definition • biotic vs abiotic factors • trophic levels and food webs • energy flow vs nutrient cycling • biogeochemical cycles (carbon, nitrogen, phosphorus, water) • ecological succession • carrying capacity • r vs K selection • major Indian biomes (Himalayan, Western Ghats, Thar, Gangetic plains, coastal)

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation

M2	**Biodiversity & Loss** Mapped: **CO2** · 7 hrs • Genetic, species, ecosystem biodiversity • biodiversity hotspots (4 in India: Himalayas, Western Ghats, Indo-Burma, Sundaland) • habitat fragmentation • invasive species (lantana, water hyacinth, African catfish in India) • IUCN Red List • CITES • in-situ vs ex-situ conservation • protected area network in India (national parks, wildlife sanctuaries, biosphere reserves, conservation reserves, community reserves)
M3	**Environmental Challenges** Mapped: **CO3** · 8 hrs • Climate change basics (greenhouse effect, anthropogenic forcing, IPCC AR6 highlights) • air pollution (PM2.5, PM10, NOx, SOx, ozone, AQI methodology) • water pollution (BOD, COD, heavy metals, microplastics, eutrophication) • soil degradation (erosion, salinization, contamination) • solid waste (segregation, landfill, composting, incineration) • plastic and e-waste • noise pollution • light pollution • emerging concern: forever chemicals (PFAS) and microplastics
M4	**Indian Environmental Law & Policy** **Mapped: CO4 · 7 hrs** • Environment Protection Act 1986 • Water (Prevention and Control of Pollution) Act 1974 • Air Act 1981 • Wildlife Protection Act 1972 • Forest Conservation Act 1980 • NGT (National Green Tribunal) • EIA Notification 2006 and 2020 amendments • CPCB and SPCBs • coastal regulation zone • e-waste rules 2016 + 2022 amendment • plastic waste management rules 2016

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• battery waste management rules 2022 • India's Updated NDC and net-zero by 2070 commitment
M5	**Sustainability & Computing** Mapped: **CO5, CO6** · 7 hrs • UN SDGs (especially 6, 7, 11, 12, 13, 14, 15) • ESG investing • circular economy principles • carbon footprint methodology (Scope 1/2/3) • life cycle assessment • sustainable computing (green data centers, PUE, embodied carbon of devices, energy-efficient algorithms, AI's carbon footprint) • e-waste management • sustainable software engineering principles • the right-to-repair movement

Reference Books:

Sno	Title
1	Bharucha, E. — Textbook of Environmental Studies for Undergraduate Courses (3rd ed.), UGC, Universities Press 2020, ISBN 978-9389211238
2	Singh, J.S., Singh, S.P., Gupta, S.R. — Ecology, Environmental Science and Conservation (1st ed.), S. Chand 2017, ISBN 978-9385676302
3	Sharma, P.D. — Ecology and Environment (15th ed.), Rastogi 2018
4	IPCC AR6 Summary for Policymakers (free, primary source)
5	Diamond, J. — Collapse: How Societies Choose to Fail or Succeed, Penguin 2011
6	Carson, R. — Silent Spring (50th anniv. ed.), Houghton Mifflin Harcourt 2022 (foundational text)
7	Wilson, E.O. — The Diversity of Life, W.W. Norton 1999

31/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

COURSE CODE: 26CA2110E
COURSE NAME:ALGORITHM DESIGN AND ANALYSIS
L-T-P-S: 3-0-6-0 **PRE-REQ: DSJV**

CO. No	Course Outcome Description	PO-M APPING	BTL
CO1	Apply asymptotic analysis (Big-O, Big-Ω, Big-Θ), recurrence relations, and the master theorem to derive the time and space complexity of recursive and iterative algorithms	PO1, PO5	3
CO2	Apply divide-and-conquer (merge sort, quicksort, binary search variants, Strassen) to design recursive algorithms and analyze them via recurrences.	PO2, PO5	3
CO3	Analyze greedy algorithms (activity selection, fractional knapsack, Huffman, MST) to determine when local optima yield global optima — and when they don't.	PO2, PO5	4
CO4	Analyze dynamic programming patterns (memoization, tabulation, optimal substructure, overlapping subproblems) on classic problems (LCS, edit distance, 0/1 knapsack, matrix chain).	PO2, PO5	4
CO5	Evaluate advanced graph algorithms (Bellman-Ford, Floyd-Warshall, max flow, strongly connected components) to choose appropriate techniques for path, flow, and connectivity problems.	PO3, PO5	5
CO6	Evaluate computational tractability via P, NP, NP-completeness, and reductions; apply approximation, heuristics, and randomization where exact solutions are infeasible.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Asymptotic Analysis & Recurrences** **Mapped: CO1 (BTL-3) · 8 hrs · Project Stage 1 — Analysis toolkit** • Asymptotic notation revisited: O, Ω, Θ, o, ω; common pitfalls • Recurrence relations: substitution, recursion tree, master theorem

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• Amortized analysis: aggregate, accounting, potential methods • Best, average, worst case analysis • Probabilistic analysis basics; randomized algorithm motivation • Lower bounds: comparison-based sorting Ω(n log n) proof • The model of computation: RAM model assumptions • Recursion-to-iteration conversion patterns *Students prove the complexity of 5 recursive algorithms by hand and verify experimentally with empirical benchmarks. Math meets code.*
M2	**Module 2 — Divide & Conquer** **Mapped: CO2 (BTL-3) · 8 hrs · Project Stage 2 — D&C library** • Divide-and-conquer paradigm; when it fits • Merge sort: in-depth implementation, stability, external sorting preview • Quicksort: partition strategies, randomization, worst-case avoidance • Binary search variants: lower_bound, upper_bound, rotated arrays • Closest pair of points (the classic geometry D&C) • Strassen's matrix multiplication (intro to subcubic algorithms) • Karatsuba multiplication (cross-link MTCA M2) • Maximum subarray problem with D&C (then preview the O(n) Kadane) *AlgoLab gains 6 D&C implementations. Students measure them against the JDK equivalents on stress-test inputs.*
M3	**Module 3 — Greedy Algorithms** **Mapped: CO3 (BTL-4) · 8 hrs · Project Stage 3 — Greedy library** • Greedy paradigm; greedy choice property; matroid structure (intuition) • Activity selection problem with proof of correctness • Fractional knapsack (greedy works) vs 0/1 knapsack (greedy fails) • Huffman coding for prefix-free codes • Minimum spanning tree: Prim's and Kruskal's revisited (cross-link DSTR) • Job scheduling: shortest job first, earliest deadline first • Counterexamples: when greedy gives suboptimal results • Exchange argument as a proof technique *Students implement Huffman coding from scratch and use it to compress a real text file. Compression ratio measured*
M4	**Module 4 — Dynamic Programming** **Mapped: CO4 (BTL-4) · 10 hrs · Project Stage 4 — DP library** • DP paradigm: optimal substructure + overlapping subproblems • Memoization (top-down) vs tabulation (bottom-up); space optimization • Fibonacci as the canonical introduction • 0/1 knapsack: classic DP

31/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522 0

	• Longest Common Subsequence (LCS) • Edit distance (Levenshtein) — used in spell-check, diff, bioinformatics • Matrix chain multiplication • Coin change (count + minimum); rod cutting; longest increasing subsequence • State design as the DP skill that takes longest to develop *Students solve 8 classic DP problems and explain the state design for each in their own words. The hardest skill is naming the state.*
M5	**Module 5 — Advanced Graph Algorithms** **Mapped: CO5 (BTL-5) · 8 hrs · Project Stage 5 — Graph library** • Bellman-Ford: handles negative edges; detects negative cycles • Floyd-Warshall: all-pairs shortest paths via DP on graphs • Topological sort revisited; DAG shortest path • Strongly connected components: Kosaraju's, Tarjan's • Max flow / min cut: Ford-Fulkerson, Edmonds-Karp • Bipartite matching as a max flow application • Network flow modeling: assignment, transportation problems • Random walks and PageRank intro *Students model a real assignment problem (e.g., students-to-projects) as bipartite matching and solve it. Visible utility.*
M6	**Module 6 — NP-Completeness, Approximation & Randomization** **Mapped: CO6 (BTL-5) · 8 hrs · Project Stage 6 — Intractability survival** • P, NP, NP-hard, NP-complete: the classes and their relationships • Polynomial-time reductions: 3-SAT, vertex cover, TSP • Cook-Levin theorem (statement only) • Approximation algorithms: vertex cover 2-approx, set cover greedy log-approx • FPTAS, PTAS classes (intro) • Heuristics: simulated annealing, genetic algorithms (concept) • Randomized algorithms: random pivot quicksort, Karger's min-cut • Practical advice: when exact algorithms fail, what's the menu? *Students attempt vertex cover or TSP on a small instance with both an approximation and a heuristic, comparing solution quality and runtime.*

Reference Books:

SNO	TITLE	AUTHOR(S)	PUBLISHER	EDITION / YEAR	ISBN
1	*Introduction to Algorithms* (CLRS)	Cormen, Leiserson,	MIT Press	4th Ed, 2022	978-0262046305

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

		Rivest, Stein			
2	*The Algorithm Design Manual*	Steven S. Skiena	Springer	3rd Ed, 2020	978-3030542559
3	*Algorithm Design*	Jon Kleinberg, Éva Tardos	Pearson	1st Ed, 2005	978-0321295354
4	*Algorithms*	Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani	McGraw Hill	1st Ed, 2008	978-0073523408
5	*Cracking the Coding Interview*	Gayle Laakmann McDowell	CareerCup	6th Ed, 2015	978-0984782857
6	*Competitive Programmer's Handbook*	Antti Laaksonen	cses.fi (free)	2017	—
7	*Algorithms Illuminated* (4-volume series)	Tim Roughgarden	Soundlikeyourself Publishing	2017–2020	978-0999282908

COURSE CODE: 26CA2111E
COURSE NAME :COMPUTER NETWORKS & APPLICATIONS
L-T-P-S: 3-0-6-0 **PRE-REQ: OSCP**

CO. No	Course Outcome Description	PO-M APPING	BTL
CO1	Apply the TCP/IP and OSI layered models — physical, data link, network, transport, application — to trace how data moves end-to-end across the Internet.	PO1, PO5	3
CO2	Apply transport-layer protocols (TCP, UDP, QUIC) — connection setup, flow control, congestion control, reliability — to choose appropriate transports for application requirements.	PO1, PO5	3
CO3	Analyze application-layer protocols (HTTP/1.1, HTTP/2, HTTP/3, WebSocket, gRPC, DNS, SMTP) using packet captures (Wireshark) and design appropriate API protocols for given use cases.	PO2, PO5	4

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

CO4	Analyze socket programming patterns — blocking, non-blocking, event loops, select/poll/epoll — to implement client/server applications in Python and Java.	PO2, PO5	4
CO5	Evaluate distributed system fundamentals — RPC, message queues, service discovery, load balancing, caching, CDNs — to design scalable network services.	PO3, PO5	5
CO6	Evaluate network security fundamentals — TLS handshake, certificates, OAuth 2.0, OWASP API Security Top 10 — to harden network applications.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Layered Models & Network Foundations** **Mapped: CO1 (BTL-3) · 10 hrs · Project Stage 1 — Packet trace** • OSI 7-layer vs TCP/IP 4-layer; what each layer actually does • Protocol data units: bit → frame → packet → segment → message • Encapsulation and demultiplexing • End-to-end principle and the dumb network • Wireshark fundamentals: capture, display filters, follow streams • Tracing a single HTTP GET from your laptop to Google's servers — every layer • IPv4 addressing recap (cross-link ANVS) • The journey of a packet: ARP → switch → router → ISP → backbone *Students capture and annotate the full packet trace of a single HTTP request, with every layer labeled. The "aha" of seeing real bytes happens in Week 2.*
M2	**Module 2 — Transport Layer: TCP, UDP, QUIC** **Mapped: CO2 (BTL-3) · 10 hrs · Project Stage 2 — Transport choice** • UDP: connectionless, unreliable, fast; when it's the right choice • TCP: connection-oriented, reliable, ordered, congestion-controlled • TCP three-way handshake; sequence numbers and ACKs • Flow control: receive window, sliding window • Congestion control: Reno, CUBIC, BBR (intuitions) • Connection close: FIN, FIN-ACK, TIME_WAIT • QUIC: HTTP/3's transport; UDP-based but reliable; head-of-line blocking solved • Choosing TCP vs UDP vs QUIC by application requirements

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K L Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

	Students implement NetChat over both TCP and UDP, observing reliability differences when packets drop. Then add QUIC and measure latency improvement.
M3	**Module 3 — Application Layer Protocols** **Mapped: CO3 (BTL-4) · 12 hrs · Project Stage 3 — Multi-protocol API** • HTTP/1.1 in detail: methods, status codes, headers, persistent connections • HTTP/2: binary framing, multiplexing, server push, HPACK compression • HTTP/3: QUIC + HTTP — solving HOL blocking • WebSocket: upgrade handshake, full-duplex frames; chat use cases • gRPC: HTTP/2 + Protocol Buffers; streaming patterns • DNS: recursive vs iterative, A/AAAA/MX/CNAME records, DNS-over-HTTPS • SMTP / IMAP / POP3 (briefly) • Picking the right protocol for the right job *NetChat exposes itself via 3 protocols: REST/HTTP, WebSocket, gRPC. Students benchmark each on the same workload.*
M4	**Module 4 — Socket Programming & I/O Models** **Mapped: CO4 (BTL-4) · 10 hrs · Project Stage 4 — Custom server** • Berkeley sockets API: socket, bind, listen, accept, connect, send, recv • Blocking I/O: simple but doesn't scale • Thread-per-connection model: scales to ~10k connections • Non-blocking I/O + select/poll: the C10k problem • epoll (Linux), kqueue (BSD/macOS), IOCP (Windows) • Event loops: Python asyncio, Node.js, Java NIO/Netty • Connection pooling and keep-alive • The real difference between concurrency and parallelism in network servers *Students implement NetChat's server in 3 styles: thread-per-connection, asyncio, and (extra credit) raw epoll. Benchmark connections and throughput.*
M5	**Module 5 — Distributed Systems & Scalability** **Mapped: CO5 (BTL-5) · 10 hrs · Project Stage 5 — Scaled out** • Why scaling network services is hard • Load balancing: L4 (TCP) vs L7 (HTTP); round-robin, least-connections, consistent hashing • Reverse proxies: nginx, HAProxy, Envoy • Service discovery: DNS-based, Consul, Kubernetes Services (cross-link

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur Dist., Andhra Pradesh-522 5

	CLDO) • Caching: HTTP caching headers, CDN basics, Redis as application cache • Message queues: RabbitMQ, Kafka, NATS — when async beats sync • RPC frameworks: gRPC, Thrift; idempotency, retries, circuit breakers • The fallacies of distributed computing (revisit Peter Deutsch's classic 8) ***NetChat is scaled to 3 application instances behind nginx, with Redis as message store and RabbitMQ for fan-out. Real horizontal scaling.***
M6	**Module 6 — Network Security** **Mapped: CO6 (BTL-5) · 10 hrs · Project Stage 6 — Hardened** • The threat model: man-in-the-middle, replay, spoofing, eavesdropping • Symmetric vs asymmetric crypto (intuition) • TLS handshake: ClientHello, certificates, key exchange, session resumption • Certificates and PKI; Let's Encrypt; certificate transparency • HTTPS in practice: HSTS, SNI, TLS 1.3 vs 1.2 • OAuth 2.0 and OpenID Connect: the right way to delegate access • API security: rate limiting, input validation, OWASP API Security Top 10 • Network monitoring: log analysis, anomaly detection (preview of CYBF for security depth) ***NetChat is secured: TLS everywhere, OAuth login with Google or GitHub, rate limits, OWASP self-audit. Public URL with HTTPS cert from Let's Encrypt.***

Reference Books:

#	TITLE	AUTHOR(S)	PUBLISHER	EDITION / YEAR	ISBN
1	*Computer Networking: A Top-Down Approach*	James Kurose, Keith Ross	Pearson	8th Ed, 2021	978-0136681557
2	*Computer Networks*	Andrew S. Tanenbaum, David J. Wetherall	Pearson	6th Ed, 2021	978-0136764052

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-52250

3	*TCP/IP Illustrated, Volume 1: The Protocols*	Kevin R. Fall, W. Richard Stevens	Addison-Wesley	2nd Ed, 2011	978-0321336316
4	*UNIX Network Programming, Volume 1*	W. Richard Stevens, Bill Fenner, Andrew Rudoff	Addison-Wesley	3rd Ed, 2003	978-0131411555
5	*High Performance Browser Networking*	Ilya Grigorik	O'Reilly (free online)	1st Ed, 2013	978-1449344764
6	*Designing Data-Intensive Applications*	Martin Kleppmann	O'Reilly	1st Ed, 2017	978-1449373320
7	*Network Programming with Go*	Adam Woodbeck	No Starch Press	1st Ed, 2021	978-1718500884

COURSE CODE: 26CA2112E
COURSE NAME: APPLIED NETWORKS & VIRTUALIZATION
L-T-P-S: 3-0-3-0 **PRE-REQ: CNAP**

CO. No	Course Outcome Description	PO-M APPING	BTL
CO1	Apply the OSI / TCP-IP layered model and core protocols (Ethernet, ARP, IPv4/IPv6, ICMP, UDP, TCP) to interpret real packet captures and trace end-to-end communication.	PO1, PO5	3
CO2	Apply DNS, DHCP, HTTP/HTTPS, and TLS protocols to set up working network services and capture/decode their wire traffic.	PO1, PO5	3
CO3	Analyze socket programming (TCP and UDP) in Python or Java to build client-server applications and reason about reliability, ordering, and concurrency.	PO2, PO5	4
CO4	Analyze virtualization fundamentals (Type-1 vs Type-2 hypervisors, KVM/QEMU, VirtualBox), and Linux network namespaces to understand container networking foundations.	PO2, PO5	4

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO5	Evaluate network design choices (subnetting, NAT, routing, VLANs, basic SDN concepts) on a Mininet-emulated topology.	PO3, PO5	5
CO6	Evaluate network performance (latency, throughput, packet loss) and basic network security (firewalls, port scanning awareness, encrypted traffic) on a working topology.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Layered Models & Wire Reality** **Mapped: CO1 (BTL-3) · 12 hrs · Project Stage 1 — First captures** • OSI 7-layer model and the pragmatic TCP/IP stack • Ethernet frames, MAC addresses, ARP — what's on the wire • IPv4: addressing, subnetting, CIDR; IPv6 essentials • ICMP: ping and traceroute mechanics • UDP vs TCP: the trade-off • Wireshark hands-on: capture filters, display filters, follow stream • tcpdump on the command line • Reading hex bytes back to protocol structure ***Students capture a real DNS query, an HTTP GET, and a TCP three-way handshake in Wireshark, annotating every byte. Networks become concrete.***
M2	**Module 2 — Application Protocols** **Mapped: CO2 (BTL-3) · 10 hrs · Project Stage 2 — Services in NetLab** • DNS: resolvers, authoritative servers, record types, dig command • DHCP: discover-offer-request-acknowledge dance • HTTP/1.1: methods, headers, status codes; HTTP/2 and HTTP/3 overview • HTTPS and TLS handshake (decoded packet by packet) • Self-signed certs vs CA-signed; Let's Encrypt overview • SSH: keypair authentication, port forwarding, tunneling • SMTP, IMAP basics (rapid coverage) • Application-layer load balancing concepts ***Students set up a tiny BIND DNS server in a VM, point a client at it, and watch***

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	every step in Wireshark. TLS handshake decoded byte-by-byte.
M3	**Module 3 — Socket Programming** **Mapped: CO3 (BTL-4) · 10 hrs · Project Stage 3 — Custom protocol** • Socket API: socket(), bind(), listen(), accept(), connect() • Blocking vs non-blocking sockets; select / epoll briefly • TCP echo server in Python; iterative vs threaded • UDP server: stateless and lossy • Designing your own application protocol (length-prefix, delimiter-based) • Serialization: JSON vs MessagePack vs Protobuf • Concurrency: thread per connection vs event loop (preview OSCP) • Backpressure, slow consumers, timeout handling ***Students implement their own simple chat protocol over TCP. Multiple clients connect to one server. Their wire format is documented in a README spec.***
M4	**Module 4 — Virtualization & Network Namespaces** **Mapped: CO4 (BTL-4) · 10 hrs · Project Stage 4 — Virtual networks** • Hypervisor types: bare-metal (Type-1) vs hosted (Type-2) • KVM/QEMU on Linux; VirtualBox cross-platform • Vagrant for reproducible VM setup (preview) • Containers vs VMs revisited (cross-link CLDO M2) • Linux namespaces: PID, mount, network, user, IPC, UTS • Network namespaces hands-on: ipnetns commands • veth pairs, bridges, building a tiny container by hand • How Docker uses namespaces under the hood ***Students build a 3-node virtual network with `ipnetns` from scratch — no Docker, no VirtualBox. They see every wire.***

3/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicat
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

M5	**Module 5 — Network Topologies & Mininet** **Mapped: CO5 (BTL-5) · 10 hrs · Project Stage 5 — Full topology** • NAT: source NAT, destination NAT, port forwarding • Routing fundamentals: static routes, default gateway • Dynamic routing (concept only): RIP, OSPF intuition • VLANs and 802.1Q tagging • Mininet: Python-defined virtual networks with real Linux kernel paths • Building a topology: hosts, switches, links, link emulation (delay, loss) • SDN intro: OpenFlow concept, controller-switch separation • POX or Ryu controller (light touch) ***Students build a campus-style topology in Mininet: 4 hosts, 2 switches, 1 router. They run iperf3 between hosts and observe link emulation effects.***
M6	**Module 6 — Performance & Security Basics** **Mapped: CO6 (BTL-5) · 8 hrs · Project Stage 6 — Audit** • Network performance: bandwidth, latency, jitter, loss • iperf3 for bandwidth testing; ping/traceroute for latency • p50/p95/p99 latency; tail latency awareness • Firewalls: iptables / nftables basics • Port scanning awareness: nmap (defensive context only) • Encrypted traffic: distinguishing SNI, JA3 fingerprint awareness • VPNs (intro): WireGuard hands-on; how a tunnel actually works • Network monitoring: NetFlow, Prometheus blackbox exporter ***Students benchmark NetLab end-to-end, build a Grafana dashboard, then deliberately add 100ms link delay and observe the impact on application latency.***

Reference Books:

#	TITLE	AUTHOR(S)	PUBLISHER	EDITION / YEAR	ISBN
1	*Computer Networking: A Top-Down Approach*	James F. Kurose, Keith W. Ross	Pearson	8th Ed, 2021	978-0136681557

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

2	*Computer Networks*	Andrew S. Tanenbaum, David J. Wetherall	Pearson	6th Ed, 2021	978-0136764052
3	*TCP/IP Illustrated, Volume 1*	Kevin R. Fall, W. Richard Stevens	Addison-Wesley	2nd Ed, 2011	978-0321336316
4	*UNIX Network Programming, Volume 1*	W. Richard Stevens, Bill Fenner, Andrew Rudoff	Addison-Wesley	3rd Ed, 2003	978-0131411555
5	*Practical Packet Analysis (with Wireshark)*	Chris Sanders	No Starch Press	3rd Ed, 2017	978-1593278021
6	*SDN: Software Defined Networks*	Thomas D. Nadeau, Ken Gray	O'Reilly	1st Ed, 2013	978-1449342302

COURSE CODE: FLP
COURSE NAME: FOREIGN LANGUAGE PROFICIENCY
L-T-P-S: 3-0-0-0

PRE-REQ: NIL

CO. No	Course Outcome Description	PO-M APPING	BTL
CO1	Apply A1 (CEFR) listening competence in the chosen language to comprehend short conversations, basic instructions, and slow speech on familiar topics.	PO1, PO5	3
CO2	Apply A1 speaking competence to introduce oneself, ask and answer simple questions, describe people/places/objects, and engage in basic transactional conversations.	PO1, PO5	3
CO3	Apply A1 reading competence to comprehend short texts (signs, menus, simple emails, brief news headlines) using context and cognates.	PO1, PO5	3
CO4	Apply A1 writing competence to compose short messages, fill forms, and write 50-80 word personal descriptions.	PO1, PO5	3
CO5	Examine cultural conventions of the target language community including greetings, formality registers, body language, and key cultural references.	PO1, PO5	4

31/7/26

HoD-CSA

HOD-Computer Science and Applica
Koneru Lakshmaiah Education Found
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-522

CO6	Examine the global business and academic relevance of the chosen language including industries, multinational companies, and study/work pathways.	PO1, PO5	4

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Phonetics & Greetings** **Mapped: CO1, CO2 · 7 hrs** • Sound system of the chosen language (German, French, Japanese, Spanish, Mandarin — pick one) • pronunciation drills • greetings (formal vs informal) • introductions • basic numbers, days, months • telling time and date • classroom vocabulary ***Daily pronunciation drills; pair conversation practice; quizlet or Anki flashcards as required practice tool***
M2	**Module 2 — Basic Grammar & Common Verbs** **Mapped: CO2, CO4 · 8 hrs** • Articles and gender (where applicable) • subject pronouns • conjugation of high-frequency verbs (to be, to have, to do, to go, to want, to like) • present tense • basic sentence structure (SVO and language-specific orderings) • negation • question formation ***Grammar drills; sentence-construction exercises; written homework with instructor feedback***
M3	**Module 3 — Daily Life Vocabulary** **Mapped: CO1, CO3 · 8 hrs** • Family and relationships • food and dining • transportation and directions • shopping and money • weather and seasons • clothing

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• body parts and health • describing personality and appearance • colors and shapes ***Vocabulary drills; role-play scenarios (restaurant, market, asking directions); video comprehension exercises***
M4	**Module 4 — Reading, Writing & Tense Expansion** **Mapped: CO3, CO4 · 7 hrs** • Past tense (perfect or simple, depending on language) • future expressions • reading short texts (graded readers, simple news, social media posts) • writing short emails and texts • describing past events ('what I did yesterday') ***Graded reader assignments; weekly 80-word writing prompts; reading comprehension quizzes***
M5	**Module 5 — Culture & Career Pathways** **Mapped: CO5, CO6 · 6 hrs** • Cultural awareness for the target country/region (Germany/France/Japan/Spanish-speaking countries/China) • business etiquette • key holidays and traditions • industries where the language is in demand (German engineering, Japanese tech, French luxury, Spanish/Latin America markets, Mandarin tech) • study-abroad and work-visa pathways • certifications recognized for higher education and employment ***Cultural deep-dive sessions; guest speaker (alumni working/studying with the language); pathway mapping exercise***

Reference Books:

#	Reference
1	Schritte international Neu (German A1) — Hueber, latest edition
2	Le Nouveau Taxi! 1 (French A1) — Hachette

3/4/26

HoD-CSA

DR. K. BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

3	Genki I (Japanese) — The Japan Times, 3rd ed. 2020
4	Aula Internacional 1 (Spanish A1) — Difusión, latest
5	Integrated Chinese Vol 1 — Cheng & Tsui, 4th ed. 2017
6	Krashen, S. — Principles and Practice in Second Language Acquisition (free PDF, primary)
7	Meyer, E. — The Culture Map, PublicAffairs 2014, ISBN 978-1610392501

COURSE CODE: 26CA2213E
COURSE NAME: ADVANCED DATA STRUCTUES & ALGORITHMS
L-T-P-S: 3-0-3-0 **PRE-REQ: ALDA**

CO. No	Course Outcome Description	PO-M APPING	BTL
CO1	Apply advanced data structures (segment trees, Fenwick / BIT, sparse tables, persistent DS, treaps, link-cut trees intro) to range query and update problems.	PO1, PO5	3
CO2	Apply string algorithms (KMP, Z, Manacher, Aho-Corasick, suffix array, suffix automaton intro) to matching, palindrome, and substring problems.	PO1, PO5	3
CO3	Analyze advanced graph algorithms (LCA via binary lifting, heavy-light decomposition, Euler tour, 2-SAT, articulation points/bridges) to attack complex tree/graph problems.	PO2, PO5	4
CO4	Analyze advanced dynamic programming patterns (bitmask DP, digit DP, DP on trees, DP optimization with monotonic queues / Knuth / divide-and-conquer / convex hull trick intro).	PO2, PO5	4
CO5	Evaluate computational geometry primitives (cross product, convex hull, line intersection, sweep line) and number-theoretic algorithms (extended Euclidean, modular inverse, Chinese Remainder, Miller-Rabin) for problem-solving.	PO3, PO5	5

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-52250

CO6	Evaluate algorithm design intuition by solving curated competitive programming problems (Codeforces Div2 D/E and ICPC regional level) and explaining solutions in clear written form.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Advanced Data Structures** **Mapped: CO1 (BTL-3) · 8 hrs · Project Stage 1 — Range query toolkit** • Segment trees: point update + range query; lazy propagation • Fenwick (Binary Indexed) trees: O(n log n) build, O(log n) query/update • Sparse tables for static range minimum query (RMQ) • Square root decomposition (Mo's algorithm intro) • Disjoint set union (DSU) revisited: small-to-large, with rollback • Persistent data structures (intro): persistent segment tree • Treap and skip list (intro) • Link-cut trees and heavy-light decomposition (preview Module 3) ***Students implement segment tree with lazy propagation for range-add range-sum, and a Fenwick tree. Both pass classic OJ problems.***
M2	**Module 2 — String Algorithms** **Mapped: CO2 (BTL-3) · 8 hrs · Project Stage 2 — String toolkit** • Hashing: polynomial rolling hash with collision avoidance • KMP / failure function for pattern matching • Z-function and Z-algorithm • Manacher's algorithm for all-palindromes in O(n) • Aho-Corasick for multi-pattern matching • Suffix array construction (basic $O(n \log^2 n)$) • Longest Common Prefix (LCP) array • Suffix automaton (intro only) ***Students implement KMP, Z-function, and Manacher. AdvAlgoLab gains a string-algorithm module proven on substring/palindrome problems.***
M3	**Module 3 — Advanced Graph & Tree Algorithms** **Mapped: CO3 (BTL-4) · 10 hrs · Project Stage 3 — Tree/graph toolkit** • Lowest Common Ancestor (LCA): binary lifting, sparse table • Euler tour technique on trees • Heavy-light decomposition • Centroid decomposition (intro)

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• Articulation points and bridges (Tarjan) • Strongly connected components revisited (Tarjan, Kosaraju) • 2-SAT via implication graphs • Bipartite matching with Hopcroft-Karp (intro) ***Students implement LCA via binary lifting and use it to solve a competitive-programming distance problem on trees.***
M4	**Module 4 — Advanced Dynamic Programming** **Mapped: CO4 (BTL-4) · 10 hrs · Project Stage 4 — DP techniques** • Bitmask DP: TSP via subset DP, assignment problems • Digit DP: counting numbers with property X up to N • DP on trees: subtree DP, rerooting technique • DP optimization: deque/monotonic queue (sliding-window minimum) • Convex hull trick (intro) • Knuth's optimization (intro for quadrangle inequality) • Profile DP (broken profile / connected component DP) • Recognizing DP problems by structure ***Students solve TSP for $n \leq 18$ with bitmask DP; they solve a digit DP problem from a real Codeforces contest.***
M5	**Module 5 — Computational Geometry & Number Theory** **Mapped: CO5 (BTL-5) · 8 hrs · Project Stage 5 — Math toolkit** • Geometry primitives: cross product, dot product, signed area • Line-line intersection; segment-segment intersection • Convex hull: Graham scan, Andrew's monotone chain • Sweep line algorithms (intro) • Number theory: GCD, extended Euclidean, modular inverse • Chinese Remainder Theorem • Miller-Rabin primality test (probabilistic) • Pollard's rho for factorization (intro) ***Students implement convex hull and use it to solve the famous "closest gift wrap" problem. They implement modular inverse and use it in counting-mod-prime problems.***
M6	**Module 6 — Problem-Solving Mastery & Mock Contests** **Mapped: CO6 (BTL-5) · 8 hrs · Project Stage 6 — Contest portfolio** • Mapping problem patterns to algorithms

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-52250

- Reading and editorialising contest problems
- Time management in 2- and 3-hour contests
- Codeforces Div2 D/E: target tier for Y3 students
- ICPC regional preparation (for top 10%)
- Writing clear problem editorials
- Mock interviews — algorithmic phone screens
- System design intersection (intro): when to bring in advanced DS in interviews

Students participate in 4 KLEF-internal mock contests and 4 external Codeforces rounds. Each problem solved is editorialized in their AdvAlgoLab repo.

Reference Books:

#	TITLE	AUTHOR(S)	PUBLISHER	EDITION / YEAR	ISBN
1	*Competitive Programming 4* (vols 1 & 2)	Steven Halim, Felix Halim	self-published	2020	978-9811292491
2	*Competitive Programmer's Handbook* (free)	Antti Laaksonen	cses.fi	2017	—
3	*Introduction to Algorithms* (CLRS)	Cormen, Leiserson, Rivest, Stein	MIT Press	4th Ed, 2022	978-0262046305
4	*The Algorithm Design Manual*	Steven S. Skiena	Springer	3rd Ed, 2020	978-3030542559
5	*Algorithms*	Robert Sedgewick, Kevin Wayne	Addison-Wesley	4th Ed, 2011	978-0321573513
6	*Cracking the Coding Interview*	Gayle Laakmann McDowell	CareerCup	6th Ed, 2015	978-0984782857
7	*Elements of Programming Interviews*	Aziz, Lee, Prakash	EPI Books	1st Ed, 2015 (Java/Python/C++ versions)	978-1517671273

COURSE CODE: 26CA2214E
COURSE NAME: CLOUD COMPUTING & DEVOPS

31/1/26

HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

L-T-P-S: 3-0-6-0 **PRE-REQ: CNAP**

CO. No	Course Outcome Description	PO-M APPING	BTL
CO1	Apply core cloud computing service models (IaaS, PaaS, SaaS) and AWS primitives (EC2, S3, IAM, VPC, RDS) to provision basic cloud infrastructure.	PO1, PO5	3
CO2	Apply containerization with Docker — images, containers, multi-stage builds, networking, volumes — to package applications consistently across environments.	PO1, PO5	3
CO3	Analyze Kubernetes architecture and primitives (pods, deployments, services, ingress, ConfigMap, secrets) to orchestrate containerized applications at scale.	PO2, PO5	4
CO4	Analyze Infrastructure-as-Code patterns using Terraform to declaratively provision and manage cloud resources reproducibly.	PO2, PO5	4
CO5	Evaluate CI/CD pipelines using GitHub Actions or Jenkins to automate the build, test, and deployment lifecycle of applications.	PO3, PO5	5
CO6	Evaluate cloud observability and reliability practices — Prometheus metrics, Grafana dashboards, log aggregation, alerting, blue-green and canary deployments — for production-grade operations.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Cloud Fundamentals & AWS Core** **Mapped: CO1 (BTL-3) · 12 hrs · Project Stage 1 — Manual cloud deploy** • Cloud computing: definitions, NIST model, deployment models • Service models: IaaS, PaaS, SaaS, FaaS — when each fits • AWS account setup, billing alarms, IAM users and policies • Compute: EC2 — instance types, AMIs, security groups, key pairs • Storage: S3 (objects), EBS (block), EFS (file) • Networking: VPC, subnets, route tables, NAT gateways, IGWs • Managed databases: RDS PostgreSQL — backup, restore, replicas • The shared responsibility model ***Students manually deploy TaskMaster to a single EC2 instance with RDS backend***

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	in Week 2. They feel the friction of "clicking through the console" — which makes the IaC pivot in Module 4 motivating.
M2	**Module 2 — Containers & Docker** **Mapped: CO2 (BTL-3) · 12 hrs · Project Stage 2 — Containerized app** • The container revolution: containers vs VMs vs processes • Linux namespaces and cgroups (the kernel features that make containers possible) • Docker basics: images, layers, containers, registries • Dockerfile: FROM, COPY, RUN, CMD, ENTRYPOINT; build context • Multi-stage builds for minimal production images • Container networking: bridge, host, port mapping. Docker Compose • Container volumes for persistence • Docker Hub, GitHub Container Registry, ECR ***TaskMaster is containerized. Multi-stage build produces a 200MB image (vs 800MB naive). Docker Compose runs the full stack (app + Postgres + Redis) locally.***
M3	**Module 3 — Kubernetes** **Mapped: CO3 (BTL-4) · 12 hrs · Project Stage 3 — k8s deployment** • Why Kubernetes: scheduling, self-healing, rolling updates • k8s architecture: control plane, kubelet, kube-proxy, etcd • Core objects: Pod, Deployment, ReplicaSet, Service, Ingress • Configuration: ConfigMap, Secret; environment-specific values • Storage: PersistentVolume, PersistentVolumeClaim, StorageClass • Health probes: liveness, readiness, startup • kubectl essentials and YAML manifests • Local k8s: minikube or kind; managed k8s: EKS ***Students run TaskMaster on minikube locally, then deploy to a small EKS cluster. The transition from Docker Compose to k8s manifests is concrete.***
M4	**Module 4 — Infrastructure as Code with Terraform** **Mapped: CO4 (BTL-4) · 12 hrs · Project Stage 4 — Provisioned IaC** • The case for IaC: reproducibility, drift, code review • Terraform: providers, resources, data sources, modules • State: local vs remote (S3 + DynamoDB locking) • Variables, outputs, locals; workspaces for multi-environment

31/7/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur Dist., Andhra Pradesh-522 302

	• Terraform plan/apply/destroy lifecycle • Idempotency and the danger of manual changes after IaC • Importing existing resources; refactoring with terraform state mv • Module composition; the public Terraform Registry ***Students recreate Module 1's manual deploy entirely as Terraform. They tear down everything and recreate it — proving the infrastructure is truly code.***
M5	**Module 5 — CI/CD Pipelines** **Mapped: CO5 (BTL-5) · 12 hrs · Project Stage 5 — Automated pipeline** • CI vs CD vs Continuous Deployment • GitHub Actions: workflows, jobs, steps, runners, secrets • Reusable workflows and composite actions • Build pipeline: checkout, setup-jdk, mvn test, build Docker image, push • Deploy pipeline: terraform apply, kubectl apply, health check • Branch strategies: trunk-based, GitFlow, GitHub Flow • Pull request workflow with required CI checks • Jenkins as an alternative (briefly): pipeline as code with Jenkinsfile ***Every push to main automatically tests, builds, and deploys TaskMaster to staging. Production deploys are manually triggered with approval.***
M6	**Module 6 — Observability & Reliability** **Mapped: CO6 (BTL-5) · 12 hrs · Project Stage 6 — Production-grade ops** • The three pillars of observability: logs, metrics, traces • Prometheus: scraping, PromQL basics, recording rules • Grafana: dashboards, alerts, data sources • Log aggregation: ELK / Loki; structured logs review • Distributed tracing intro: OpenTelemetry, Jaeger • Alerting: Alertmanager; SLOs, error budgets, alert fatigue • Deployment strategies: rolling, blue-green, canary • Cost monitoring; FinOps fundamentals; the AWS bill that almost killed your startup ***TaskMaster has full observability. Grafana dashboard shows request rates, latency percentiles, error rates. An alert fires when p95 > 500ms and the student demonstrates the on-call workflow.***

31/7/25

HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-52250

Reference Books:

#	TITLE	AUTHOR(S)	PUBLISHER	EDITION / YEAR	ISBN
1	*The DevOps Handbook*	Kim, Humble, Debois, Willis, Forsgren	IT Revolution	2nd Ed, 2021	978-1950508402
2	*Site Reliability Engineering*	Beyer, Jones, Petoff, Murphy (Google)	O'Reilly	1st Ed, 2016 (free online)	978-1491929124
3	*Kubernetes: Up and Running*	Hightower, Burns, Beda	O'Reilly	3rd Ed, 2022	978-1098110208
4	*Terraform: Up &Running*	Yevgeniy Brikman	O'Reilly	3rd Ed, 2022	978-1098116743
5	*Docker Deep Dive*	Nigel Poulton	self-published	2023 Ed	978-1916585256
6	*AWS Certified Solutions Architect Study Guide*	Ben Piper, David Clinton	Sybex / Wiley	4th Ed, 2022	978-1119982623
7	*Continuous Delivery*	Jez Humble, David Farley	Addison-Wesley	1st Ed, 2010	978-0321601919

COURSE CODE: 26CA2215E
COURSE NAME: DEEP LEARNING FOUNDATIONS
L-T-P-S: 3-0-3-0 **PRE-REQ: AAML**

CO. No	Course Outcome Description	PG-M AI 1N6	BTL
CO1	Apply foundational neural network concepts (perceptron, MLP, forward pass, backpropagation, gradient descent) by implementing them from scratch in NumPy and verifying against PyTorch.	PO1, PO5	3
CO2	Apply PyTorch fundamentals (tensors, autograd, nn.Module, optimizers, DataLoader) to train a feedforward network on a real dataset.	PO1, PO5	3

31/7/26
HoD-CSA

DR. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

CO3	Analyze convolutional neural networks (convolution, pooling, feature maps, classic architectures: LeNet → AlexNet → ResNet) for image classification tasks.	PO2, PO5	4
CO4	Analyze sequence models (RNN, LSTM, GRU) and their limitations to motivate the Transformer architecture.	PO2, PO5	4
CO5	Evaluate the Transformer architecture (self-attention, multi-head attention, positional encoding) and apply transfer learning with HuggingFace pretrained models.	PO3, PO5	5
CO6	Evaluate training-time best practices (regularization, dropout, batch normalization, learning rate scheduling, mixed precision) and deploy a trained deep learning model as a usable service.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Neural Networks & Backpropagation from Scratch** **Mapped: CO1 (BTL-3) · 12 hrs · Project Stage 1 — Backprop notebook** • Perceptron and the geometry of linear classifiers • Multi-layer perceptron (MLP); the universal approximation theorem (statement only) • Activation functions: sigmoid, tanh, ReLU, GELU; vanishing-gradient intuition • Forward pass mathematics (cross-link MTCA M2: matrix multiplication) • Loss functions: MSE for regression, cross-entropy for classification • Backpropagation: chain rule applied through the network (cross-link MTCA M3) • Gradient descent: SGD, mini-batch, momentum; intuition for learning rate • Hand-derive gradients for a 2-layer MLP; implement in pure NumPy ***Students implement a 2-layer MLP from scratch in NumPy, train on MNIST. Then re-implement in PyTorch and verify identical loss curves. "It's not magic; it's calculus."***
M2	**Module 2 — PyTorch Fundamentals** **Mapped: CO2 (BTL-3) · 10 hrs · Project Stage 2 — PyTorch baseline** • PyTorch tensors: dtype, device, broadcasting (mirror NumPy) • autograd: requires_grad, .backward(), .grad; computation graph • nn.Module: parameters, modules, sub-modules; the Module pattern • Common layers: Linear, ReLU, Dropout, BatchNorm

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• Loss functions: nn.CrossEntropyLoss, nn.MSELoss • Optimizers: SGD, Adam, AdamW • Dataset and DataLoader; collate functions • Training loop scaffold; evaluation loop; reproducibility (seed, deterministic) ***Students rewrite the Module 1 model in PyTorch with proper Module/Dataset/DataLoader/optimizer structure. Train on Fashion-MNIST. Tensorboard logs.***
M3	**Module 3 — Convolutional Neural Networks** **Mapped: CO3 (BTL-4) · 12 hrs · Project Stage 3 — Image classifier** • Why convolutions: spatial locality and translation invariance • Convolution operation; kernel, stride, padding • Pooling: max, average; downsampling and receptive field • Classic architectures: LeNet, AlexNet, VGG (intuition) • ResNet: skip connections solve the vanishing gradient problem at depth • Modern variants: EfficientNet, ConvNeXt (briefly) • Data augmentation: flips, rotations, RandAugment, MixUp, CutMix • Transfer learning: feature extraction vs fine-tuning ***Students build a CNN from scratch on CIFAR-10, then fine-tune a pretrained ResNet-18 on a custom dataset (15+ classes, > 5000 images). Compare accuracy.***
M4	**Module 4 — Sequence Models & Recurrent Networks** **Mapped: CO4 (BTL-4) · 8 hrs · Project Stage 4 — Text classifier** • Why sequences need different architectures • Vanilla RNN: hidden state, recurrence, BPTT • The vanishing-gradient problem in RNNs • LSTM and GRU: gates and the long-memory problem • Encoder-decoder seq2seq architecture • Attention mechanism (additive, multiplicative) - the precursor to Transformers • Word embeddings: Word2Vec, GloVe, FastText (briefly) • Limitations of RNNs that motivated Transformers ***Students train an LSTM text classifier on a sentiment dataset (e.g., IMDb). Note training time and per-epoch cost. This sets up the "why Transformers" punchline in Module 5***
M5	**Module 5 — Transformers & Transfer Learning** **Mapped: CO5 (BTL-5) · 12 hrs · Project Stage 5 — Fine-tuned BERT** • Self-attention: query, key, value; the dot-product attention formula • Multi-head attention; why multiple heads help • Positional encoding (sinusoidal, learned, RoPE briefly) • The encoder-only (BERT), decoder-only (GPT), encoder-decoder (T5) trio

3/1/26
HoD-CSA

DR. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• Pretraining objectives: MLM (BERT), CLM (GPT), span corruption (T5) • Tokenization revisited: WordPiece, BPE, SentencePiece • HuggingFace Transformers: AutoTokenizer, AutoModel, AutoModelForSequenceClassification • Fine-tuning vs feature extraction; LoRA preview ***Students fine-tune DistilBERT for a multi-class text classification task. Compare to the LSTM from Module 4 — typically 10x less training time, much higher accuracy.***
M6	**Module 6 — Training Best Practices & Deployment** **Mapped: CO6 (BTL-5) · 12 hrs · Project Stage 6 — Deployed model** • Regularization: L2, dropout, early stopping, label smoothing • Batch normalization, layer normalization; when each fits • Learning rate scheduling: cosine annealing, warm-up, OneCycle • Mixed precision training (AMP) for 2x speedup • Gradient accumulation for memory-constrained training • Hyperparameter tuning: Optuna, Ray Tune (intro) • Model export: TorchScript, ONNX • Deployment: ONNX Runtime, FastAPI service, ONNX → web (briefly) ***Best deep learning model from M3 or M5 is exported, deployed, and served via FastAPI. Public URL with Streamlit demo on top.***

Reference Books:

#	TITLE	AUTHOR(S)	PUBLISHER	EDITION / YEAR	ISBN
1	*Deep Learning*	Ian Goodfellow, Yoshua Bengio, Aaron Courville	MIT Press	1st Ed, 2016 (free online)	978-0262035613
2	*Dive into Deep Learning* (D2L)	Zhang, Lipton, Li, Smola	d2l.ai (free)	2024	—
3	*Deep Learning with PyTorch*	Stevens, Antiga, Viehmann	Manning	1st Ed, 2020	978-1617295263
4	*Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*	Aurélien Géron	O'Reilly	3rd Ed, 2022	978-1098125974

31/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

5	*Programming PyTorch for Deep Learning*	Ian Pointer	O'Reilly	1st Ed, 2019	978-1492045359
6	*Natural Language Processing with Transformers*	Tunstall, von Werra, Wolf	O'Reilly	Revised Ed, 2022	978-1098136796
7	*Neural Networks and Deep Learning*	Michael Nielsen	online (free)	2015 (revised)	—

COURSE CODE: 26CA2216E
COURSE NAME: CYBERSECURITY FOUNDATIONS
L-T-P-S: 3-0-3-0 **PRE-REQ: CNAP**

CO. No	Course Outcome Description	PO-11 APP/ANG	BTL
CO1	Apply the CIA triad, AAA model, and threat-modeling frameworks (STRIDE, attack trees) to identify security goals and risks for an application.	PO1, PO5	3
CO2	Apply cryptographic primitives (symmetric: AES; asymmetric: RSA, ECC; hashes: SHA-2, SHA-3, bcrypt; HMAC; digital signatures) correctly using standard libraries.	PO1, PO5	3
CO3	Analyze network security mechanisms (TLS handshake, firewalls, IDS/IPS basics, VPNs) and detect basic attacks via packet capture analysis.	PO2, PO5	4
CO4	Analyze the OWASP Top 10 web vulnerabilities (injection, broken auth, XSS, CSRF, SSRF, IDOR, broken access control) by exploiting and remediating them on a deliberately vulnerable app.	PO2, PO5	4
CO5	Evaluate authentication and identity management designs (password storage, MFA, OAuth 2.0, OIDC, session management) for production applications.	PO3, PO5	5
CO6	Evaluate secure-coding practices, conduct a structured security audit using OWASP ZAP and manual review, and produce a remediation report for a real application (TaskMaster).	PO3, PO5	5

Syllabus:

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Module No.	Syllabus
M1	**Module 1 — Security Foundations & Threat Modeling** **Mapped: CO1 (BTL-3) · 8 hrs · Project Stage 1 — Threat model** • CIA triad: confidentiality, integrity, availability • AAA: authentication, authorization, accounting • Threat actor categorization: script kiddie, criminal, nation-state • Attack lifecycle: recon, weaponization, exploitation, lateral movement, exfiltration • STRIDE threat model framework • Attack trees and DREAD scoring (intro) • Risk = likelihood × impact • Defense in depth; least privilege; zero trust intro ***Students apply STRIDE to TaskMaster, producing a 1-page threat model identifying 8+ realistic threats. Foundation for the rest of the course.***
M2	**Module 2 — Cryptography (Applied)** **Mapped: CO2 (BTL-3) · 10 hrs · Project Stage 2 — Crypto correctness** • Symmetric crypto: AES; modes (ECB - bad, CBC, GCM - good); IV usage • Asymmetric crypto: RSA, ECC; key sizes; performance • Hashing: SHA-2, SHA-3 (use these); MD5/SHA-1 (don't use these) • Password hashing: bcrypt, scrypt, Argon2 (NOT plain SHA) • HMAC for authenticated integrity • Digital signatures: how they prove authorship • Key derivation functions (PBKDF2, HKDF) • Common cryptographic mistakes: ECB mode, fixed IV, custom crypto, MD5 for passwords ***Students implement: AES-GCM encrypt/decrypt; bcrypt password hashing; HMAC verification. Then they decrypt an ECB-encrypted image and clearly see the pattern leak (the famous Tux image example)***
M3	**Module 3 — Network Security** **Mapped: CO3 (BTL-4) · 10 hrs · Project Stage 3 — Wire-level audit** • TLS handshake deep dive (cross-link ANVS M2) • Certificate validation; PKI; mistrust scenarios • HSTS, certificate pinning • Firewalls: stateful inspection; iptables hardening (cross-link ANVS) • IDS/IPS: signature vs anomaly detection (intro) • VPNs deep dive: WireGuard, OpenVPN; what they protect against • Wireless security: WPA2/WPA3 essentials • Common network attacks: ARP spoofing, MITM, DNS poisoning (defensive context)

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	Students harden TaskMaster's network exposure with iptables rules, set up HTTPS with HSTS, and capture a TLS handshake to verify cert chain.
M4	**Module 4 — OWASP Top 10 (Hands-On)** **Mapped: CO4 (BTL-4) · 12 hrs · Project Stage 4 — Vulnerability hunting** • OWASP Top 10 — current edition • Injection: SQL injection (with examples) • Broken authentication and session management • Cross-site Scripting (XSS): reflected, stored, DOM-based • Cross-site Request Forgery (CSRF): SameSite cookies + CSRF tokens • Server-Side Request Forgery (SSRF): the cloud-metadata pivot • IDOR (Insecure Direct Object References) and broken access control • Security misconfigurations: defaults, debug endpoints • Vulnerable / outdated components; SBOM concepts ***Students attack a deliberately vulnerable application (DVWA, OWASP Juice Shop, or HackTheBox starter machine) and document successful exploitation of all 10 categories.***
M5	**Module 5 — Authentication & Identity** **Mapped: CO5 (BTL-5) · 8 hrs · Project Stage 5 — Auth hardening** • Password policies: NIST 800-63B (length, no rotation, no composition) • Multi-factor authentication: TOTP, FIDO2/WebAuthn, SMS (and why SMS is weakest) • OAuth 2.0 flows: authorization code (with PKCE), client credentials • OpenID Connect basics on top of OAuth • Session management: cookies, JWTs, refresh tokens revisited • Account lockout vs rate limiting; CAPTCHA strategies • Password reset flow security (token entropy, expiration) • Common attack patterns: credential stuffing, brute force, phishing ***Students harden TaskMaster auth: bcrypt cost factor verification, TOTP MFA optional, rate limiting login attempts, secure password reset flow.***
M6	**Module 6 — Security Audit & Secure Coding** **Mapped: CO6 (BTL-5) · 12 hrs · Project Stage 6 — Final SecAudit** • Secure coding principles: validate input, encode output, fail securely • Static Application Security Testing (SAST): SonarQube, Semgrep • Dynamic Application Security Testing (DAST): OWASP ZAP, Burp Suite Community • Software Composition Analysis (SCA): Dependabot, OWASP Dependency-Check, Snyk • Security audit methodology: scope, recon, scan, manual review, exploit

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

	attempt, report • CVE / CVSS scoring; vulnerability disclosure • Logging for security: what to log, what NOT to log (PII, passwords) • Security headers: CSP, HSTS, X-Frame-Options, X-Content-Type-Options ***Students run OWASP ZAP against their own TaskMaster, find issues, fix them, re-run. Final SecAudit report follows industry format with CVSS scoring.***

Reference Books:

#	TITLE	AUTHOR(S)	PUBLISHER	EDITION / YEAR	ISBN
1	*Computer Security: Principles and Practice*	William Stallings, Lawrie Brown	Pearson	4th Ed, 2017	978-0134794105
2	*The Web Application Hacker's Handbook*	Dafydd Stuttard, Marcus Pinto	Wiley	2nd Ed, 2011	978-1118026472
3	*Cryptography and Network Security*	William Stallings	Pearson	8th Ed, 2022	978-9356062993
4	*Real-World Cryptography*	David Wong	Manning	1st Ed, 2021	978-1617296710
5	*The Tangled Web*	Michał Zalewski	No Starch Press	1st Ed, 2011	978-1593273880
6	*OWASP Testing Guide v4*	OWASP Foundation	OWASP (free)	2020	—
7	*Security Engineering*	Ross Anderson	Wiley	3rd Ed, 2020 (free online)	978-1119642787

COURSE CODE: 26CA2317E
COURSE NAME: GENERATIVE AI FOUNDATIONS
L-T-P-S: 3-0-3-0 **PRE-REQ: DLFO**

CO.	Course Outcome Description	PO's	BTL

31/7/26
HoD-CSA

Dr. [illegible] GAVARI
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

No		MAPPING	
CO1	Apply foundational generative-AI concepts — language modeling, autoregressive generation, tokenization, embeddings — to use modern LLMs (OpenAI, Anthropic, open-source) via APIs and SDKs.	PO1, PO5	3
CO2	Apply prompt engineering patterns (zero-shot, few-shot, chain-of-thought, ReAct, structured output) to elicit reliable, structured responses from LLMs.	PO1, PO5	3
CO3	Analyze the Retrieval-Augmented Generation (RAG) architecture — chunking, embeddings, vector stores, retrieval, re-ranking, generation — to ground LLM responses in private documents.	PO2, PO5	4
CO4	Analyze fine-tuning approaches (full fine-tuning, LoRA, QLoRA, instruction tuning) to specialize a base LLM for a domain task with limited compute.	PO2, PO5	4
CO5	Evaluate agentic AI patterns — tool use, function calling, multi-step planning, reflection, multi-agent orchestration — to build LLMs that take real actions, not just emit text.	PO3, PO5	5
CO6	Evaluate generative AI systems on cost, latency, faithfulness, hallucination rates, safety, and bias to make informed deployment decisions.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	LLM Foundations & Tokenization Mapped: CO1 (BTL-3) · 12 hrs · Project Stage 1 — First LLM call • What an LLM is — autoregressive next-token prediction at scale • Brief Transformer intuition • Pre-training vs alignment (RLHF, DPO) • Tokenization (BPE, SentencePiece) • Embeddings & cosine similarity • Open vs closed models • Pricing models • API basics (OpenAI, Anthropic, HuggingFace, Ollama) Students build a chatbot UI comparing outputs from two LLM providers.
M2	Prompt Engineering Mapped: CO2 (BTL-3) · 12 hrs · Project Stage 2 — Prompted assistant • System/user/assistant prompts • Zero-shot, few-shot, CoT, ReAct

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• Structured output (JSON, Pydantic) • Self-critique & prompt defense • Token optimization Students implement prompt engineering for quiz generation.
M3	Retrieval-Augmented Generation (RAG) Mapped: CO3 (BTL-4) · 12 hrs · Project Stage 3 — Notes-grounded answers • Hallucination & grounding • Chunking, embeddings, vector DBs • Retrieval, reranking, citations Students upload PDFs and answer questions with citations.
M4	Fine-Tuning & Adaptation Mapped: CO4 (BTL-4) · 10 hrs · Project Stage 4 — Domain adaptation • Fine-tuning vs Prompting vs RAG • LoRA, QLoRA • Instruction tuning • SFT & DPO overview Students fine-tune Phi-3-mini/Llama-3.2-3B using QLoRA.
M5	Agentic AI & Tool Use Mapped: CO5 (BTL-5) · 12 hrs · Project Stage 5 — Agentic StudyMate • Function calling • Planning & memory • Multi-agent orchestration • LangChain, LangGraph, CrewAI Students add scheduling and search tools.
M6	Evaluation, Cost, Safety & Deployment Mapped: CO6 (BTL-5) · 12 hrs · Project Stage 6 — Production AI • Ragas, DeepEval, Promptfoo • Faithfulness & latency • Safety, bias, observability • Deployment (HF Spaces, Streamlit Cloud, AWS Bedrock) Students evaluate and deploy StudyMate.

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	Hands-On Large	Jay Alammar,	O'Reilly	1st Ed,	978-

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	Language Models	Maarten Grootendorst		2024	1098150969
2	Generative Deep Learning	David Foster	O'Reilly	2nd Ed, 2023	978-1098134181
3	Natural Language Processing with Transformers	Tunstall, von Werra, Wolf	O'Reilly	Revised Ed, 2022	978-1098136796
4	Building LLMs for Production	Louis-François Bouchard, Louie Peters	Towards AI	1st Ed, 2024	979-8324731472
5	AI Engineering	Chip Huyen	O'Reilly	1st Ed, 2024	978-1098166304
6	Designing Machine Learning Systems	Chip Huyen	O'Reilly	1st Ed, 2022	978-1098107963
7	Prompt Engineering for LLMs	John Berryman, Albert Ziegler	O'Reilly	1st Ed, 2024	978-1098156152

COURSE CODE: 26CA2318E
COURSE NAME: MACHINE LEARNING OPERATIONS (MLOps)
L-T-P-S: 3-0-6-0 **PRE-REQ: DLFO**

CO. No	Course Outcome Description	PO – M APPING	BTL
CO1	Apply ML lifecycle stages (data collection, training, evaluation, deployment, monitoring) and reproducibility tools (MLflow, DVC) to manage machine learning experiments and artifacts.	PO1, PO5	3
CO2	Apply data versioning, feature stores, and data validation frameworks (Great Expectations / Pandera) to build trustworthy machine learning data pipelines.	PO1, PO5	3
CO3	Analyze machine learning deployment patterns (REST API, batch, streaming, edge) and model serving frameworks	PO2, PO5	4

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-52250

	(BentoML, Seldon, KServe, TorchServe) to select appropriate serving architectures for given service-level agreements (SLAs).		
CO4	Analyze CI/CD/CT (Continuous Integration, Continuous Deployment, Continuous Training) pipelines for machine learning using GitHub Actions, Argo Workflows, and Kubeflow.	PO2, PO5	4
CO5	Evaluate machine learning monitoring practices—data drift, concept drift, prediction drift, and model staleness—and design alerting and automated retraining strategies.	PO3, PO5	5
CO6	Evaluate end-to-end MLOps maturity using Google's MLOps Levels (0/1/2) and Microsoft's MLOps Maturity Model for real-world ML systems and propose improvements.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	• The ML lifecycle vs the software lifecycle: what's different and why • The reproducibility crisis in ML; "works on my machine" disasters • MLflow: tracking, model registry, projects, recipes • Weights & Biases as an alternative • Hyperparameter sweeps and experiment comparison • Model registry: staging vs production, model versioning • Notebooks vs scripts; when to graduate from Jupyter • Conda/uv environment pinning; the dependency hell problem
M2	• Data versioning with DVC; large-file handling in git • Data lake vs data warehouse vs feature store (intuition) • Feature stores: Feast (open source), why they matter for online + batch parity • Data validation: Great Expectations, Pandera schemas • Schema evolution: backward / forward / full compatibility • Pipeline orchestration intro: Airflow, Prefect, Dagster (compare) • Streaming data basics: Kafka, Kinesis, Pub/Sub • The training-serving skew problem
M3	• Deployment patterns: online (REST/gRPC), batch, streaming, edge • BentoML: package + serve any model; bentos + yatai • Seldon Core, KServe — k8s-native serving • TorchServe, TFServing for framework-specific deployment • FastAPI as the simple route (revisit AIML) • ONNX for cross-framework portability • GPU vs CPU inference; batching for throughput • Model size optimization: quantization, distillation, pruning intro

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

M4	• CI for ML: tests for data, tests for code, tests for model behavior • CD for ML: model promotion. A/B / shadow / canary deploys • CT for ML: continuous training triggered by drift or schedule • GitHub Actions for ML: training in CI, GPU runners • Kubeflow Pipelines and Argo Workflows; pipeline-as-code • Reproducible training runs in CI; cache vs always-fresh • Model approval gates: human-in-the-loop for high-stakes models • The promotion pipeline: dev → staging → prod with eval gates
M5	• What to monitor: input distributions, predictions, latency, errors • Data drift: KS test, PSI, Wasserstein distance • Concept drift: when the ground truth changes; delayed labels • Prediction drift: subtler than data drift • Tools: EvidentlyAI, WhyLabs, Arize, NannyML • Drift dashboards in Grafana • Alerting: when to wake the on-call ML engineer • Shadow models and champion/challenger setups
M6	• Google's MLOps maturity levels (0, 1, 2) • Microsoft's MLOps maturity model (0–4) • Model governance: model cards, datasheets, lineage • Reproducibility & auditability requirements (GDPR, EU AI Act briefly) • Cost: training cost, inference cost, storage cost; FinOps for ML • Spot instances, autoscaling for inference, caching predictions • The MLOps team: roles, RACI, on-call • Anti-patterns: "the magic notebook," "the lonely data scientist"

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	*Designing Machine Learning Systems*	Chip Huyen	O'Reilly	1st Ed, 2022	978-1098107963
2	*Machine Learning Engineering*	Andriy Burkov	True Positive	1st Ed, 2020	978-1999579579
3	*Practical MLOps*	Noah Gift, Alfredo Deza	O'Reilly	1st Ed, 2021	978-1098103019
4	*Reliable Machine Learning*	Cathy Chen, Niall Murphy, Kranti Parisa, D. Sculley, Todd Underwood	O'Reilly	1st Ed, 2022	978-1098106225
5	*Introducing MLOps*	Mark Treveil et al.	O'Reilly	1st Ed, 2020	978-1492083290

31/3/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur Dist., Andhra Pradesh-522502

6	*Building Machine Learning Pipelines*	Hannes Hapke, Catherine Nelson	O'Reilly	1st Ed, 2020	978-1492053194
7	*Machine Learning Design Patterns*	Lakshmanan, Robinson, Munn	O'Reilly	1st Ed, 2020	978-1098115784

COURSE CODE:26CA2319E
COURSE NAME: AI PLATFORM ENGINEERING
L-T-P-S: 3-0-3-0 **PRE-REQ: CCDO**

CO. No	Course Outcome Description	PO-M APPING	BTL
CO1	Apply LLM serving patterns — provider APIs, vLLM, Ollama, TGI, Triton — to deploy LLMs and embedding models for production inference.	PO1, PO5	3
CO2	Apply LLM gateway patterns (LiteLLM, Portkey, Kong AI Gateway, AWS Bedrock) to provide a unified API across multiple model providers with routing, fallback, and rate limiting.	PO1, PO5	3
CO3	Analyze vector database operations — collection design, sharding, replication, hybrid search, indexing trade-offs (HNSW, IVF) — for production-scale RAG.	PO2, PO5	4
CO4	Analyze AI observability requirements — token usage, latency, cost, quality drift, prompt/response logging — and design observability for LLM applications.	PO2, PO5	4
CO5	Evaluate AI cost engineering practices — model cascading, caching, batching, prompt compression, fine-tuning vs prompting trade-offs — to manage operational costs.	PO3, PO5	5
CO6	Evaluate AI platform security and governance — prompt injection defenses, PII redaction, content moderation, audit trails, data residency — for enterprise	PO3, PO5	5

31/1/26
HoD-CSA
Dr. K. BHARGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	deployment.		

Syllabus:

Module No.	Syllabus
M1	**Model Serving Foundations** • Hosted vs self-hosted: when does each make sense • Provider APIs: OpenAI, Anthropic, Google, AWS Bedrock • Self-hosting options: Ollama (dev), vLLM (production), Text Generation Inference, Triton • GPU economics: A10G vs A100 vs H100; per-hour cost; tokens/sec/$ • Quantization tradeoffs: FP16, INT8, INT4 (GPTQ, AWQ, GGUF) — accuracy vs throughput • Embedding model serving: sentence-transformers, fastembed, hosted (OpenAI, Cohere) • Throughput optimization: batching, paged attention, speculative decoding (intuition) • Local dev workflow with Ollama for cost-free iteration
M2	**LLM Gateways & Routing** Mapped: **CO2 (BTL-3)** · 8 hrs · Project Stage 2 — Unified gateway • The N×M problem — N applications × M providers • LLM gateway pattern: one API in, many providers out • LiteLLM, Portkey, Kong AI Gateway, OpenRouter (intuition) • Building a minimal gateway: provider routing by request param • Fallback chains: provider A fails → automatic retry on provider B • Rate limiting: per-user, per-tenant, per-model (revisit CNNA) • API key management: rotation, scoping, audit • OpenAI-compatible API surface as the de-facto standard
M3	**Vector Database Operations** Mapped: **CO3 (BTL-4)** · 10 hrs · Project Stage 3 — RAG at scale • Vector database internals: HNSW vs IVF vs flat; recall vs latency trade-offs • Embedding dimensionality: 384 (BGE-small) vs 1536 (OpenAI ada) — cost vs quality • Collections, namespaces, multi-tenancy in vector stores • Sharding and replication: Qdrant clusters, Pinecone pods • Hybrid search: BM25 + dense vector with reciprocal rank fusion • Re-ranking with cross-encoders (e.g., bge-reranker) • Indexing pipelines: streaming ingestion, change data capture

31/1/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• pgvector for Postgres-native vector storage; cost vs dedicated vector DB
M4	**AI Observability & Cost Engineering** Mapped: **CO4, CO5 (BTL-4/5)** · 12 hrs · Project Stage 4 — Observed & optimized • What to measure: tokens (in/out), latency (TTFT, total), cost ($/request, $/user) • Quality metrics in production: hallucination flags, faithfulness sampling • Tracing: LangSmith, Langfuse, Helicone, Phoenix, OpenLLMetry standard • Per-tenant dashboards: cost by tenant, top users, anomaly detection • Caching strategies: exact-match cache, semantic cache (embedding-based) • Model cascading: try cheap model first; route to expensive only if confidence low • Prompt compression: LLMLingua, summary-then-process patterns • Batching for throughput; queue management for fairness
M5	**AI Security & Governance** Mapped: **CO6 (BTL-5)** · 10 hrs · Project Stage 5 — Enterprise-ready • Prompt injection threat model; OWASP LLM Top 10 • Defenses: input sanitization, output filtering, sandboxing tool execution • PII detection and redaction: Microsoft Presidio, AWS Comprehend • Content moderation: OpenAI moderation, Llama Guard, Azure AI Content Safety • Audit logging: who asked what, what was returned, retention policy • Data residency and compliance: India DPDP Act, GDPR, EU AI Act briefly • Tenant data isolation: row-level security, separate vector collections, encryption • Model card publication and risk disclosure for the platform itself

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	*AI Engineering*	Chip Huyen	O'Reilly	1st Ed, 2024	978-1098166304
2	*Building LLMs for Production*	Bouchard, Peters	Towards AI	1st Ed, 2024	979-8324731472
3	*Designing Machine Learning*	Chip Huyen	O'Reilly	1st Ed, 2022	978-1098107963

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	Systems				
4	*Reliable Machine Learning*	Cathy Chen et al.	O'Reilly	1st Ed, 2022	978-1098106225
5	*Designing Data-Intensive Applications*	Martin Kleppmann	O'Reilly	1st Ed, 2017	978-1449373320
6	*The Kubernetes Book*	Nigel Poulton	self-published	2024 Ed	—
7	*OWASP Top 10 for LLM Applications* (free)	OWASP Foundation	OWASP (free PDF)	v1.1, 2024	—

COURSE CODE: 26CA2320E
COURSE NAME :SOFTWARE ENGINEERING & AGILE
L-T-P-S : 3-0-3-0 **PRE-REQ : NIL**

CO. No	Course Outcome Description	PO-M APP INV	BTL
CO1	Apply Git collaboration workflows (feature branches, pull requests, rebasing, conflict resolution, tagged releases) and conduct constructive code reviews.	PO1, PO5	3
CO2	Apply Scrum and Kanban practices — backlog grooming, sprint planning, daily stand-ups, retrospectives, story pointing — to deliver software in short, predictable cycles.	PO1, PO5	3
CO3	Analyze test-driven development (TDD), behavior-driven development (BDD), and the test pyramid (unit, integration, end-to-end) to design effective automated test suites.	PO2, PO5	4
CO4	Analyze classical and contemporary software design patterns (Strategy, Factory, Observer, Repository, Hexagonal Architecture) and SOLID principles to refactor messy code into clean, testable code.	PO2, PO5	4
CO5	Evaluate software quality through static analysis, code metrics (cyclomatic complexity, coverage, maintainability index), technical debt accounting, and architecture decision records (ADRs).	PO3, PO5	5

HoD-CSA 31/7/26

HOD-Computer Science and Applica
Koneru Lakshmaiah Education Found
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-52

CO6	Evaluate software project management practices — estimation, dependencies, blockers, on-call rotations, postmortems — and contribute to a healthy team engineering culture.	PO3, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Git Mastery & Collaborative Workflows** **Mapped: CO1 (BTL-3) · 8 hrs · Project Stage 1 — Team boots up** • Git mental model: working tree, index, local repo, remote • Branching strategies: trunk-based vs git-flow vs GitHub flow; pick wisely • Pull request workflow; feature branch lifetime; rebase vs merge • Conflict resolution: 3-way merges; reading conflict markers; rerere • Code review etiquette: nits vs blocking comments; the "approve once it's better than what's there" rule • Conventional Commits; semantic versioning; tagged releases • Git hooks for local checks (pre-commit framework) • GitHub repository hygiene: protected branches, required reviews, CODEOWNERS
M2	**Agile, Scrum & Kanban** **Mapped: CO2 (BTL-3) · 8 hrs · Project Stage 2 — First sprint** • The Agile Manifesto and the 12 principles • Scrum framework: roles (PO, SM, Devs), events, artifacts • User stories: INVEST, story format, acceptance criteria • Story pointing: planning poker, Fibonacci, t-shirt sizes • Sprint planning, daily stand-ups, sprint reviews, retrospectives • Velocity, burndown charts, sprint goal • Kanban: WIP limits, lead time, cycle time • When Scrum vs Kanban; SAFe / LeSS overview (briefly)
M3	**Test-Driven Development & Test Pyramid** **Mapped: CO3 (BTL-4) · 10 hrs · Project Stage 3 — Test suite** • Why tests: regression safety net, design tool, documentation • The test pyramid: unit (many, fast), integration (some), E2E (few) • Red-Green-Refactor: TDD cycle in practice • Test doubles: stubs, mocks, fakes, spies; when each fits • Property-based testing: Hypothesis (Python), jqwik (Java) • Behavior-Driven Development (BDD): Cucumber, Given-When-Then • Testing in JS/TS: Jest, Vitest; Java: JUnit 5; Python: pytest • Code coverage: line vs branch; pitfalls of "100% coverage"

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
District, Andhra Pradesh-522502.

M4	**Design Patterns, SOLID & Clean Code** **Mapped: CO4 (BTL-4) · 10 hrs · Project Stage 4 — Refactoring** • SOLID principles with concrete examples in Java & Python • Creational patterns: Singleton (and its dangers), Factory, Builder • Structural patterns: Adapter, Decorator, Facade • Behavioral patterns: Strategy, Observer, Command, Template Method • Architectural patterns: MVC, Repository, Hexagonal (Ports & Adapters) • Clean Code (Robert Martin): names, functions, classes, comments • Code smells and refactoring catalog (Martin Fowler) • Domain-Driven Design (DDD) overview: ubiquitous language, bounded contexts
M5	**Quality, Technical Debt & Engineering Culture** **Mapped: CO5, CO6 (BTL-5) · 12 hrs · Project Stage 5 — Sustainable engineering** • Static analysis: ESLint, Prettier, mypy, SonarQube, Ruff • Code metrics: cyclomatic complexity, cognitive complexity, maintainability index • Technical debt: principal vs interest; debt registry; "good" vs "bad" debt • Architecture Decision Records (ADRs): the "why" trail • Estimation: 3-point estimates; the cone of uncertainty; planning fallacy • Postmortems and blameless culture • On-call practices; runbooks; alert hygiene • Engineering org models: pods, guilds, communities of practice

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	*Clean Code: A Handbook of Agile Software Craftsmanship*	Robert C. Martin	Prentice Hall	1st Ed, 2008	978-0132350884
2	*The Pragmatic Programmer: Your Journey to Mastery*	David Thomas, Andrew Hunt	Addison-Wesley	20th Anniv. Ed, 2019	978-0135957059
3	*Refactoring: Improving the Design of Existing Code*	Martin Fowler	Addison-Wesley	2nd Ed, 2018	978-0134757599

31/7/26

HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

4	*Design Patterns: Elements of Reusable Object-Oriented Software*	Gamma, Helm, Johnson, Vlissides(GoF)	Addison-Wesley	1st Ed, 1994	978-0201633610
5	*Test-Driven Development: By Example*	Kent Beck	Addison-Wesley	1st Ed, 2002	978-0321146533
6	*The Scrum Guide* (free PDF)	Ken Schwaber, Jeff Sutherland	scrumguides.org	2020 update	—
7	*Software Engineering at Google* (free online)	Titus Winters, Tom Manshreck, Hyrum Wright	O'Reilly	1st Ed, 2020	978-1492082798

COURSE CODE: 26IE3041E
COURSE NAME: TECHNICAL INTERNSHIP
L-T-P-S: 0-0-6-0 **PRE-REQ: SICE**

CO. No	Course Outcome Description	PO-M APPING	BTL
CO1	Apply software engineering practices in an industry setting including version control, code review, branching strategies, ticketing systems, and onboarding processes used by the host organization.	PO1, PO5	3
CO2	Apply technical contribution discipline (writing PRs, addressing review feedback, attending sprint ceremonies, communicating blockers) to ship measurable work product within the internship period.	PO1, PO5	3
CO3	Examine the host organization's product, business model, technical architecture, customer base, and team structure to contextualize own contribution within the broader system.	PO1, PO5	4
CO4	Examine cross-functional collaboration patterns (developer-PM-design-QA-DevOps) and the cadence of professional engineering work (planning, daily standups, retrospectives, on-call).	PO1, PO5	4
CO5	Apply professional communication and documentation standards in industry-grade artifacts including design docs, runbooks, README files, postmortems, and status updates.	PO1, PO5	5

3/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

CO6	Architect a structured internship reflection that maps observations to learning outcomes and articulates intentional career direction informed by the experience.	PO1, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Internship Selection & Onboarding** **Mapped: CO1, CO3 · 6 hrs** • Identifying suitable internship hosts (startup vs midstage vs enterprise vs research lab vs government tech) • evaluating internship offers (learning vs stipend vs brand vs role fit) • negotiation basics • pre-internship preparation (reading the public docs, learning the tech stack) • first-day onboarding etiquette • setting up dev environment • navigating internal documentation • meeting your manager and team • setting 6-week goals with the manager ***Pre-internship 2-day workshop including offer review, negotiation practice, onboarding checklist***
M2	**Module 2 — Field Week 1-2: Onboarding & Ramp-up** **Mapped: CO1, CO4 · 48 hrs** • First PR (typically docs or trivial bug fix) • navigating the codebase • understanding deployment pipelines • attending standups, planning, retrospectives • learning the issue-tracking system • understanding code-review norms • building relationships with team members • identifying the right slack channels and communication norms ***Daily structured journal entry capturing what was learned, what was confusing, what was attempted; weekly mentor check-in***
M3	**Module 3 — Field Week 3-5: Substantive Contribution** **Mapped: CO2, CO5 · 72 hrs** • Owning a meaningful piece of work (small feature, refactor, bug-cluster fix, internal tool, documentation rewrite, test coverage improvement)

31/7/26

HoD-CSA

DR. K. BHAGAVAN
HOD-Computer Science and Applica
Koneru Lakshmaiah Education Found
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-52

	• navigating ambiguity • asking for help effectively • estimating accurately • communicating progress • handling blockers • addressing code-review feedback constructively • iterating on design ***At least 3 PRs merged into the production codebase (or equivalent contribution for non-engineering internships); design doc for one feature***
M4	**Module 4 — Field Week 6: Closure & Knowledge Transfer** **Mapped: CO5 · 24 hrs** • Wrapping up open work • documenting context for the next person • handover meetings • final demo • collecting feedback from manager and peers • expressing gratitude • staying in touch professionally • LinkedIn updates ***Closure documentation; demo to team; manager feedback in writing***
M5	**Module 5 — Post-Internship Reflection & Career Mapping** **Mapped: CO6 · 4 hrs** • Synthesizing learnings into a structured reflection • identifying what kind of work / company / culture energizes you • mapping internship-acquired skills against career interests • articulating reasoning behind future career direction (full-time at host, full-time elsewhere, higher studies, founder track) • preparing the internship narrative for placement interviews ***15-page reflection report; 8-min presentation to faculty + alumni panel***

Reference Books:

Slno	Reference
1	Larson, W. — An Elegant Puzzle: Systems of Engineering Management, Stripe Press 2019, ISBN 978-1732265189

3/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh

2	Larson, W. — Staff Engineer: Leadership Beyond the Management Track, Stripe Press 2021 (free PDF)
3	Skelton, M., Pais, M. — Team Topologies, IT Revolution 2019, ISBN 978-1942788812
4	Camille Fournier — The Manager's Path, O'Reilly 2017, ISBN 978-1491973899 (also useful for IC contributors)
5	Sebastian Daschner — Architecting Modern Java EE Applications, Packt 2017
6	Industry-specific reading: company engineering blog (Stripe, Netflix, Airbnb, Uber, Cloudflare engineering blogs are exemplary)

COURSE CODE: 26UC0032E
COURSE NAME: INSIGHTS TO SUSTAINABLE DEVELOPMENT GOALS
L-T-P-S: 3-0-0-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO-H	BTL
CO1	Examine the 17 UN Sustainable Development Goals, their 169 targets, and the 248 indicators framework adopted in 2015 with target year 2030.	PO1, PO5	4
CO2	Analyze India's progress on each SDG using NITI Aayog's SDG India Index, the State and District Indices, and identify regional disparities.	PO2, PO5	4
CO3	Examine the role of technology (AI, IoT, blockchain, satellite imagery, mobile platforms) in accelerating progress toward specific SDGs.	PO1, PO5	4
CO4	Examine corporate ESG reporting, BRSR (Business Responsibility & Sustainability Report) framework mandated for top 1000 listed Indian companies, and SDG-aligned investment.	PO1, PO5	4
CO5	Apply SDG-impact assessment to evaluate the contribution of a project, product, or organization to specific goals using verifiable indicators.	PO3, PO5	3
CO6	Architect an SDG-aligned intervention or technology proposal for a real Indian challenge with measurable impact pathway and stakeholder analysis.	PO1, PO5	5

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Syllabus:

Module No.	Syllabus
M1	**Module 1 — The 17 SDGs & 2030 Agenda** **Mapped: CO1 · 7 hrs** • History of MDGs (2000-2015) and lessons learned • 2015 Paris Agenda • the 17 goals and their interconnections • the indivisibility principle • means of implementation (SDG 17) • the 5 Ps framework (People, Planet, Prosperity, Peace, Partnership) • the Voluntary National Reviews (VNRs) • SDG synergies and trade-offs ***Each student gets assigned 2 SDGs to deeply analyze; presents to class***
M2	**Module 2 — India's SDG Progress** **Mapped: CO2 · 8 hrs** • NITI Aayog SDG India Index: methodology, scoring, latest results • State Performance Index • District Index • Aspirational Districts Programme • sector deep-dives: poverty (SDG 1, MGNREGA, PMAY, PM-KISAN), hunger (SDG 2, PMGKAY, PMPOSHAN), health (SDG 3, Ayushman Bharat, IMR/MMR trends), education (SDG 4, NEP 2020, Samagra Shiksha), gender (SDG 5, Beti Bachao Beti Padhao, women's labor force), water (SDG 6, Jal Jeevan Mission, Swachh Bharat), energy (SDG 7, Saubhagya, PM Surya Ghar), inequality (SDG 10), climate action (SDG 13, India NDC) ***State-level SDG performance comparison; identification of aspirational districts and improvement strategies***
M3	**Module 3 — Tech for SDGs** **Mapped: CO3 · 8 hrs** • AI for healthcare diagnostics (SDG 3) • EdTech for low-resource learning (SDG 4) • satellite imagery for agriculture and disaster (SDG 2, 11, 13) • blockchain for land records and supply chain (SDG 1, 16)

3/1/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

	• IoT for water management (SDG 6, 11) • mobile money and digital identity (SDG 1, 8, 16) — Aadhaar, UPI, ONDC • UN ICT4SDG playbook • Indian examples: Digital Public Goods (DPGs), DPI ecosystem, JAM trinity, ABDM, FastTag, e-NAM, BHIM ***Each team picks one SDG and surveys 5 Indian tech-enabled interventions for it***
M4	**Module 4 — ESG, BRSR & Corporate Sustainability** **Mapped: CO4 · 7 hrs** • What ESG investing is • ESG ratings (MSCI, Sustainalytics, S&P Global) • BRSR framework (mandatory for top 1000 listed Indian firms from 2022-23) • BRSR Core (45 KPIs) for top 250 from 2024-25 • SEBI's enhanced disclosure requirements 2025 • greenwashing risks • SDG-aligned corporate reporting (GRI Standards, IFRS S1 & S2) • impact investing in India • CSR mandate (Section 135, Companies Act 2013) and its evolution ***Analysis of one Indian company's BRSR / sustainability report against actual practices***
M5	**Module 5 — SDG-Aligned Intervention Design** **Mapped: CO5, CO6 · 7 hrs** • Theory of change • outcome-based logic models • impact-measurement frameworks (IRIS+, Acumen Lean Data, social return on investment) • selecting indicators that are SMART • data sources for SDG indicators (NSO, NFHS, NSS, satellite open data, MoSPI) • stakeholder mapping • pilot design • randomized controlled trials in development • ethical considerations • Indian case studies of effective SDG interventions (Pratham, Akanksha, BRAC, GiveDirectly, Educate Girls) ***Each team architects an SDG-aligned intervention proposal as the final project***

21/7/25
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicat
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-52

Reference Books:

Slno	Reference
1	Sachs, J. — The Age of Sustainable Development, Columbia 2015, ISBN 978-0231173155
2	Sachs, J. — The End of Poverty, Penguin 2006, ISBN 978-0143036586
3	Banerjee, A., Duflo, E. — Poor Economics, PublicAffairs 2011, ISBN 978-1610390934
4	UN — Transforming our world: the 2030 Agenda (free, primary)
5	NITI Aayog — SDG India Index 4.0 Reports (free, latest annual)
6	Sen, A. — Development as Freedom, Anchor 2000, ISBN 978-0385720274
7	Banerjee, A., Duflo, E. — Good Economics for Hard Times, PublicAffairs 2019

COURSE CODE: 26CA3121E
COURSE NAME: HIGH PERFORMANCE COMPUTING
L-T-P-S: 3-0-0-6 **PRE-REQ: CCDO**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply parallel computing fundamentals — Amdahl's law, Gustafson's law, latency vs throughput, strong vs weak scaling — to predict and measure speedup on a real workload.	PO1, PO2	3
CO2	Apply shared-memory parallelism in Python — multiprocessing, concurrent.futures, threading limitations (GIL) — and vectorization with NumPy & Numba to accelerate CPU-bound data tasks.	PO1, PO2	3
CO3	Analyze GPU computing fundamentals — SIMT execution model, memory hierarchy, kernel launch overhead — and apply CUDA via Numba CUDA, CuPy, or PyTorch to accelerate numerical workloads.	PO2, PO3	4
CO4	Analyze distributed-computing patterns — MapReduce, dataflow, partitioning, shuffle costs — using Apache Spark / PySpark for data-	PO2, PO3	4

3/7/26

HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

	parallel workloads larger than one machine.		
CO5	Evaluate performance using profilers (cProfile, py-spy, line_profiler, NVIDIA Nsight, Spark UI) and reason about bottlenecks: compute-bound, memory-bound, I/O-bound, communication-bound.	PO4, PO5	5
CO6	Evaluatecost-performance trade-offs of CPU vs GPU vs distributed cluster on a fixed workload — including cloud GPU economics — and recommend the right tool for the job.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Parallelism Fundamentals & Profiling** Mapped: **CO1, CO5 (BTL-3/5)** · 8 hrs · Project Stage 1 — Baseline + profiles • Why parallel: end of Dennard scaling, Moore's law slowdown • Flynn's taxonomy: SISD, SIMD, MIMD, SIMT • Amdahl's law and the ceiling on speedup • Gustafson's law and the case for scaling work • Latency vs throughput; strong scaling vs weak scaling • Profiling Python: cProfile, line_profiler, py-spy (sampling), memray (memory) • Identifying bottlenecks: compute-bound vs memory-bound vs I/O-bound • Roofline model intuition *Students pick a baseline single-threaded Python data task (e.g., compute statistics over 1 GB CSV). Profile it, identify the hot path, predict theoretical speedup ceiling using Amdahl.*
M2	**Shared-Memory Parallelism & Vectorization** Mapped: **CO2 (BTL-3)** · 10 hrs · Project Stage 2 — Multi-core CPU • The Python GIL: what it actually does, and when it bites • Threading vs multiprocessing in Python; when each fits • concurrent.futures: ThreadPoolExecutor, ProcessPoolExecutor • multiprocessing.Pool, Manager, shared_memory module • NumPy vectorization: SIMD-friendly operations vs Python loops • Numba @jit: just-in-time compilation; @vectorize for ufuncs • Cython basics (preview); when to drop to C • Cache-friendly data layouts (struct of arrays vs array of structs)

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

	FastPipe gets parallelized across all CPU cores with multiprocessing, then vectorized with NumPy + Numba. Speedup measured at each step; deviation from Amdahl explained.
M3	***GPU Computing*** *Mapped:* ***CO3 (BTL-4)*** *· 10 hrs · Project Stage 3 — GPU acceleration* • *GPU architecture: streaming multiprocessors, warps, threads* • *SIMT execution model: branch divergence costs* • *GPU memory hierarchy: registers, shared memory, L2, global* • *Kernel launch overhead — when GPU isn't worth it* • *Host-device memory transfer cost; pinning; unified memory* • *Numba CUDA: writing a simple kernel in Python* • *CuPy: NumPy-compatible API on GPU* • *PyTorch tensors as a lightweight GPU compute library* *The hottest part of FastPipe is rewritten as a CuPy/Numba CUDA kernel. Cost of host-device transfer measured; net speedup vs CPU vectorized version reported.*
M4	**Distributed Computing with Spark** **Mapped: CO4 (BTL-4) · 10 hrs · Project Stage 4 — Distributed scale** • **When data exceeds one machine; the case for distribution** • **MapReduce model: map, shuffle, reduce — and why shuffle is the killer** • **Apache Spark architecture: driver, executors, cluster manager** • **RDDs vs DataFrames vs Datasets — the API evolution** • **PySpark fundamentals: transformations vs actions; lazy evaluation** • **Partitioning strategies; avoiding skew** • **Broadcast variables and accumulators** • **Spark UI for performance debugging; the shuffle stage diagnostics** ***FastPipe is reimplemented in PySpark on a 3-node cluster (Docker Compose locally; or AWS EMR free tier). Same workload now scales to 5 GB+. Shuffle costs analyzed in Spark UI.***
M5	**Cost-Performance Engineering & Cloud HPC** Mapped: **CO6 (BTL-5)** · 12 hrs · Project Stage 5 — Right tool, right cost • The cost-performance Pareto frontier: time vs $ • Cloud GPU economics: spot instances, AWS p4d/p5, GCP A2/A3

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)

	• Per-job cost: dev cost + run cost + idle cost • When CPU + multiprocessing wins (small data, debugging) • When GPU wins (matrix ops, parallel arithmetic, ML inference) • When distributed wins (data > one machine's RAM) • Cluster-as-a-service: Databricks, EMR, Dataproc • Right-sizing: the most common waste in HPC budgets • HPC workload patterns in industry: batch ETL, ML training, scientific simulation, genomics *Students compute the cost (₹) and time (sec) of running FastPipe on each of: laptop CPU, single GPU (Colab/Modal), 3-node Spark cluster. Recommend the right tool for 3 different workload sizes (1 GB, 100 GB, 10 TB).*

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	*High Performance*	Micha Gorelick, Ian	O'Reilly	2nd Ed, 2020	978-1492055020
2	*Python Concurrency with asyncio*	Matthew Fowler	Manning	1st Ed, 2022	978-1617298660
3	*Programming Massively Parallel Processors: A Hands-on Approach*	David B. Kirk, Wen-mei W. Hwu	Morgan Kaufmann	4th Ed, 2022	978-0323912310
4	*Spark: The Definitive Guide*	.Bill Chambers, Matei Zaharia	O'Reilly	1st Ed, 2018	978-1491912218
5	*Learning Spark, 2nd Edition*	Jules Damji et al.	O'Reilly	2nd Ed, 2020	9978-1491912218
6	*Dearning Spark, 2nd Edition*	Martin Kleppmann	O'Reilly	1st Ed, 2017	978-1449373320

31/7/24

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-522

7	*Introduction to High Performance Computing for Scientists and Engineers*	Georg Hager, Gerhard Wellein	CRC Press	1st Ed, 2010	978-1439811924

COURSE CODE: 26CA3122E
COURSE NAME:OFFENSICE SECURITY
L-T-P-S: 3-0-3-0 **PRE-REQ: CYBF**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply the penetration testing methodology (PTES, OWASP Testing Guide, OSSTMM) — pre-engagement, recon, threat modeling, vulnerability assessment, exploitation, post-exploitation, reporting — within an explicit authorization scope.	PO1, PO2	3
CO2	Apply reconnaissance and enumeration techniques — passive OSINT, DNS enumeration, port scanning, service fingerprinting — using tools such as Nmap, Gobuster, and OSINT platforms on authorized targets.	PO1, PO2	3
CO3	Analyze web application vulnerabilities (OWASP Top 10 — injection, broken auth, XSS, IDOR, SSRF, deserialization) using Burp Suite Community Edition on intentionally-vulnerable applications (DVWA, Juice Shop, WebGoat, PortSwigger Web Security Academy).	PO2, PO3	4
CO4	Analyze network and Active Directory attack surfaces — SMB, Kerberos, NTLM, common misconfigurations — on authorized lab targets (Hack The Box / TryHackMe AD-themed boxes).	PO2, PO3	4
CO5	Evaluate ethical, legal, and contractual constraints on penetration testing (Indian IT Act 2000 §66, GDPR, contract scoping, rules of engagement, responsible disclosure, bug bounty terms).	PO4, PO5	5
CO6	Evaluate findings, write professional penetration test reports (executive summary, technical detail, CVSS scoring, remediation guidance, retest plan), and propose defensive controls.	PO4, PO5	5

3/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

Syllabus:

Module No.	Syllabus
M1	**Methodology, Ethics & Legal Framework** Mapped: **CO1, CO5 (BTL-3/5)** · 8 hrs · Project Stage 1 — Engagement letter • Why methodology matters: structured testing vs ad-hoc "hacking" • Industry methodologies: PTES, OWASP Testing Guide, OSSTMM, NIST SP 800-115 • Pen test phases: pre-engagement, recon, threat modeling, vulnerability assessment, exploitation, post-exploitation, reporting • Rules of engagement: scope, exclusions, time windows, escalation contacts, evidence handling • Legal framework: Indian IT Act 2000 §66; CFAA; GDPR; data residency • Authorization documents: pen test contract clauses; safe harbor letters; CTF/lab terms of service • Responsible disclosure vs full disclosure vs bug bounty programs • Ethics of red teaming: physical and social engineering boundaries *Each student writes a complete engagement letter for the RangerLab range, including scope, exclusions, rules of engagement, and emergency stop procedures. Reviewed and signed by faculty before any practical work begins.*
M2	**Reconnaissance & Enumeration** Mapped: **CO2 (BTL-3)** · 10 hrs · Project Stage 2 — Recon report • Passive vs active reconnaissance • OSINT: search operators, certificate transparency logs, public DNS, Shodan/Censys, GitHub leaks • DNS enumeration techniques and limits • Port scanning fundamentals: TCP three-way handshake, SYN/FIN/UDP scans (cross-link CNNA) • Nmap basics: host discovery, service version detection, scripts (NSE) — using safe defaults • Web enumeration: robots.txt, sitemap, subdomain enumeration, content discovery • Service-specific enumeration: SMB, FTP, SSH banners • Mapping the attack surface; producing a recon report *Students enumerate a 5-host TryHackMe / HTB Academy range. Output: structured recon report with host inventory, open services, identified versions.*

HoD-CSA 21/7/26

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur, Andhra Pradesh-522

M3	**Web Application Security & OWASP Top 10** Mapped: **CO3 (BTL-4)** · 12 hrs · Project Stage 3 — Web vuln assessment • OWASP Top 10 (2021): walk through each category • Injection: SQL injection, command injection — concept + how Burp/sqlmap detect (on DVWA / WebGoat) • Broken authentication: session fixation, weak credential storage • Cross-site scripting (XSS): reflected, stored, DOM-based — concept on practice ranges • Insecure direct object references (IDOR) and broken access control • Server-side request forgery (SSRF) • Insecure deserialization: when input becomes code • Burp Suite Community: proxy, repeater, intruder; using on PortSwigger Web Security Academy labs *Students complete the PortSwigger Web Security Academy "apprentice" track for SQL Injection, XSS, IDOR, and SSRF. Findings documented with CVSS scoring.*
M4	**Network & AD Lab Exercises** **Mapped: CO4 (BTL-4) · 10 hrs · Project Stage 4 — Network/AD attack chain** • **Common Windows / AD misconfigurations on lab ranges (Kerberoasting, AS-REP roasting, ACL abuse) — conceptual + practiced on HTB / TryHackMe AD rooms** • **SMB and NTLM concepts; relevant CVEs (e.g., EternalBlue) explored only against patched lab boxes labeled for that purpose** • **Linux misconfigurations: SUID, sudo misconfigurations, cron jobs (on practice rooms)** • **Privilege escalation enumeration tools (LinPEAS, WinPEAS) on intentionally-vulnerable hosts** • **Pivoting fundamentals: SSH tunnels, port forwarding (on lab networks)** • **Post-exploitation: evidence collection, hash gathering — for report generation, not exfiltration** • **Detecting these techniques (defender's view) — preview of DEFN** • **Honeypots and detection signatures** ***Students complete a 2-host HTB / TryHackMe AD-themed network. They document the chain: initial foothold → lateral movement → privilege escalation, with appropriate CVSS scoring.***
M5	**Reporting & Defensive Recommendations** Mapped: **CO6 (BTL-5)** · 10 hrs · Project Stage 5 — Final report • Anatomy of a great pen test report: executive summary, technical detail, evidence, remediation • Writing for executives: clear risk language, business impact • Writing for engineers: precise reproduction steps, code-level remediation • CVSS v3.1 scoring methodology: base, temporal, environmental metrics

31/3/26
HoD-CSA
Dr. K. [illegible]VAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• OWASP Risk Rating methodology (alternative) • Remediation guidance: short-term mitigations, long-term controls • Retest planning and validation • Defensive recommendations: secure coding practices, security monitoring, threat modeling *Each student delivers a professional pen test report (12-20 pages) covering the RangerLab engagement. Faculty grades using a published rubric. The report is the primary deliverable.*

Reference Books:

TI	AUTHOR(S) TITLE	AUTHOR(S)	PUBLISHER	EDI	PUBLISHER	EDITION / YEAR	ISBN	TITLE
1	*The Web Application Hacker's Handbook*				Dafydd Stuttard, Marcus Pinto	Wiley	2nd Ed, 2011	978-11180 26472
2	*The Hacker Playbook 3: Practical Guide to Penetration Testing*				Peter Kim	self-published	3rd Ed, 2018	978-19809 01754
3	*Penetration Testing: A Hands-On Introduction to Hacking*				Georgia Weidman	No Starch Press	1st Ed, 2014	978-15932 75648
4	*The Pentester BluePrint: Starting a Career as an Ethical Hacker*				Phillip L. Wylie, Kim Crawley	Wiley	1st Ed, 2020	978-11196 84305
5	*OWASP Testing Guide v4.2* (free)				OWASP Foundation	OWASP	v4.2, 2020	—
6	*Real-World Bug Hunting*				Peter Yaworr	No Starc	1st Ed	978-15932

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicat
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-52

		ski	h Press	, 2019	78618
7	*NIST SP 800-115: Technical Guide to Information Security Testing and Assessment* (free)	NIST	NIST	2008	—

COURSE CODE: 26CA3123E
COURSE NAME:AGENTIC AI &APPLICATIONS
L-T-P-S: 3-0-3-0 **PRE-REQ: GAIF**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply foundational agent patterns — ReAct, function calling, planning, reflection — to build single-agent systems that complete multi-step tasks.	PO1, PO2	3
CO2	Apply shared-memory parallelism in Python — multiprocessing, concurrent.futures, threading limitations (GIL) — and vectorization with NumPy & Numba to accelerate CPU-bound data tasks.	PO1, PO2	3
CO3	Analyze agent memory architectures (short-term context, episodic, semantic, procedural) and orchestration frameworks (LangGraph, CrewAI, AutoGen, OpenAI Swarm, Anthropic SDK).	PO2, PO3	4
CO4	Analyze multi-agent collaboration patterns — supervisor-worker, debate, hierarchical teams, role specialization, handoff protocols — to compose teams of agents.	PO2, PO3	4
CO5	Evaluate agent behavior using trajectory evals, rubric-based scoring, automated red teaming, and benchmark suites (AgentBench, SWE-bench, GAIA, τ-bench).	PO4, PO5	5
CO6	Evaluate agent safety controls — sandboxing, human-in-the-loop, action confirmation, prompt injection defenses, audit trails — for production agent deployment.	PO4, PO5	5

Syllabus:

Module No.	Syllabus

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

M1	**Single-Agent Patterns: ReAct, Planning, Reflection** **Mapped: CO1 (BTL-3) · 8 hrs · Project Stage 1 — First working agent** • **What makes an LLM call "agentic" vs "just a prompt"** • **The ReAct loop: Reason → Act → Observe → Loop until done** • **Tool calling / function calling spec — OpenAI & Anthropic flavors** • **Planning patterns: task decomposition, ToT (Tree of Thoughts), plan-then-execute** • **Reflection: self-critique, self-refine, reasoning traces** • **The "agent" as a system, not a model — context engineering matters more than model choice** • **Failure modes: infinite loops, hallucinated tool calls, lost context** • **Cost & latency profile of agentic systems vs single-shot prompts** ***Students build a single-agent system using vanilla Anthropic SDK or OpenAI tool use — no framework yet. Three tools: web search, calculator, write-file. Agent solves "calculate compound returns on top 5 NSE stocks last quarter and write the result to results.md".***
M2	**Tool Engineering & Computer Use** Mapped: **CO2 (BTL-3)** · 10 hrs · Project Stage 2 — Custom tooling • Tool design: name, description, typed args, return shape, idempotency • The "clear instructions" problem — agents fail when tool descriptions are ambiguous • Code execution sandboxes: E2B, Modal, Daytona, local Docker • Browser automation as a tool: Playwright, Browserbase, Claude Computer Use • API tools: REST wrappers, GraphQL, gRPC tools • MCP (Model Context Protocol): the emerging standard for agent-tool plumbing • RAG-as-a-tool vs RAG-as-context — when each fits • Tool composition: tools that call other tools *AgentLab gains code execution (sandboxed Python via E2B or Docker), browser automation (Playwright), and at least 2 custom domain tools. Tested for idempotency and graceful failure.*
M3	***Memory, State & Frameworks*** *Mapped: **CO3 (BTL-4)** · 10 hrs · Project Stage 3 — Stateful agent* • *Memory taxonomy: short-term (context), episodic (past sessions), semantic (facts), procedural (learned skills)* • *Context window management: summarization, retrieval, compaction strategies* • *Vector-store memory: storing & retrieving past episodes* • *State machines for agents: when to use explicit state vs LLM-judgment*

31/7/26

HoD-CSA

HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• *Frameworks compared: LangGraph (graph-based), CrewAI (role-based), AutoGen (conversational), OpenAI Swarm (handoff-based), Anthropic SDK (vanilla)* • *When NOT to use a framework — simplicity wins surprisingly often* • *Graph-based agent design: nodes, edges, conditional routing in LangGraph* • *Persistent state across sessions; resumable agents* *AgentLab is rewritten in LangGraph with explicit state. Episodic memory persists across sessions. Same task across restarts proves resumability..*
M4	**Multi-Agent Collaboration** **Mapped: CO4 (BTL-4) · 10 hrs · Project Stage 4 — Agent team** • **Why multi-agent: specialization, parallelism, role separation** • **Patterns: supervisor-worker, debate, hierarchical teams, peer collaboration** • **Handoff protocols: structured outputs, role context passing** • **Communication: shared scratchpad, explicit message passing, blackboard** • **OpenAI Swarm handoff pattern; CrewAI role assignment** • **The "who decides" problem — supervisor vs democratic** • **Avoiding multi-agent overengineering — most tasks need 1 agent + good tools** • **Cost explosion in multi-agent systems; budgeting per-agent token spend** ***AgentLab grows a 3-agent team: Researcher (web + RAG), Analyst (code execution), Writer (markdown report). Supervisor coordinates. Same task as Module 1 — measure quality vs cost vs single agent.***
M5	**Evaluation, Safety & Production** Mapped: **CO5, CO6 (BTL-5)** · 12 hrs · Project Stage 5 — Production-ready agent • Why evaluating agents is hard: trajectories, not single outputs • Eval methods: outcome-based (did it succeed?), trajectory-based (what steps?), rubric-based (model-as-judge) • Benchmark suites: AgentBench, GAIA, SWE-bench, τ-bench (tau-bench), WebArena • Tracing tools: LangSmith, Langfuse, Phoenix — viewing full trajectories • Red teaming: prompt injection, indirect injection via tool outputs, jailbreaking via system-prompt extraction • OWASP LLM Top 10 applied to agents (LLM05, LLM07, LLM08 especially) • Sandboxing: code-execution isolation, network egress controls, file-system limits • Human-in-the-loop: when to require confirmation, designing for interruptibility • Audit trails: full action log, replayable trajectories, tamper-evident storage *AgentLab gets a trajectory eval suite (≥ 30 ground-truth tasks), a red-team report (≥ 5*

31/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K L Deemed to be University)
Green Fields, VADDESWARAM

	attack vectors), human-in-loop confirmation for destructive actions, and an audit log of every tool call.

Reference Books:

1	*AI Engineering*	Chip Huyen	O'Reilly	1st Ed, 2024	978-1098166304
2	*Building LLMs for Production*	Bouchard, Peters	Towards AI	1st Ed, 2024	979-8324731472
3	*Hands-On Large Language Models*	Jay Alammar, Maarten Grootendorst	O'Reilly	1st Ed, 2024	978-1098150969
4	*OWASP Top 10 for LLM Applications* (free)	OWASP Foundation	OWASP	v1.1, 2024	—
5	*Designing Data-Intensive Applications*	Martin Kleppmann	O'Reilly	1st Ed, 2017	978-1449373320
6	*Reinforcement Learning: An Introduction*	Richard S. Sutton, Andrew G. Barto	MIT Press	2nd Ed, 2018 (free online)	978-0262039246
7	*NIST AI Risk Management Framework* (AI RMF 1.0, free)	NIST	NIST	2023	—
1	*AI Engineering*	Chip Huyen	O'Reilly	1st Ed, 2024	978-1098166304

PROFESSIONAL ELECTIVE -1

Stream A: Applied AI & GenAI

COURSE CODE: 26CA31A1E
COURSE NAME: CONVERSATIONAL AI & SPEECH SYSTEMS
L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO.	Course Outcome Description	PO	BTL

Dr. HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

No		MAPPING	
CO1	Apply automatic speech recognition (ASR) and text-to-speech (TTS) APIs to build accurate transcription and natural-sounding voice synthesis pipelines for production voice applications.	PO1, PO2	3
CO2	Analyse intent classification, entity extraction, and dialogue state tracking architectures to design conversational systems that handle multi-turn interactions and context..	PO1, PO2	3
CO3	Examine voice UI patterns, barge-in handling, latency budgets, and conversational repair strategies to evaluate user experience quality of voice-first applications.	PO2, PO3	4
CO4	Design a real-time voice agent integrating ASR, NLU, dialog management, LLM reasoning, and TTS with end-to-end latency under one second and graceful error recovery..	PO2, PO3	4
CO5	Evaluate speech model performance using WER, MOS, intent F1, task success rate, and conversation analytics to identify failure modes and optimisation opportunities.	PO4, PO5	5
CO6	Architect a multilingual, telephony-integrated voice assistant deployed on cloud infrastructure with monitoring, fallback policies, and PII redaction for regulated production use..	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Speech Recognition Foundations** **Mapped: CO1, CO5 · 6 hrs** • **Acoustic modelling primer (MFCC, spectrograms)** • **modern ASR architectures (Conformer, Whisper, Wav2Vec2)** • **streaming vs batch transcription** • **word error rate (WER) and character error rate (CER)** • **language model rescoring**

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM

	• **speaker diarisation** • **voice activity detection (VAD)** • **noise robustness and echo cancellation** • **multilingual and code-switched ASR.** ***Lab-driven: students transcribe Indian English, Hindi, and Telugu audio samples with Whisper and OpenAI/AssemblyAI APIs; benchmark WER across accents and noise conditions; build a streaming transcription demo using faster-whisper.***
M2	**Speech Synthesis & Voice Cloning** Mapped: **CO1, CO3** · 6 hrs • TTS architectures (Tacotron2, FastSpeech2, VITS, neural vocoders) • SSML for prosody control • emotion and style transfer • multi-speaker and zero-shot voice cloning (XTTS, ElevenLabs) • latency-quality trade-offs (cloud vs on-device) • audio post-processing • ethical considerations in voice cloning and deepfake prevention • consent and watermarking. *Lab: students synthesise voice samples using Coqui XTTS, ElevenLabs API, and Azure Neural TTS; compare MOS (mean opinion score) via peer rating; implement SSML prosody control. Ethics seminar on voice cloning consent.*
M3	***NLU & Dialogue Management*** ***Mapped: CO2, CO5 · 6 hrs*** • ***Intent classification and slot filling (Rasa, Dialogflow CX)*** • ***entity extraction (regex, spaCy, LLM-based)*** • ***dialogue state tracking*** • ***rule-based vs ML dialogue policies*** • ***LLM-driven dialogue management*** • ***conversation memory and context windows*** • ***handling interruptions, corrections, and clarifications*** • ***intent confidence thresholds and fallback strategies.*** ***Build a Rasa-based booking assistant with 8+ intents and 5+ entities; then rebuild same flow using an LLM with function calling and compare maintainability, accuracy, and latency.***
M4	**End-to-End Voice Agents** **Mapped: CO3, CO4 · 6 hrs** • **Real-time voice agent architecture (LiveKit Agents, Pipecat, Vapi)** • **ASR-LLM-TTS pipelines with streaming and interruption handling**

3/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• **barge-in and turn-taking** • **latency budget engineering (sub-second target)** • **voice UI patterns (confirmation, clarification, error recovery)** • **conversational design principles** • **persona and tone consistency** • **integration with backend APIs and tools.** ***Capstone of module: students build a working voice ordering agent for a restaurant scenario using LiveKit Agents + Whisper + GPT-4o-mini + Cartesia TTS; measure end-to-end latency and tune for under 1 second.***
M5	***Production Voice Systems*** *Mapped:* ***CO4, CO6*** *· 6 hrs* • *Telephony integration (Twilio Voice, Plivo, Exotel for India)* • *IVR migration to AI agents* • *multilingual voice agents (Indian language coverage)* • *call analytics and conversation intelligence* • *PII detection and redaction in transcripts* • *compliance (TRAI, DPDP Act)* • *cost analysis (per-minute economics)* • *monitoring (latency p95, drop rate, CSAT)* • *A/B testing voice flows* • *fallback to human agents.* *Students integrate their voice agent with Twilio Voice (free trial), demonstrate live phone calls; implement PII redaction; produce a deployment runbook with cost projections*

Reference	
1	Jurafsky, D. & Martin, J. H. — Speech and Language Processing (3rd ed., draft chapters on ASR/dialog), Stanford 2024.
2	McTear, M. — Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots (2nd ed.), Morgan & Claypool 2022.
3	Pearl, C. — Designing Voice User Interfaces, O'Reilly 2017 (still definitive on conversational design).
4	Cohn, M. — Voice Tech in 2024: Insights, Trends, and Predictions (Voicebot.ai annual

31/7/26

HoD-CSA

Dr. K. [illegible] GAVARI
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Pradesh-5225

	report).
5	OpenAI Whisper paper: Radford et al. — Robust Speech Recognition via Large-Scale Weak Supervision, 2022.
6	LiveKit Agents documentation and reference implementations (docs.livekit.io/agents).
7	Hugging Face Audio Course — free, hands-on, includes ASR, TTS, audio classification (huggingface.co/learn/audio-course).

Stream B: Cloud Native & SRE

COURSE CODE: 26CA31B1E
COURSE NAME:CLOOUD -NATIVE INFRASTRUCTURE & GITOPS
L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse advanced Kubernetes constructs (operators, CRDs, admission controllers, RBAC) to design extensions that automate operational concerns beyond stock Deployment/Service primitives..	PO1, PO2	3
CO2	Design declarative infrastructure using Terraform and Pulumi modules, workspaces, and state management strategies suitable for multi-environment, multi-team cloud platforms...	PO1, PO2	3
CO3	Implement GitOps workflows using ArgoCD or Flux including ApplicationSets, sync waves, drift detection, and progressive delivery to manage application and infrastructure changes via Git as the source of truth.	PO2, PO3	4
CO4	Examine service mesh architectures (Istio, Linkerd, Cilium) to evaluate trade-offs in observability, security, and traffic management for microservice deployments.	PO2, PO3	4
CO5	Architect multi-cluster Kubernetes topologies including federation, fleet management, and cross-cluster networking to support production-grade availability and disaster recovery requirements.	PO4, PO5	5

3/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

CO6	Build policy-as-code guardrails using OPA Gatekeeper or Kyverno that enforce security, cost, and compliance constraints across the cloud-native platform without blocking developer velocity.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Advanced Kubernetes & Operators** **Mapped: CO1, CO5 · 6 hrs** • **Recap of core Kubernetes (skim only)** • **custom resource definitions (CRDs)** • **the operator pattern and reconciliation loops** • **Operator SDK and kubebuilder** • **admission controllers (validating and mutating webhooks)** • **RBAC deep-dive (Roles, ClusterRoles, ServiceAccounts, aggregation)** • **pod security standards (replacing PSPs)** • **ephemeral containers** • **Vertical Pod Autoscaler (VPA), Cluster Autoscaler, Karpenter** • **node taints/tolerations and affinity at scale.** ***Lab: build a tiny custom controller in Go using kubebuilder that watches a CRD (e.g., 'BackupRequest') and triggers a Job; deploy to local kind cluster and verify reconciliation behaviour.***
M2	**Infrastructure as Code at Scale** Mapped: **CO2, CO5** · 6 hrs • Terraform fundamentals refresh (HCL, providers, state) • module design and versioning (registry patterns) • workspace strategies (single state vs per-env vs per-component) • remote state backends (S3 + DynamoDB lock, Terraform Cloud) • drift detection • Terragrunt for DRY multi-env • Pulumi as TypeScript-based alternative • OpenTofu fork context • cost estimation (Infracost) in pipelines • Crossplane as Kubernetes-native IaC. *Lab: provision a 3-environment AWS VPC + EKS setup (dev/stage/prod) using Terraform modules + workspaces; integrate Infracost in a GitHub Actions PR; demonstrate drift detection.*

31/3/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

M3	***GitOps Workflows*** ***Mapped: CO3, CO5 · 6 hrs*** • ***GitOps principles (declarative, versioned, reconciled, automated)*** • ***ArgoCD architecture (Application, ApplicationSet, AppProject, sync waves, sync hooks)*** • ***Flux v2 alternatives (Kustomization, HelmRelease)*** • ***repository structure patterns (mono-repo vs multi-repo, env folders vs branches vs Kustomize overlays)*** • ***progressive delivery with Argo Rollouts (canary, blue-green)*** • ***secrets in GitOps (Sealed Secrets, External Secrets Operator with AWS Secrets Manager)*** • ***image automation*** • ***rollback strategies*** • ***preview environments per PR.*** ***Lab: deploy ArgoCD on EKS; structure a repo with 3 environments using Kustomize overlays; configure ApplicationSets to auto-sync; trigger a canary rollout via Argo Rollouts on a sample app.***
M4	**service Mesh & Multi-Cluster** **Mapped: CO4, CO5 · 6 hrs** • **Service mesh problem space (mTLS, retries, traffic shaping, observability without app code)** • **Istio architecture (control plane, sidecars, ambient mesh)** • **Linkerd as lighter alternative** • **Cilium service mesh (eBPF-based, sidecar-free)** • **traffic policies (VirtualService, DestinationRule, traffic mirroring, fault injection)** • **multi-cluster patterns (cluster federation, KubeFed deprecation context, Karmada, Open Cluster Management)** • **cross-cluster networking (Submariner, Cilium Cluster Mesh)** • **fleet management at scale.** ***Lab: install Linkerd on a kind cluster, inject sidecars on a 3-service app, observe golden metrics in Linkerd dashboard, apply traffic split for canary; optional Istio comparison.***
M5	***Policy-as-Code & Platform Engineering*** *Mapped: **CO1, CO6** · 6 hrs* • *Policy-as-code rationale (security, cost, compliance, consistency)* • *OPA Rego language fundamentals* • *Gatekeeper as Kubernetes admission controller* • *Kyverno as Kubernetes-native alternative*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur (Dist), Andhra Pradesh-522

	• *common policies (no privileged containers, required labels, image registry allowlist, resource limits required)* • *CIS Kubernetes Benchmark automation (kube-bench)* • *platform engineering and Internal Developer Platforms (IDPs)* • *Backstage as service catalogue* • *golden paths and self-service templates* • *FinOps for Kubernetes (Kubecost, OpenCost).* *Lab: install Gatekeeper, write a constraint that blocks containers without resource limits, demonstrate enforcement on a violating Deployment; add a Kyverno policy that auto-mutates missing labels.*

#	Reference
1	Burns, B. & Beda, J. & Hightower, K. — Kubernetes: Up and Running (3rd ed.), O'Reilly 2022.
2	Dobies, J. & Wood, J. — Kubernetes Operators, O'Reilly 2020 (operator pattern definitive).
3	Brikman, Y. — Terraform: Up and Running (3rd ed.), O'Reilly 2022.
4	Castelaiyn, S. & Spillner, A. — Continuous Delivery with ArgoCD, Packt 2024.
5	Calcote, L. & Butcher, Z. — Istio: Up and Running, O'Reilly 2019.
6	CNCF GitOps Working Group — GitOps Principles document (free, definitive).
7	Platform Engineering on Kubernetes — Mauricio Salatino, Manning 2023.

Stream C: Applied Cybersecurity

COURSE CODE: 26CA31C1E
COURSE NAME: WEB & API SECURITY
L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL

31/7/26
HoD-CSA
Dr. K. BHARGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO1	Analyse OWASP Top 10 (2021) and OWASP API Security Top 10 (2023) vulnerability classes to identify root causes and impact in modern web applications.	PO1, PO2	3
CO2	Design defence-in-depth controls for authentication and authorisation including OAuth 2.0, OpenID Connect, session management, JWT validation, and role-based and attribute-based access control.	PO1, PO2	3
CO3	Examine injection-class vulnerabilities (SQL injection, command injection, SSRF, XXE, XSS, template injection, LDAP injection) to construct effective input validation, output encoding, and parameterisation strategies.	PO2, PO3	4
CO4	Implement automated security testing pipelines combining SAST, DAST, IAST, SCA, and secrets scanning integrated with CI/CD for continuous web/API security verification.	PO2, PO3	4
CO5	Evaluate API-specific protections including rate limiting, schema validation, mTLS, GraphQL query cost analysis, and Web Application and API Protection (WAAP) deployment patterns.	PO4, PO5	5
CO6	Architect a comprehensive security review of a web application and its APIs, producing a remediation roadmap with prioritised fixes, compensating controls, and detection rules.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	***OWASP Foundations & Threat Modelling*** *Mapped:* ***CO1, CO6*** *· 6 hrs* • *OWASP Top 10 (2021) walkthrough with real CVE examples* • *OWASP API Security Top 10 (2023)* • *OWASP ASVS (Application Security Verification Standard) levels 1-3* • *threat modelling methodologies (STRIDE, PASTA, attack trees)* • *data flow diagrams for security* • *trust boundaries* • *security requirements engineering* • *attack surface mapping for web + API + mobile*

31/7/26
HoD-CSA

Dr. K. [illegible]AVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur Dist. Andhra Pradesh-522502.

	• *common attack vectors in modern SPAs and microservice APIs.* *Workshop: students perform a STRIDE threat model on a sample SaaS application architecture (provided); identify top 5 risks with mitigation proposals; peer-review another team's model.*
M2	**Authentication & Session Management** **Mapped: CO2, CO3 · 6 hrs** • **Password storage (Argon2id, bcrypt** • **never SHA-x)** • **credential stuffing defences (rate limiting, account lockout, CAPTCHA, breached password lists)** • **MFA (TOTP, WebAuthn/passkeys, push)** • **session management (cookie attributes — HttpOnly, Secure, SameSite, __Host- prefix)** • **session fixation** • **OAuth 2.0 flows (authorisation code with PKCE, client credentials)** • **OpenID Connect ID tokens vs access tokens** • **JWT pitfalls (alg=none, key confusion, weak signing, no expiry, sensitive data in payload)** • **refresh token rotation** • **logout in distributed systems.** ***Lab: set up Auth0 or Keycloak; implement OIDC login with PKCE in a sample SPA + API; intentionally introduce 3 JWT vulnerabilities and demonstrate exploitation in Burp Suite; then fix.***
M3	***Authorisation & Injection Vulnerabilities*** ***Mapped: CO2, CO3 · 6 hrs*** • ***Authorisation models (DAC, MAC, RBAC, ABAC, ReBAC)*** • ***Broken Object-Level Authorisation (BOLA) — the #1 API risk*** • ***Broken Function-Level Authorisation (BFLA)*** • ***IDOR*** • ***multi-tenant isolation*** • ***centralised vs decentralised authorisation (OPA, Cerbos, Oso). SQL***

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

	injection (parameterised queries, ORMs done right) • *NoSQL injection* • *command injection* • *SSRF (and impact on cloud metadata services)* • *XXE* • *LDAP injection* • *template injection (SSTI in Jinja2, Twig)* • *deserialisation vulnerabilities* • *XSS (reflected, stored, DOM-based) and modern defences (CSP, Trusted Types, output encoding by context).* *Lab: students exploit BOLA, SSRF, stored XSS, SSTI, and SQL injection in OWASP Juice Shop on the authorised practice range; then implement proper fixes in a parallel codebase.*
M4	**API Security Specifics** **Mapped: CO5, CO4 · 6 hrs** • **REST API security (URI design, mass assignment, excessive data exposure, lack of rate limiting)** • **OpenAPI/Swagger as security artefact (schema validation as a defence)** • **GraphQL security (query depth limiting, query cost analysis, introspection control, batching attacks)** • **gRPC security (mTLS, interceptors)** • **webhook security (signed payloads, replay protection)** • **idempotency keys** • **API gateway security patterns (Kong, Tyk, AWS API Gateway, Apigee)** • **WAAP / API protection products (Cloudflare, AWS WAF, Akamai)** • **bot detection** • **rate limiting algorithms (token bucket, sliding window)** • **content security policy (CSP) for browser clients.** *Lab: deploy a vulnerable GraphQL API; demonstrate query depth attack and resource exhaustion; implement query cost analysis and depth limiting; deploy AWS WAF / Cloudflare WAF in front and verify rule effectiveness.*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

M5	***Automated Security Testing & DevSecOps Integration*** *Mapped:* ***CO4, CO6*** *· 6 hrs* • *SAST tools (Semgrep, SonarQube, CodeQL)* • *DAST tools (OWASP ZAP, Burp Suite Pro, Nuclei)* • *IAST overview* • *SCA / dependency scanning (Trivy, Snyk, Dependabot, OWASP Dependency-Check, OSV-Scanner)* • *secrets scanning (gitleaks, trufflehog)* • *container image scanning* • *SBOM generation (Syft) and signing (cosign)* • *fuzzing for APIs (RESTler, schemathesis)* • *shift-left integration in CI* • *security gate policies* • *vulnerability triage (CVSS, EPSS, exploitability, business impact)* • *responsible disclosure* • *bug bounty programmes (HackerOne, Bugcrowd) — ethical context.* *Lab: build a GitHub Actions pipeline integrating Semgrep + Trivy + ZAP baseline scan + gitleaks; configure failing thresholds; produce a security gate report on a sample repo; tune rules to minimise false positives.*

BOOKSEFERENCE	
1	Stuttard, D. & Pinto, M. — The Web Application Hacker's Handbook (2nd ed.), Wiley 2011 (still definitive on web app security fundamentals).
2	Shema, M. — Hacking Web Apps, Syngress 2012.
3	Ball, C. & Hopper, V. — Hacking APIs, No Starch Press 2022 (current API security primer).
4	OWASP ASVS v4.0.3 + OWASP Top 10 (2021) + OWASP API Security Top 10 (2023) — free, primary references.
5	PortSwigger Web Security Academy materials (free, comprehensive, hands-on).
6	Hoffmann, B. — Real-World Bug Hunting, No Starch Press 2019.
7	Bynens, M. (and others) — Modern Web Security recommendations from Google + MDN docs

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

	(free, primary).

Stream D: Blockchain & Fintech

COURSE CODE: 26CA31D1E
COURSE NAME: SMART CONTRACT DEVELOPMENT & SECURITY
L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse the Ethereum Virtual Machine execution model (gas accounting, opcodes, storage layout, calldata vs memory) to predict cost and behaviour of compiled Solidity contracts.	PO1, PO2	3
CO2	Design Solidity contracts implementing major token standards (ERC-20, ERC-721, ERC-1155, ERC-4626) using OpenZeppelin libraries with appropriate access control and event emission patterns..	PO1, PO2	3
CO3	Implement comprehensive smart contract test suites combining unit, fuzz, invariant, and fork tests using Foundry to achieve high confidence before deployment.	PO2, PO3	4
CO4	Architect upgradeable contract systems using transparent proxy, UUPS, and Diamond patterns with explicit migration strategies and access controls for upgrade authority.	PO2, PO3	4
CO5	Examine smart contract security vulnerability classes (reentrancy, access control, integer issues, oracle manipulation, MEV, signature replay) and apply defensive coding patterns.	PO4, PO5	5
CO6	Build a production-grade dApp end-to-end including contract suite, deployment scripts, indexing layer, and React frontend integrated via wagmi/viem on a public testnet.	PO4, PO5	5

3/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applica
Koneru Lakshmaiah Education Found
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-52

Syllabus:

Module No.	Syllabus
M1	***Ethereum & EVM Foundations*** *Mapped:* ***CO1, CO5*** *· 6 hrs* • *Blockchain primer (assumed from prior reading)* • *Ethereum accounts (EOA vs contract), transactions, blocks, state trie* • *the EVM (stack-based, 256-bit words, opcodes)* • *gas accounting (intrinsic gas, opcode costs, refunds, EIP-1559 base fee + priority fee)* • *storage vs memory vs calldata vs stack* • *storage slot packing* • *calldata costs* • *events as cheap storage* • *the Merkle Patricia Trie at high level* • *transaction lifecycle (mempool, ordering, MEV introduction)* • *JSON-RPC interface* • *EVM-compatible chains (Polygon, Arbitrum, Optimism, Base) and what 'EVM-compatible' actually means.* *Lab: deploy a minimal counter contract to Anvil local node + Sepolia testnet using Foundry; inspect storage slots, gas usage per transaction, and event logs; visualise account state in Etherscan.*
M2	***Solidity Language Deep-Dive*** *Mapped:* ***CO2, CO5*** *· 6 hrs* • *Solidity types (uint/int variants, address payable vs address, bytes, string, structs, enums, mappings, arrays)* • *visibility (public/external/internal/private)* • *state mutability (pure, view, payable)* • *modifiers and their pitfalls* • *inheritance and the linearisation algorithm* • *abstract contracts and interfaces* • *libraries (linked vs internal)* • *events and topics* • *custom errors (cheaper than require strings)* • *receive vs fallback* • *low-level calls (call, delegatecall, staticcall) and their dangers* • *assembly basics with Yul* • *common Solidity idioms and anti-patterns.* *Lab: implement an ERC-20 from scratch (no OpenZeppelin), then refactor to extend OpenZeppelin's ERC20; benchmark gas costs and analyse storage layout differences.*

21/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

M3	***Token Standards & Composability*** ***Mapped: CO2, CO6 · 6 hrs*** • *ERC-20 (fungible tokens, allowance/approve pattern and its dangers)* • *ERC-721 (NFTs, metadata, tokenURI)* • *ERC-1155 (multi-token)* • *ERC-4626 (tokenised vaults — yield-bearing tokens)* • *Permit (EIP-2612) gasless approvals* • *meta-transactions and account abstraction (ERC-4337)* • *composability (contracts calling contracts, the 'money lego' pattern)* • *standard interfaces and ERC-165 introspection* • *upgrade-safe constructor patterns (initializers)* • *OpenZeppelin Contracts as the reference library* • *OpenZeppelin Wizard for scaffolding* • *cross-contract calls and approval patterns.* ***Lab: implement an ERC-721 NFT collection with on-chain metadata and royalty (ERC-2981); implement an ERC-4626 vault wrapping the ERC-20 from M2; demonstrate composability by depositing the ERC-20 into the vault via a script..***
M4	**Testing, Security & Upgradeability** **Mapped: CO3, CO4 · 6 hrs** • **Foundry test patterns (forge test, vm cheatcodes, makeAddr, deal)** • **unit tests** • **fuzz testing (forge's built-in fuzzing)** • **invariant testing (stateful properties)** • **fork tests (testing against mainnet state)** • **coverage reports** • **common vulnerability classes — reentrancy (CEI pattern, ReentrancyGuard, single-function vs cross-function), integer overflow/underflow (Solidity 0.8+ checks), access control (Ownable, AccessControl, OnlyRole), unchecked external calls, oracle manipulation, signature replay (nonces, domain separators), front-running and MEV** • **upgradeability (transparent proxy vs UUPS vs Beacon vs Diamond/EIP-2535)** • **storage gap pattern** • **security tooling (Slither, Mythril, Aderyn, Echidna, Halmos)** • **audit process and audit report reading.** ***Lab: write a vulnerable lending contract (intentional reentrancy + access control + integer issues); write Foundry tests that exploit each; refactor to fix; run Slither and validate clean output. Then deploy a UUPS upgradeable version with one upgrade migration.***

31/1/26

HoD-CSA

HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM

M5	***Production dApp Engineering*** *Mapped:* ***CO5, CO6*** *· 6 hrs* • *Deployment scripting in Foundry (Solidity scripts, broadcast, verification on Etherscan)* • *environment management (anvil, Sepolia, Holesky, mainnet)* • *private key management (do not commit* • *use ledger or env-based secrets via cast)* • *dApp frontend stack (wagmi, viem, RainbowKit, Reown for wallet connection* • *Next.js or Vite + React)* • *reading from contracts (calls), writing (transactions), event subscriptions* • *indexing solutions (The Graph, Goldsky, Ponder, custom indexer)* • *IPFS for off-chain metadata* • *gas-cost UX patterns (estimate before send, error handling)* • *transaction lifecycle UX (pending, included, finalised)* • *Layer 2 deployment (Optimism, Arbitrum, Base, zkSync) — bridging assets* • *bug bounty platforms (Immunefi, Code4rena, Sherlock).* *Lab: build a minimal dApp UI with Next.js + wagmi + RainbowKit; connect to deployed ERC-721 from M3 on Sepolia; show user's tokens, allow minting and transferring; deploy frontend to Vercel and verify in browser with MetaMask*

REFERNCE BOOKS

#	Reference
1	Antonopoulos, A. & Wood, G. — Mastering Ethereum, O'Reilly 2018 (free on GitHub; foundational).
2	Foundry Book — book.getfoundry.sh (free, primary; comprehensive Foundry reference).
3	Solidity documentation — docs.soliditylang.org (free, primary, language reference).
4	OpenZeppelin Contracts documentation + Wizard (free, primary).
5	Cyfrin Updraft — free comprehensive Solidity + security curriculum (updraft.cyfrin.io).
6	Ethereum Yellow Paper (formal spec, advanced reading).

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Stream E: Data Science

COURSE CODE:26CA31E1E
COURSE NAME:DATA ENGINEERING AT SCALE
L-T-P-S: 3-0-3-0 **PRE-REQ:NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse modern data architecture choices (data warehouse vs data lake vs lakehouse) including Iceberg, Delta Lake, and Apache Hudi to evaluate trade-offs in cost, performance, schema evolution, and tool compatibility.	PO1, PO2	3
CO2	Design batch data pipelines using orchestrators (Airflow, Prefect, Dagster) including DAG decomposition, dependency management, retry semantics, backfill, and lineage tracking.	PO1, PO2	3
CO3	Implement Change Data Capture (CDC) and streaming pipelines using Debezium, Kafka, and Flink to deliver near-real-time data from operational systems to analytical stores..	PO2, PO3	4
CO4	Examine dimensional modelling (Kimball star schema, Data Vault 2.0, One Big Table, Activity Schema) to design analytical schemas appropriate to query patterns and team maturity.	PO2, PO3	4
CO5	Architect data quality and observability frameworks using Great Expectations, dbt tests, Soda, and column-level lineage to detect and prevent silent data corruption before it reaches consumers.	PO4, PO5	5
CO6	Build a production-grade data platform integrating ingestion, transformation, modelling, quality checks, orchestration, and BI consumption with documented cost and SLA targets.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Modern Data Architectures** Mapped: **CO1, CO6** · 6 hrs • The data warehouse era (Inmon, Kimball, Teradata, Vertica)

31/7/26
HoD-CSA

HOD-Computer Science and Applica
Koneru Lakshmaiah Education Found
(K L Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-52

	• the data lake reaction (HDFS, S3, raw flexibility, data swamps) • the lakehouse synthesis (table formats over object storage) • cloud data warehouses (Snowflake architecture, BigQuery columnar serverless, Redshift, Databricks SQL) • open table formats — Apache Iceberg (REST catalog, hidden partitioning, time travel), Delta Lake (Databricks-led, ACID), Apache Hudi (incremental processing) • object storage as the storage layer (S3, GCS, Azure Blob, MinIO) • separation of compute and storage • the medallion architecture (bronze, silver, gold layers) • cost models compared (per-TB stored, per-query scanned, compute-second). *Workshop: students take a sample mid-size analytical workload (50TB, 1000 daily queries) and produce cost projections across Snowflake, BigQuery, and a self-managed Iceberg-on-S3 + Trino stack.*
M2	***Batch Pipelines & Orchestration*** *Mapped:* ***CO2, CO5*** *· 6 hrs* • *What an orchestrator does (vs cron + scripts)* • *Airflow architecture (DAG, scheduler, executor, web UI, providers)* • *Prefect 2/3 (task-flow, no-DAG approach, Cloud and self-hosted)* • *Dagster (asset-centric, software-defined assets, more recent paradigm)* • *orchestrator selection criteria* • *DAG design principles (idempotency of tasks, atomicity, partitioning by time)* • *backfill strategies* • *retry semantics with backoff* • *SLA monitoring* • *dependencies across teams* • *pipeline-as-code best practices* • *the dbt revolution (SQL transformations as a build system)* • *dbt Core vs dbt Cloud* • *dbt project structure (sources, models, tests, snapshots, macros, exposures).* *Lab: implement a small Airflow DAG with 5 tasks (extract from API, load to staging, transform with dbt, run tests, alert on failure); demonstrate backfill for a missed day; introduce a failure and verify retry + alert behaviour.*
M3	**Streaming & Change Data Capture** Mapped: **CO3, CO5** · 6 hrs • Why streaming (latency requirements, change data capture, real-time analytics, event-driven activation) • CDC architecture (Debezium reading PostgreSQL WAL or MySQL binlog) • event ordering and transactional consistency in CDC

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

	• Kafka as the streaming backbone • streaming sinks (Iceberg via Kafka Connect, ClickHouse, Pinot, Druid) • stream processing engines — Apache Flink (true streaming, exactly-once, savepoints), Spark Structured Streaming (micro-batch), Kafka Streams • stateful processing and watermarks • lambda vs kappa architecture (and why kappa is winning) • real-time analytics databases (ClickHouse, Apache Pinot, Apache Druid, StarRocks) • the CDC + lakehouse pattern (Debezium → Kafka → Iceberg via Flink). *Lab: set up Debezium reading from a PostgreSQL local instance; pipe changes to a local Kafka via Docker Compose; use a small Flink job (or Kafka Streams) to count events per minute; sink to a local Iceberg table; query with DuckDB.*
M4	**Modelling & Transformation** **Mapped: CO4, CO6 · 6 hrs** • **Why analytical modelling differs from operational modelling** • **Kimball dimensional modelling (fact tables, dimension tables, slowly changing dimensions Type 1/2/3, conformed dimensions)** • **Inmon corporate information factory (still useful conceptually)** • **Data Vault 2.0 (hubs, links, satellites — flexibility for source change)** • **One Big Table (OBT) and Activity Schema (modern denormalised approaches)** • **semantic layers (LookML, dbt Semantic Layer, Cube.dev)** • **metric definitions as code (no more 'whose definition of MAU is correct?')** • **incremental modelling (dbt incremental materialisations, MERGE statements)** • **slowly changing dimensions implementation in dbt** • **choosing modelling approach by team scale.** ***Lab: take a sample e-commerce dataset (orders, customers, products); build a dbt project with star schema (fact_orders + dim_customer + dim_product + dim_date); implement SCD Type 2 on dim_customer; add tests; document with dbt docs.***
M5	**Data Quality, Observability & DataOps** Mapped: **CO5, CO6** · 6 hrs • The silent data quality problem (your dashboard shows numbers • nobody knows they're wrong) • data quality dimensions (accuracy, completeness, consistency, timeliness, uniqueness, validity)

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur Dist., Andhra Pradesh

	• test frameworks (Great Expectations, dbt tests, Soda Core, Elementary) • column-level lineage (Marquez, OpenLineage, Manta) • data observability platforms (Monte Carlo, Acceldata, Bigeye) • incident response for data (data SLAs, freshness alerts, anomaly detection on metric distributions) • data contracts (the producer-consumer agreement movement) • DataOps principles • cost monitoring (Snowflake credit usage, BigQuery slot allocation) • query optimisation patterns (partition pruning, columnar projection, broadcast joins) • data team org models (centralised vs decentralised vs data mesh) • GDPR/DPDP overlay on data warehouses (right-to-be-forgotten in append-only stores). *Lab: extend the dbt project from M4 with Great Expectations tests on key metrics (revenue range, customer count growth bounds, no NULL in critical columns); set up a Slack alert on test failure; introduce intentional data corruption and verify alert.*

Reference Books:

Reference Books

#	Reference
1	Reis, J. & Housley, M. — Fundamentals of Data Engineering, O'Reilly 2022 (current canonical text).
2	Kimball, R. & Ross, M. — The Data Warehouse Toolkit (3rd ed.), Wiley 2013 (still definitive on dimensional modelling).
3	Hopwood, R. — Designing Cloud Data Platforms, Manning 2021.
4	Lakshmanan, V. — Data Mesh in Action, Manning 2024.
5	Apache Iceberg + Delta Lake + Hudi documentation (free, primary).

3/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

6	dbt documentation (free, primary; comprehensive).
7	Apache Airflow + Prefect + Dagster documentation (free, primary).

COURSE CODE:26CA3224E
COURSE NAME: DEFENSIVE SECURITY(SOC, SIEM, DFIR)
L-T-P-S: 3-0-6-0 **PRE-REQ: OFSE**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply the SOC operating model — analyst tiers (T1/T2/T3), shift handoffs, alert triage, MITRE ATT&CK mapping, runbooks — to handle simulated security alerts end-to-end.	PO1, PO2	3
CO2	Apply SIEM technologies (Splunk Free, Elastic Security / ELK Stack, Wazuh) to ingest, parse, and query security telemetry from Linux/Windows endpoints, network devices, and applications.	PO1, PO2	3
CO3	Analyze detection engineering — Sigma rules, KQL, SPL, the Pyramid of Pain — and write detections for OWASP Top 10 attacks, AD lateral movement, and persistence techniques.	PO2, PO3	4
CO4	Analyze digital forensics & incident response (DFIR) workflows — evidence acquisition, chain of custody, memory analysis (Volatility), disk forensics (Autopsy), timeline analysis — for Linux and Windows systems.	PO2, PO3	4
CO5	Evaluate threat intelligence integration — MITRE ATT&CK navigator, threat feeds (MISP), IoC management, threat hunting hypotheses — and propose threat-informed defense improvements.	PO4, PO5	5
CO6	Evaluate incident response process maturity (NIST SP 800-61) — preparation, detection & analysis, containment, eradication & recovery, post-incident — through purple team exercises and tabletop simulations.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Object-Oriented Foundations & Class Design** Mapped: **CO1 (BTL-3)** · 12 hrs (8L + 4P) · Project Stage 1 — Domain Model • Java syntax essentials: primitives, control flow, arrays, strings • Class as a first-class abstraction: fields, methods, constructors, access

3/7/26
HoD-CSA
Dr. K. BHARGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	modifiers • Encapsulation in practice: getter/setter discipline, immutable objects • Inheritance & polymorphism: method override, dynamic dispatch • Abstraction: abstract classes vs interfaces; when to use each • Packages, imports, project structure (Maven src/main/java) • equals(), hashCode(), toString() — the contract & why it matters • Java 17+ features at the foundation: records, sealed types, var *Concepts are introduced through the lens of modeling the project domain. Students design User, Asset (abstract), Stock, and MutualFund classes by Module 1 end. Lectures alternate with hands-on coding using GitHub Copilot suggestions reviewed for correctness.*
M2	**Module 2 — SIEM Engineering: Ingestion, Parsing, Querying** Mapped: **CO2 (BTL-3)** · 12 hrs · Project Stage 2 — Working SIEM • Telemetry sources: endpoint logs (Sysmon, auditd), network (Zeek/Suricata), apps (web logs) • SIEM architecture: collector → forwarder → indexer → search • Splunk Free: limits, indexing, SPL fundamentals • Elastic Security / ELK Stack: Beats, Logstash, Elasticsearch, Kibana • Wazuh as an open-source alternative (lightweight) • Parsing: regex, grok, normalization to common schema (ECS) • Time-series queries; dashboards; saved searches • Search performance: indexing strategies, time-bucketed queries *BlueRange ingests Sysmon (Windows lab), auditd (Linux lab), and Suricata logs. Kibana / Splunk dashboards show: top processes, login activity, network connections.*
M3	**Module 3 — Detection Engineering & Sigma Rules** Mapped: **CO3 (BTL-4)** · 10 hrs · Project Stage 3 — Custom detections • The detection development lifecycle: research → write → test → tune → deploy • The Pyramid of Pain: hashes (easy to evade) → TTPs (hard to evade) • Sigma: vendor-agnostic detection format; converters to SPL/KQL/EQL • SPL (Splunk Processing Language) crash course • KQL (Kusto / Microsoft Sentinel) crash course • Detections for the OFNS attacks: SQLi attempts, brute force, web shells • Detections for AD attacks: Kerberoasting, AS-REP roasting, ACL abuse • Living-off-the-land binaries (LOLBins): PowerShell, WMI, certutil — detection patterns *Students write 8 detection rules in Sigma format covering OWASP and AD attack categories. Each rule has test data, expected hit rate, expected false-positive rate.*

3/4/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

M4	**Module 4 — Digital Forensics & Incident Response (DFIR)** Mapped: **CO4 (BTL-4)** · 12 hrs · Project Stage 4 — Forensic investigation • The DFIR lifecycle: acquisition, examination, analysis, reporting • Chain of custody; evidence handling; legal admissibility • Live response vs dead-disk: when each applies • Memory acquisition: WinPMem, LiME; analysis with Volatility 3 • Disk acquisition: dd, FTK Imager; analysis with Autopsy / SleuthKit • Timeline analysis: log2timeline (Plaso); super-timeline construction • Linux-specific artifacts: /var/log, .bash_history, journalctl, audit.log • Windows-specific artifacts: registry hives, Prefetch, Amcache, ShimCache, NTFS MFT *Students conduct DFIR on 2 pre-prepared forensic images (one Linux, one Windows) — both compromised by faculty-staged attacks. Findings doc identifies attacker actions on a timeline.*
M5	**Module 5 — Threat Intelligence & Threat Hunting** Mapped: **CO5 (BTL-5)** · 10 hrs · Project Stage 5 — Threat hunt • Threat intelligence taxonomy: strategic, operational, tactical, technical • MITRE ATT&CK Navigator: heat-mapping coverage; identifying gaps • Open-source TI feeds: AlienVault OTX, abuse.ch, MISP • IoC types and lifecycles; STIX/TAXII formats • Threat hunting vs detection: hypothesis-driven vs alert-driven • Hypothesis examples: "adversary is using LOLBins for execution"; hunt for it • Diamond Model and Cyber Kill Chain frameworks • Attribution caution; the limits of "who did it" *Students conduct one threat hunt: form a hypothesis based on TI, query the SIEM, document findings, and propose a new detection rule based on what they found (or didn't find).*
M6	**Module 6 — Incident Response & Purple Team Exercises** Mapped: **CO6 (BTL-5)** · 10 hrs · Project Stage 6 — Live IR exercise • NIST SP 800-61 IR lifecycle: prepare, detect & analyze, contain, eradicate & recover, post-incident • Containment strategies: isolate vs observe; the legal-vs-operational tradeoff

3/1/26

HoD-CSA

HOD-Computer Science and Applica
Koneru Lakshmaiah Education Found
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-52

	• Eradication and verification: "how do we know they're gone?" • Communication: internal stakeholders, customers, regulators (DPDP Act notice timing) • Tabletop exercises: walking through a scenario without live action • Purple team: blue team and red team collaborate to improve detections • Post-incident review: blameless culture, lessons learned, action items • SIRP / SOAR (intro): automation in the SOC *Students participate in a live purple team exercise: OFNS-trained classmates (or faculty) launch an attack on the lab range; BlueRange must detect, contain, and produce an after-action report.*

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	*Blue Team Handbook: Incident Response Edition*	Don Murdoch	self-published	2nd Ed, 2014	978-1500734756
2	*The Practice of Network Security Monitoring*	Richard Bejtlich	No Starch Press	1st Ed, 2013	978-1593275099
3	*Practical Threat Intelligence and Data-Driven Threat Hunting*	Valentina Costa-Gazcón	Packt	1st Ed, 2021	978-1838556372
4	*The Art of Memory Forensics*	Michael Hale Ligh, Andrew Case, Jamie Levy, AAron Walters	Wiley	1st Ed, 2014	978-1118825099
5	*Incident Response & Computer Forensics*	Jason Luttgens, Matthew Pepe, Kevin Mandia	McGraw-Hill	3rd Ed, 2014	978-0071798686
6	*NIST SP 800-61 Rev. 2: Computer Security Incident Handling Guide* (free)	NIST	NIST	2012 (rev pending)	—

31/7/25

HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh 522502

PROFESSIONAL ELECTIVE -II

Stream A: Applied AI & GenAI

COURSE CODE: 26CA32A2E

COURSE NAME: COMPUTER VISION & VISUAL INTELLIGENCE

L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply pretrained object detection, segmentation, and OCR models to extract structured information from images and video streams using contemporary frameworks.	PO1, PO2	3
CO2	Analyse data labeling strategies, annotation quality, and dataset bias in computer vision pipelines to identify failure modes before model deployment.	PO1, PO2	4
CO3	Examine multimodal vision-language models (CLIP, GPT-4V, Gemini Vision) and evaluate their fitness for zero-shot, few-shot, and fine-tuned visual reasoning tasks.	PO2, PO3	4
CO4	Design an end-to-end production CV pipeline including data ingestion, model serving, drift monitoring, human-in-the-loop review, and cost-controlled inference.	PO2, PO3	5
CO5	Evaluate CV model performance using mAP, IoU, FPS, latency p95, and domain-specific business metrics to make deployment go/no-go decisions.	PO4, PO5	4
CO6	Architect a domain-specific visual intelligence application (retail analytics, document AI, video surveillance, or industrial inspection) with edge or cloud deployment, monitoring, and continuous learning.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Modern CV Foundations & Tooling** **Mapped: CO1, CO5 · 6 hrs** • Image fundamentals refresher (colour spaces, transforms,

31/7/26
HoD-CSA
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Pradesh-522

	augmentation) • modern detection (YOLOv8/v11, RT-DETR) • modern segmentation (SAM2, Mask2Former) • CV evaluation metrics (mAP, IoU, AR) • inference frameworks (PyTorch, ONNX Runtime, TensorRT) • model zoos (Ultralytics, Hugging Face, torchvision) • GPU vs CPU vs edge inference trade-offs. *Lab-driven: students benchmark YOLOv11 (n/s/m variants) on COCO subset; measure mAP and FPS on Colab T4; export to ONNX and compare inference speeds. Visual debugging with FiftyOne.*
M2	**Data Pipeline & Annotation** Mapped: **CO2, CO5** · 6 hrs • Data collection ethics (consent, privacy) • annotation tools (Roboflow, CVAT, Label Studio) • annotation quality metrics (inter-annotator agreement) • active learning and weak supervision • synthetic data generation (Stable Diffusion, Blender, Unity Perception) • class imbalance and long-tail handling • dataset versioning (DVC) • train/val/test splits and leakage • dataset bias auditing (geographic, demographic, environmental). *Hands-on: each student annotates 200 images of a domain object using Roboflow free tier; pair-annotates 50 of those images with another student; computes inter-annotator agreement; trains YOLOv11 fine-tune on their dataset and analyses failure modes.*
M3	**Document AI & OCR** Mapped: **CO1, CO3** · 6 hrs • Modern OCR engines (PaddleOCR, EasyOCR, Tesseract 5, Surya) • layout analysis and document understanding (LayoutLMv3, Donut, Nougat) • table extraction • key-value pair extraction • handwriting recognition • multilingual OCR (Indian scripts) • LLM-based document AI (GPT-4V, Claude vision, Gemini for invoice/form parsing) • structured output (JSON schemas) • accuracy benchmarking.

3/7/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh 522502.

	Lab: build invoice extraction pipeline using PaddleOCR + LayoutLMv3 + LLM post-processing; benchmark on 30 real Indian GST invoices; compare against pure GPT-4V approach on accuracy and cost per document.
M4	**Multimodal Vision-Language Models** Mapped: **CO3, CO4** · 6 hrs • CLIP and image embeddings • semantic image search • zero-shot classification • vision-language models (LLaVA, GPT-4V, Gemini Vision, Claude vision, Qwen-VL) • visual question answering • image captioning • visual grounding • prompt engineering for VLMs • cost economics of VLM APIs vs self-hosted • fine-tuning open VLMs (LLaVA, Idefics) with LoRA. *Lab 1: build a semantic image search engine over 1000 images using CLIP + FAISS. Lab 2: compare GPT-4V, Claude 3.5, Gemini 1.5 Pro, and self-hosted LLaVA on a 50-question VQA benchmark; produce cost-quality Pareto chart.*
M5	**Video Analytics & Production CV** Mapped: **CO4, CO6** · 6 hrs • Object tracking (ByteTrack, DeepSORT, BoT-SORT) • action recognition • video understanding models (VideoMAE, V-JEPA) • real-time streaming pipelines (GStreamer, OpenCV VideoCapture) • edge deployment (NVIDIA Jetson Nano, Coral TPU, Raspberry Pi 5) • model serving (Triton Inference Server, TorchServe, BentoML) • drift detection (input distribution, concept drift) • human-in-the-loop review queues • A/B testing CV models • cost-per-inference optimisation. *Lab: build a multi-object tracking pipeline on traffic CCTV footage (YOLOv11 + ByteTrack); deploy as Triton service; instrument with Prometheus; produce a 5-minute analysis video showing detection + tracking + count metrics over time.*

HoD-CSA

Dr. K. BHAGAV
HOD-Computer Science and
Koneru Lakshmaiah Educatio
(K.L. Deemed to be Un
Green Fields, VADDES
Guntur Dist., A.P., Pin

Reference Books:

S no	Title	Author(s)	Publisher	Edition / Year	ISBN
1	Computer Vision: Algorithms and Applications	Richard Szeliski	Springer	2nd Edition / 2022	9783031343772
2	Deep Learning for Coders with fastai and PyTorch	Jeremy Howard, Sylvain Gugger	O'Reilly Media	2020	9781492045526
3	Deep Learning	Ian Goodfellow, Yoshua Bengio, Aaron Courville	MIT Press	2016	9780262035613
4	YOLO Documentation	Ultralytics Team	Ultralytics	Latest	N/A
5	Hugging Face Computer Vision Course	Hugging Face Team	Hugging Face	Latest	N/A
6	Roboflow Computer Vision Best Practices Guide	Roboflow Team	Roboflow	Latest	N/A
7	Segment Anything Model 2 (SAM2), CLIP, LayoutLMv3, Donut (Research Papers)	Meta AI; OpenAI; Microsoft Research; NAVER CLOVA AI	Various	2021–2024	N/A

31/1/26
HoD-CSA

Dr. K. SRINIVASAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002, Ph: +91 - 866 - 3500122, 2576129

Stream B: Cloud Native & SRE

COURSE CODE:26CA33B4E
COURSE NAME:EVENT-DRIVEN & SERVERLESS PATTERN
L-T-P-S: 3-0-3-0 **PRE-REQ: CNIG**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse serverless function-as-a-service execution models (cold starts, concurrency limits, runtime sandboxes, billing units) across AWS Lambda, Cloudflare Workers, and Vercel/Netlify Functions to choose the right platform for a workload.	PO1, PO2	4
CO2	Design event-driven architectures using event buses, message queues, and stream processors (EventBridge, SNS/SQS, Kafka, Cloudflare Queues) to decouple producers from consumers at scale.	PO1, PO2	5
CO3	Implement edge-native applications that execute compute close to the user using V8 isolates, WebAssembly modules, and edge databases for sub-100ms global response times.	PO2, PO3	5
CO4	Examine serverless data layers (DynamoDB, D1/Turso/Neon, Cloudflare KV/R2, Upstash Redis) to evaluate fit-for-purpose across access patterns, consistency requirements, and cost models.	PO2, PO3	4
CO5	Evaluate serverless and edge applications using observability tooling (CloudWatch, Cloudflare Analytics, Datadog Serverless) on cold start latency, p95/p99, error rate, and cost per invocation.	PO4, PO5	4
CO6	Architect a globally-distributed application combining edge functions, durable storage, queues, and traditional backends with explicit failure-mode handling and cost projections at scale.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Serverless Foundations** Mapped: **CO1, CO5** · 6 hrs • FaaS execution model (handler, request lifecycle, statelessness)

3/11/25
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

	• cold starts (causes, mitigation strategies, provisioned concurrency, SnapStart) • runtimes (Node.js, Python, Go, Rust, custom) • deployment artefacts (zip, container images) • IAM and least-privilege • environment variables vs secrets • Lambda vs Cloudflare Workers vs Vercel/Netlify Functions architectural differences (containers vs V8 isolates) • pricing models comparison (per-invocation, GB-second, request-only). *Lab: deploy the same hello-world endpoint to Lambda, Cloudflare Workers, and Vercel Functions; benchmark cold start, p95 latency, and projected cost at 100K req/day.*
M2	**Event-Driven Architectures** Mapped: **CO2, CO5** · 6 hrs • Event-driven patterns (event notification, event-carried state transfer, event sourcing) • SNS pub-sub • SQS queues (standard vs FIFO) • EventBridge with rules and schemas • Kinesis and Kafka for streaming • Cloudflare Queues • idempotency keys • outbox pattern • dead letter queues • backpressure • ordering guarantees • saga pattern for distributed transactions • event-driven testing strategies. *Lab: build a 3-step order-processing pipeline (order created → inventory check → payment → fulfilment) using EventBridge + Lambda + SQS; introduce a failure and verify DLQ behaviour.*
M3	**Edge Computing** Mapped: **CO3, CO5** · 6 hrs • Why edge: latency, data sovereignty, origin offload • V8 isolates as execution model (Cloudflare Workers) • WebAssembly at edge (WASI, WasmEdge, Fastly Compute) • Lambda@Edge / CloudFront Functions • Vercel Edge Functions • edge runtime constraints (CPU time limits, no Node APIs, cold-start-

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh

	free) • use cases (auth at edge, A/B testing, image transformation, geo-routing, bot detection) • edge-side rendering (Next.js / Remix on edge) • CDN integration (caching strategies, stale-while-revalidate). *Lab: build a Cloudflare Worker that does JWT verification + geo-routing + A/B test cookie assignment in a single edge handler; deploy globally and measure p99 latency from 3 geographic locations.*
M4	**Serverless Data & State** Mapped: **CO4, CO6** · 6 hrs • Stateless vs stateful at the edge • DynamoDB single-table design refresh • Cloudflare KV (eventually consistent global key-value), R2 (S3-compatible object storage), D1 (SQLite at edge), Durable Objects (strongly consistent edge state) • Turso (libSQL replicas at edge) • Neon (serverless Postgres with branching) • PlanetScale (serverless MySQL) • Upstash (serverless Redis + Kafka) • pricing models • choosing storage by access pattern (read-heavy, write-heavy, hot key, range queries) • connection pooling problem with traditional databases (RDS Proxy, Prisma Accelerate). *Lab: build a URL shortener using Cloudflare Workers + D1 (SQL) and a second version using KV (key-value); compare write/read latency, cost projections, and consistency semantics.*
M5	**Production Serverless & Edge** Mapped: **CO5, CO6** · 6 hrs • Observability (structured logging, distributed tracing with X-Ray / OpenTelemetry, metrics) • cold start mitigation (provisioned concurrency, Lambda SnapStart, keep-warm patterns vs anti-patterns) • deployment frameworks (SST, Serverless Framework, AWS SAM, CDK, Wrangler, OpenNext) • local development (sst dev, miniflare, vercel dev) • testing strategies (unit, integration with localstack, contract tests) • cost optimisation (Lambda right-sizing with Power Tuning, ARM64/Graviton, CloudFront caching)

31/7/24

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

	• security (least-privilege IAM, signed URLs, Web Application Firewall at edge) • DPDP and data residency considerations for Indian users. *Lab: instrument an existing Lambda app with OpenTelemetry; run AWS Lambda Power Tuning to find optimal memory; produce a cost-performance report.*

Reference Books:

S no	Title	Author(s)	Publisher	Edition / Year	ISBN
1	Serverless Architectures on AWS	Peter Sbarski, Sam Kroonenburg, Mike Roberts	Manning Publications	2nd Edition / 2022	9781617296870
2	Serverless Apps with Node and Claudia.js	Peter Sbarski, Sam Kroonenburg	Manning Publications	2018	9781617295019
3	Cloudflare Workers Documentation	Cloudflare, Inc.	Cloudflare	Latest Online Documentation	N/A
4	AWS Well-Architected Framework – Serverless Lens	Amazon Web Services (AWS)	Amazon Web Services	Latest Online Documentation	N/A
5	Building Microservices	Sam Newman	O'Reilly Media	2nd Edition / 2021	9781492034025
6	Serverless Architecture Patterns and Best Practices	Adam Kelly	Packt Publishing	2023	9781804616689
7	OpenTelemetry Documentation	OpenTelemetry Authors	Cloud Native Computing Foundation (CNCF)	Latest Online Documentation	N/A

3/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002, Ph. +91 - 866 - 3500122, 2576129

Stream C: Applied Cybersecurity

COURSE CODE: 26CA32C2E
COURSE NAME: PRIVACY ENGINEERING & DATA PROTECTION
L-T-P-S: 3-0-3-0 **PRE-REQ: WAPS**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse the Indian Digital Personal Data Protection (DPDP) Act 2023 and compare with GDPR (EU) and CCPA (California) to identify obligations applicable to a given product or service.	PO1, PO2	4
CO2	Design privacy-by-design and privacy-by-default architectures applying data minimisation, purpose limitation, and storage limitation principles to real product specifications.	PO1, PO2	5
CO3	Examine de-identification techniques (anonymisation, pseudonymisation, k-anonymity, l-diversity, t-closeness, differential privacy) to choose appropriate methods for analytics, ML training, and data sharing scenarios.	PO2, PO3	4
CO4	Implement privacy-enhancing technologies (PETs) including secure aggregation, federated learning at small scale, homomorphic encryption primitives, and consent management platforms.	PO2, PO3	5
CO5	Architect data subject rights (DSR) workflows including access, correction, erasure, portability, and consent withdrawal across distributed systems with proof of completion.	PO4, PO5	5
CO6	Build a Data Protection Impact Assessment (DPIA) process and a breach notification runbook compliant with DPDP Act timelines and content requirements.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Privacy Law Landscape** Mapped: **CO1, CO6** · 6 hrs • Why privacy is engineering's problem (not just legal)

31/7/26
HoD-CSA

Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• Indian DPDP Act 2023 — scope, key terms (Data Fiduciary, Data Principal, Significant Data Fiduciary, consent, processing, transfer), grounds for processing, rights of Data Principals, obligations, penalties (up to ₹250 crore), Data Protection Board • comparison with EU GDPR (legal bases, DPO requirement, transfers under SCCs, fines up to 4% global turnover) • CCPA/CPRA (California) • sectoral regulations in India (RBI for fintech, Telecom, healthcare) • data localisation requirements • cross-border transfer mechanisms. *Workshop: students take a sample product (a fitness app) and produce a one-page legal applicability matrix (DPDP / GDPR / sectoral) with citations to specific Act sections.*
M2	**Privacy by Design & Data Mapping** Mapped: **CO2, CO5** · 6 hrs • Privacy by Design (Cavoukian's 7 principles) • Privacy Engineering disciplines (NIST Privacy Framework) • FIPPs (Fair Information Practice Principles) • data mapping and inventory (Records of Processing Activities — RoPA) • data flow diagrams from a privacy lens (PII identification, sensitivity classification, retention windows, lawful basis, recipients) • data classification taxonomies (Public, Internal, Confidential, Restricted PII / SPI) • collection minimisation • purpose limitation enforcement in code • storage limitation and automated deletion • introducing privacy requirements into PRDs and user stories. *Lab: students take an existing core project (e.g., TaskMaster) and produce a complete RoPA + data flow diagram + data classification labels for every field; identify 5 minimisation opportunities.*
M3	**De-identification & Privacy-Preserving Analytics** Mapped: **CO3, CO4** · 6 hrs • Anonymisation vs pseudonymisation (legal and technical distinctions) • direct vs quasi-identifiers

3/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• re-identification attacks (Netflix Prize, AOL search logs case studies) • k-anonymity, l-diversity, t-closeness with worked examples • differential privacy intuition and ε parameter • Google's RAPPOR / Apple's local DP / Census Bureau's TopDown • production DP libraries (OpenDP, Google DP, IBM diffprivlib) • secure multi-party computation overview • homomorphic encryption (partial vs fully) overview with a tiny worked example • trusted execution environments (Intel SGX, AWS Nitro Enclaves) overview. *Lab: students apply k-anonymity (with ARX tool) on a sample medical dataset; then run a re-identification attack on the original; finally apply local differential privacy (ε=1.0) using IBM diffprivlib on a count query.*
M4	***Privacy-Preserving ML & Federated Learning*** *Mapped:* ***CO4, CO5*** *· 6 hrs* • *Why ML is a privacy concern (training data memorisation, membership inference attacks, model inversion, prompt injection extracting training data)* • *differential privacy in ML training (DP-SGD, Opacus library)* • *federated learning architecture (centralised aggregator, secure aggregation, federated averaging)* • *FL frameworks (Flower, NVIDIA FLARE)* • *split learning* • *gradient leakage attacks* • *consent for model training* • *right to be forgotten in ML (machine unlearning, retraining triggers)* • *privacy in foundation models (data minimisation in fine-tuning, output filtering).* *Lab: train a small classifier with DP-SGD using Opacus on PyTorch; compare accuracy at ε=∞ (no privacy) vs ε=8 vs ε=1; run membership inference attack on each.*
M5	**DSRs, Breach Response & Operational Privacy** Mapped: **CO5, CO6** · 6 hrs • Data subject rights operationalisation (access, correction, erasure, portability, consent withdrawal, objection) • DSAR workflow design across microservices and data lakes • identity verification for DSARs (without collecting more data) • third-party processor obligations and contracts (DPAs) • consent management platforms (OneTrust, TrustArc, Didomi, open-

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	source alternatives like Consentmanager) • cookie consent and ePrivacy considerations • breach response — DPDP Act notification timelines (without delay), GDPR 72-hour rule • breach assessment severity (volume, sensitivity, harm potential) • regulatory + data principal notification • root cause analysis • the Data Protection Officer (DPO) role. *Lab: build a DSAR workflow for a sample app — endpoint that on request, gathers all PII for a user from 3 microservices + audit log + warehouse, returns JSON portable copy, and on erasure cascades deletion with verification token.*

Reference Books:

S no	Title	Author(s)	Publisher	Edition / Year	ISBN
1	Privacy by Design: The 7 Foundational Principles	Ann Cavoukian	Information and Privacy Commissioner of Ontario	2009	N/A
2	Privacy Is Hard and Seven Other Myths	Jaap-Henk Hoepman	MIT Press	2021	978026254262
3	The Privacy Engineer's Manifesto: Getting from Policy to Code to QA to Value	Michelle Dennedy, Jonathan Fox, Thomas Finneran	Apress	2014	978143026355
4	Practical Privacy by Design	Scott Hinde	Manning Publications	2024	978163343716
5	Digital Personal Data Protection Act, 2023 (DPDP Act)	Government of India	Ministry of Electronics and Information Technology (MeitY)	2023	N/A
6	General Data Protection	European Union; European Data	European Union Publications Office	Latest Version	N/A

3/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	Regulation (GDPR) & EDPB Guidelines	Protection Board (EDPB)			
7	NIST Privacy Framework v1.0 & NIST SP 800-53 Privacy Controls	National Institute of Standards and Technology (NIST)	NIST	Version 1.0 / Latest SP 800-53 Revision	N/A

Stream D: Blockchain & Fintech

COURSE CODE:26CA32D2E
COURSE NAME:DECENTRALIZED FINANCE & TOKENOMICS
L-T-P-S: 3-0-3-0 **PRE-REQ: SCDS**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse Automated Market Maker (AMM) mathematical models including constant product (Uniswap V2), concentrated liquidity (Uniswap V3), and stable-asset curves (Curve) to evaluate suitability for asset pair characteristics.	PO1, PO2	4
CO2	Examine major DeFi primitives (lending, derivatives, staking, liquid staking, restaking, real-world assets) by reading the actual protocol contracts and whitepapers of Aave, Compound, Lido, EigenLayer, MakerDAO, and Ondo.	PO1, PO2	4
CO3	Design tokenomics for a hypothetical protocol including supply schedules, vesting cliffs, distribution mechanisms, governance rights, fee accrual, and value-capture pathways with scenario simulations.	PO2, PO3	5
CO4	Examine MEV (Maximal Extractable Value) including frontrunning, sandwich attacks, JIT liquidity, and atomic arbitrage to design protocols and user flows that minimise extractable value.	PO2, PO3	4
CO5	Architect cross-chain protocol integrations using oracles (Chainlink, Pyth) and bridges (canonical bridges, LayerZero, Wormhole, Across) with explicit awareness of the security trade-offs of each.	PO4, PO5	5

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

CO6	Implement a working DeFi protocol slice (a simple AMM, lending pool, or staking vault) on a testnet with proper security patterns, event emission, and governance hooks.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	***AMMs & Liquidity Mathematics*** *Mapped:* ***CO1, CO6*** *· 6 hrs* • *Why DEXes replaced order books on-chain (gas + UX constraints)* • *constant product AMM (x*y=k) — Uniswap V2 maths* • *price impact and slippage* • *impermanent loss derivation* • *LP economics* • *concentrated liquidity (Uniswap V3) — tick math, range orders, capital efficiency* • *LP fee tiers* • *stable-asset curves (Curve StableSwap) — A parameter, low slippage near peg* • *weighted pools (Balancer)* • *CPMM with fees* • *Uniswap V4 hooks introduction* • *reading Uniswap V2 source code line-by-line* • *flash swaps and their applications.* *Lab: implement a minimal x*y=k AMM in Solidity (single pair, swap + add/remove liquidity); test on Anvil with fuzz tests on slippage bounds; visualise LP returns vs HODL via a Python notebook computing impermanent loss across price paths.*
M2	**Lending, Stablecoins & Money Markets** Mapped: **CO2, CO3** · 6 hrs • Over-collateralised lending model (Aave, Compound v3) • collateral factor, LTV, liquidation threshold, liquidation bonus • interest rate models (linear vs kinked vs adaptive) • cToken / aToken accounting • flash loans (use cases: liquidation, refinancing, arbitrage) • MakerDAO and DAI mechanics (CDP, stability fee, MKR governance, surplus/deficit auction)

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

	• algorithmic stablecoins post-Terra/UST (lessons learned) • centralised stablecoins (USDC, USDT mechanics) • RWA stablecoins (BUIDL, USYC) • India context (RBI position on stablecoins, CBDC pilot — Digital Rupee). • *Lab: study Compound v3's source code for one asset market; implement a tiny lending pool (deposit + borrow + interest accrual + liquidation) with a single ERC-20 collateral pair; demonstrate liquidation flow.*
M3	**Tokenomics Design** Mapped: **CO3, CO4** · 6 hrs • Token utility taxonomy (governance, work/security, payment/medium-of-exchange, meme/community) • supply schedules (fixed, inflationary, deflationary, burn mechanisms) • distribution mechanisms (airdrop, fair launch, public sale, ICO, IDO, LBP, points programmes — Friend.tech, EigenLayer, Blast) • vesting cliffs and linear unlocks • governance designs (token-weighted, quadratic voting, delegation, vote-escrow ve-tokens — Curve veCRV, Convex CVX) • fee accrual to token holders (buyback-and-burn, fee-sharing, ve-models) • the value capture problem (why most protocol tokens fail to accrue value) • tokenomics simulation tools (Cadcad, Token Engineering Commons, simple Python). *Workshop: students design tokenomics for a hypothetical lending protocol; produce a 5-page tokenomics whitepaper section with supply schedule chart, distribution pie chart, and 3 scenario simulations (bull / bear / steady state).*
M4	***MEV, Oracles & Protocol Security*** *Mapped: **CO4, CO5** · 6 hrs* • *MEV definition (max value extractable through transaction ordering)* • *attack types — sandwich attacks, JIT liquidity, generalised frontrunning, atomic arbitrage, time-bandit attacks* • *the MEV supply chain (searchers, builders, relays, validators)* • *MEV-Boost and PBS* • *Flashbots* • *user-level mitigations (private mempools — Flashbots Protect, MEV-*

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	Share) • *protocol-level mitigations (commit-reveal, batch auctions like CowSwap, on-chain RFQ)* • *oracle architectures (Chainlink push-based, Pyth pull-based, Uniswap TWAP)* • *oracle manipulation attacks (Mango Markets case study)* • *flash loan + oracle manipulation combo (CREAM, Harvest cases)* • *security audit firms in Web3 (Trail of Bits, OpenZeppelin, Cyfrin, Spearbit, ChainSecurity).* *Lab: build a simple sandwich attack simulator on a local AMM (educational only, on Anvil) — show how an attacker extracts value from a victim swap. Then redesign the swap to use commit-reveal and demonstrate the attack failing.*
M5	**L2s, Bridges & Cross-Chain DeFi** Mapped: **CO2, CO5** · 6 hrs • Why L2s for DeFi (gas costs on mainnet) • optimistic rollups (Optimism, Arbitrum, Base) — fraud proofs, 7-day exit delay, sequencer centralisation concerns • ZK rollups (zkSync Era, Starknet, Linea, Polygon zkEVM) — validity proofs • data availability layers (Celestia, EigenDA) • rollup-as-a-service (Conduit, Caldera) • native bridge architectures vs lock-and-mint vs liquidity network bridges • LayerZero, Wormhole, Across, CCIP • the bridge security tax (most major DeFi hacks have been bridge hacks) • intent-based bridging (Across) • restaking economics (EigenLayer) • shared sequencing concepts • the modular vs monolithic blockchain debate. *Lab: deploy the AMM from M1 to two L2s (Optimism Sepolia + Base Sepolia); use a testnet bridge (e.g., Hop testnet or canonical bridge) to move ERC-20 between them; instrument event tracking across both chains.*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Reference Books:

S no	Title	Author(s)	Publisher	Edition / Year	ISBN
1	Mastering Ethereum: Building Smart Contracts and DApps	Andreas M. Antonopoulos, Gavin Wood	O'Reilly Media	2018	9781491971949
2	SoK: Decentralized Finance (DeFi)	Sarah Werner et al.	IEEE / Academic Publication	2021	N/A
3	How to DeFi: Beginner / Advanced	Liam Daly	CoinGecko	2021	N/A
4	Uniswap V2, V3 & V4 Whitepapers	Hayden Adams et al.	Uniswap Labs	2020–2023	N/A
5	Aave V3, Compound V3 & MakerDAO Whitepapers	Aave Companies; Compound Labs; MakerDAO	Various Organizations	Latest Editions	N/A
6	Flashbots Research Blog (MEV)	Flashbots Team	Flashbots	Latest Online Publication	N/A
7	Token Economy: How the Web3 Reinvents the Internet	Shermin Voshmgir	BlockchainHub Berlin	2020	9783982103821

Stream E: Data Science

COURSE CODE:26CA32E2E
COURSE NAME:CUSTOMER & GROWTH ANALYTICS
L-T-P-S: 3-0-3-0 **PRE-REQ: DEAS**

CO. No	Course Outcome Description	BTL
CO1	Analyse the difference between correlation and causation using directed acyclic graphs (DAGs), confounders, mediators, and colliders to identify when statistical models will produce misleading conclusions.	4

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

CO2	Design causal identification strategies including randomised controlled trials, instrumental variables, regression discontinuity, difference-in-differences, and synthetic controls suitable to the data-generating process at hand.	5
CO3	Implement propensity score methods (matching, weighting IPTW, stratification) and double machine learning to estimate average treatment effects from observational data.	5
CO4	Examine generative model families (GANs, VAEs, diffusion models, normalising flows, autoregressive models) for their suitability to tabular synthesis, image augmentation, and counterfactual generation.	4
CO5	Examine synthetic data pipeline design choices using CTGAN, TVAE, and tabular diffusion models to generate privacy-preserving and class-balanced training data with documented utility-fidelity-privacy trade-offs.	4
CO6	Build uplift modelling pipelines (heterogeneous treatment effect estimation) using meta-learners (S/T/X-learners) and causal forests to inform business decisions including marketing targeting and policy interventions.	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Causal Reasoning Foundations** Mapped: **CO1, CO2** · 6 hrs Why correlation isn't causation (Simpson's paradox, Berkson's paradox, ecological fallacy) the ladder of causation (association, intervention, counterfactual — Pearl) structural causal models and DAGs confounders, mediators, colliders, and the back-door / front-door criteria potential outcomes framework (Rubin) ATE, ATT, ATU, CATE definitions randomised controlled trials as gold standard and their limits (cost, ethics, external validity, partial compliance) A/B testing in product analytics (sample size, power, MDE) the SUTVA assumption and its violations causal vs predictive ML — when each is appropriate.

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh

M2	**Module 2 — Quasi-Experimental Designs** Mapped: **CO2, CO5** · 6 hrs When you can't randomise (most of the real world): instrumental variables (IV) — the conditions for valid instruments, weak instrument problem, two-stage least squares (2SLS) regression discontinuity (sharp vs fuzzy RD, bandwidth selection, McCrary density test) difference-in-differences (DiD) — parallel trends assumption, staggered adoption, two-way fixed effects synthetic control method (Abadie, donor pool, weights) event studies matching methods (exact, coarsened exact, Mahalanobis distance, propensity scores) the modern econometrics renaissance (Imbens, Athey, Card on causal credibility) causal inference in tech companies (Microsoft ExP, Uber, Airbnb experimentation platforms). *Lab: implement a difference-in-differences analysis on a public policy dataset (e.g., minimum wage change in one state vs neighbour, smoking ban impact on bar revenue); produce parallel-trends plot; report ATT with confidence intervals; discuss assumption violations.*
M3	**Module 3 — Modern Causal ML** Mapped: **CO3, CO6** · 6 hrs Propensity scores (estimation via logistic regression and gradient-boosted models) inverse probability of treatment weighting (IPTW) matching on propensity scores (1:1, k:1, with caliper, with replacement) stratification doubly robust estimation double machine learning (Chernozhukov et al.) causal forests and generalised random forests (Wager & Athey, EconML, GRF library) CATE estimation meta-learners (S-learner, T-learner, X-learner, R-learner) uplift modelling (the right metric for marketing, retention, treatment selection problems) Qini curves and AUUC for uplift evaluation the EconML library by Microsoft CausalML library by Uber DoWhy library by Microsoft for causal inference workflow.

31/7/26

HoD-CSA

Dr. K. DHASAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	Lab: implement an end-to-end causal analysis on a marketing campaign dataset using DoWhy + EconML — define DAG, estimate propensity scores, compute ATE via IPTW + double ML, estimate CATE via causal forest, identify highest-uplift segments.
M4	**Module 4 — Generative Models for Data Science** Mapped: **CO4, CO5** · 6 hrs Generative model taxonomy refresher (GANs, VAEs, diffusion, normalising flows, autoregressive) why generative models matter to data science (synthetic data, augmentation, counterfactual analysis, anomaly detection via density modelling) tabular generative models — CTGAN and TVAE (Synthetic Data Vault), tabular diffusion (TabDDPM), TabDDPM vs VAE comparison image augmentation with diffusion models for ML training conditional generation for class imbalance (SMOTE-GAN) evaluating synthetic tabular data — fidelity (statistical similarity), utility (downstream model performance), privacy (membership inference attacks, distance to closest record) the Synthetic Data Vault (SDV) library production synthetic data platforms (Mostly AI, Tonic.ai, Gretel.ai). *Lab: take a small tabular dataset (e.g., adult income); train CTGAN and TVAE to generate synthetic data; evaluate fidelity (column distributions, correlation matrix), utility (logistic regression accuracy on real test set), and privacy (DCR distance to closest record); produce a quality report.*
M5	**.Module 5 — Counterfactual Analysis & Production Considerations** Mapped: **CO5, CO6** · 6 hrs Counterfactual generation (what would have happened if treatment had been different) using generative models combined with causal models counterfactual explanations for ML predictions (DiCE library — finding the minimal change to flip a prediction) algorithmic fairness via counterfactuals (counterfactual fairness) A/B testing in production at scale (Microsoft ExP architecture, sequential testing, CUPED variance reduction, peeking problem) switchback experiments for marketplace settings quasi-experimental causal inference at companies (uplift on the entire user base when you can't AB test) MLOps for causal models (model drift differs from prediction drift) communicating causal findings to non-technical stakeholders (effect sizes in business terms, confidence intervals, robustness checks).

31/3/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

	Lab: take a binary classifier (e.g., loan default predictor) trained earlier in the term; use DiCE library to generate counterfactual explanations for 10 denied applicants — what minimal feature change would have flipped the decision; evaluate plausibility of generated counterfactuals.

Reference Books:

#	REFERENCE
1	Pearl, J. & Mackenzie, D. — The Book of Why. Basic Books 2018 (accessible introduction).
2	Pearl, J., Glymour, M., Jewell, N. — Causal Inference in Statistics: A Primer, Wiley 2016 (technical primer).
3	Cunningham, S. — Causal Inference: The Mixtape, Yale University Press 2021 (free online — most accessible textbook for econometric methods).
4	Hernán, M. & Robins, J. — Causal Inference: What If, CRC Press 2024 (free online; epidemiological perspective).
5	Imbens, G. & Rubin, D. — Causal Inference for Statistics, Social, and Biomedical Sciences, Cambridge 2015.
6	Goodfellow, Bengio, Courville — Deep Learning, MIT Press 2016 (free, generative models chapters).
7	Synthetic Data Vault (SDV) + EconML + DoWhy + DiCE documentation (free, primary).

31/7/26

HoD-CSA

Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

PROFESSIONAL ELECTIVE III

Stream A: Applied AI & GenAI

COURSE CODE: 26CA32A3E
COURSE NAME:DEEP LEARNING ARCHITECTURES
L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse the mathematical and architectural foundations of attention mechanisms and transformer encoder-decoder structures to explain why they outperform recurrent and convolutional baselines on sequence and vision tasks.	PO1, PO2	4
CO2	Examine modern vision architectures (Vision Transformers, ConvNeXt, hybrid models) and graph neural networks to compare inductive biases and select architectures suited to domain data characteristics.	PO1, PO2	4
CO3	Analyse generative architectures (VAEs, GANs, diffusion models, normalising flows) to evaluate trade-offs in sample quality, training stability, and inference cost.	PO2, PO3	4
CO4	Design self-supervised and contrastive learning pipelines (SimCLR, MoCo, MAE, DINO) that pretrain representations on unlabelled data for downstream tasks with limited labels.	PO2, PO3	5
CO5	Implement at least three modern architectures from primary research papers using PyTorch, including custom training loops, mixed-precision, and gradient checkpointing for memory efficiency.	PO4, PO5	5
CO6	Critique recent deep learning research papers using a structured methodology and synthesise findings into architectural design decisions for a novel application.	PO4, PO5	5

Syllabus:

Module No.	Syllabus

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM

M1	**Advanced Attention & Modern Transformer Variants** Mapped: **CO1, CO5 (BTL-4/5)** · 6 hrs · Project Stage 1 — vision arch reproduction • Brief recap (assumes DLDO foundations on vanilla Transformer + multi-head attention) • Modern positional encodings: RoPE (rotary), ALiBi, no-PE designs • Attention efficiency: Flash Attention 2/3, sliding-window, sparse, multi-query (MQA), grouped-query (GQA) • KV-cache mechanics in inference; cache compression; speculative decoding • Long-context techniques: YaRN, NTK-aware scaling, Ring Attention • Mixture-of-Experts: token routing, load balancing, MoE in Mixtral / DeepSeek-V3 • State-space alternatives: Mamba, Mamba-2, hybrid SSM-attention models • Quantisation impact on attention quality (INT8, INT4, FP8) *Lab: extend a small Transformer (from DLDO) with RoPE positional encoding; benchmark vs sinusoidal on a long-context task; profile attention compute and memory.*
M2	**Vision Transformers & Modern CNNs** Mapped: **CO2, CO5** · 6 hrs • ViT (patch embeddings, class token, position embeddings) • Swin Transformer (windowed attention) • ConvNeXt (modernised CNNs) • hybrid architectures (CoAtNet, MobileViT) • inductive biases (translation equivariance vs none) • data efficiency comparison (ViT vs CNN at small dataset scale) • hierarchical feature pyramids • DETR for detection • SAM/SAM2 for segmentation as ViT downstream. *Lab: train ViT-tiny vs ResNet-18 on CIFAR-100 with same compute budget; analyse where each fails; reproduce attention rollout visualisation.*
M3	**Generative Architectures** Mapped: **CO3, CO5** · 6 hrs • Variational autoencoders (ELBO, reparameterisation, posterior collapse) • GANs (DCGAN, StyleGAN3, mode collapse, FID) • diffusion models from scratch (DDPM math, noise schedules, U-Net backbone, classifier-free guidance, latent diffusion / Stable Diffusion architecture) • normalising flows • comparison matrix (sample quality, diversity, training stability, inference cost, controllability). *Lab: train a tiny DDPM on MNIST or Fashion-MNIST from scratch; sample 64*

HoD-CSA

HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	images at different inference steps; benchmark FID. Optional GAN comparison.
M4	**Graph Neural Networks & Self-Supervised Learning** Mapped: **CO2, CO4** · 6 hrs • Graph fundamentals (nodes, edges, message passing) • GCN, GAT, GraphSAGE, GIN • node-level vs edge-level vs graph-level tasks • over-smoothing • PyTorch Geometric. Self-supervised learning: contrastive (SimCLR, MoCo, BYOL) • masked image modelling (MAE) • DINO and self-distillation • CLIP as vision-language SSL • downstream evaluation protocol (linear probe, fine-tune). *Lab 1: train a GCN on Cora citation graph (PyTorch Geometric). Lab 2: pretrain a tiny MAE on STL-10 unlabelled split, linear-probe on labelled split.*
M5	**Architectural Reasoning & Research Reading** Mapped: **CO6, CO5** · 6 hrs • Reading research papers efficiently (3-pass method) • reproducing results from a paper • ablation study design • choosing architectures for a problem (data scale, compute budget, latency target, interpretability) • efficient architectures (MobileNet, EfficientNet, distillation, quantisation) • state-space models (Mamba, S4) as transformer alternatives • mixture-of-experts (MoE) at high level • future directions (test-time training, world models). *Each student presents a 15-minute paper review on a 2024-25 architecture paper using a structured rubric (problem, method, key insight, ablations, limitations, reproducibility).*

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	Deep Learning	Ian Goodfellow, Yoshua Bengio,	MIT Press	2016	978-0262035613

31/7/26

HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

		Aaron Courville			
2	Understanding Deep Learning	Simon J.D. Prince	MIT Press	2024	978-0262048644
3	Deep Learning: Foundations and Concepts	Christopher M. Bishop, Hugh Bishop	Springer	2024	NA
4	*Attention Is All You Need*	Ashish Vaswani et al.	NeurIPS (Conference Paper)	2017	NA
5	*An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale*	Alexey Dosovitskiy et al.	ICLR (Conference Paper)	2021	NA
6	*Denoising Diffusion Probabilistic Models*	Jonathan Ho et al.	NeurIPS (Conference Paper)	2020	NA
7	*Hugging Face Diffusion Course + Deep RL Course*	Hugging Face	Hugging Face	Online Course	NA

Stream B: Cloud Native & SRE

COURSE CODE: 26CA32B3E
COURSE NAME: SITE RELIABILITY ENGINEERING & OBSERVABILITY
L-T-P-S: 3-0-3-0
PRE-REQ: CNIG

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse Site Reliability Engineering principles (SLI, SLO, error budgets, toil, eliminating heroics) to translate vague reliability	PO1, PO2	4

31/7/26
HoD-CSA
Dr. K. SRIRAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

	expectations into measurable engineering contracts.		
CO2	Design observability strategies combining metrics, structured logs, and distributed traces using OpenTelemetry, Prometheus, Loki, and Tempo to make production systems debuggable without re-deployment.	PO1, PO2	5
CO3	Examine alerting philosophies (symptom-based vs cause-based, page vs ticket vs nothing) to construct alert rules that minimise noise and catch real customer-impacting issues.	PO2, PO3	4
CO4	Implement chaos engineering experiments using Litmus or Chaos Mesh to validate hypotheses about system resilience before incidents occur in production.	PO2, PO3	5
CO5	Architect incident response processes including on-call rotations, runbooks, war-room facilitation, and blameless postmortems to learn from failures and improve systems.	PO4, PO5	5
CO6	Build capacity planning and performance engineering practices including load testing, resource right-sizing, and predictive scaling to anticipate failure modes before they arrive.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**SRE Principles & Service-Level Objectives** Mapped: **CO1, CO5** · 6 hrs • SRE history (Google SRE book), Dev vs Ops vs SRE • the four golden signals (latency, traffic, errors, saturation) • USE method (utilisation, saturation, errors) • RED method (rate, errors, duration) • SLI selection (request-based vs window-based, user journey vs technical metric) • SLO setting (target, time window, customer-meaningful) • error budget arithmetic • error budget policies (feature freeze, rollback triggers) • toil identification and elimination • the 50% rule (engineering work vs operational work). *Workshop: students take a sample multi-service application and define 5 SLIs +*

3/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	corresponding SLOs with error budget calculations; defend choices in peer review.
M2	**Observability — Metrics & Logs** Mapped: **CO2, CO5** · 6 hrs • Observability vs monitoring distinction • OpenTelemetry architecture (SDK, collector, exporters) • Prometheus (data model, PromQL, recording rules, alerting rules) • cardinality and label hygiene • Grafana (dashboards, panels, variables, alerts) • structured logging (JSON, log levels, correlation IDs) • log aggregation (Loki + LogQL, ELK alternative) • log sampling and cost control • Sentry for error tracking • cost economics of observability vendors (Datadog, New Relic, Honeycomb). *Lab: instrument a sample Go/Node app with OpenTelemetry; export metrics to Prometheus and traces to Tempo; build a Grafana dashboard showing the four golden signals.*
M3	**Distributed Tracing & Alerting** Mapped: **CO2, CO3** · 6 hrs • Distributed tracing concepts (trace, span, context propagation, baggage) • W3C trace context standard • sampling strategies (head-based, tail-based, dynamic) • Tempo, Jaeger, Honeycomb backends • tracing across async boundaries (queues, retries) • alerting design (multi-window multi-burn-rate alerts based on Google SRE workbook) • symptom-based vs cause-based alerts • alert fatigue and noise reduction • PagerDuty / Opsgenie / Grafana OnCall escalation policies • runbook conventions. *Lab: implement multi-window multi-burn-rate alerting rule in Prometheus for a sample SLO; verify it triggers correctly under simulated 1% and 10% error injection.*
M4	**Chaos Engineering & Resilience Patterns** Mapped: **CO4, CO6** · 6 hrs • Chaos engineering principles (Netflix Principles of Chaos) • steady state hypothesis

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

	• blast radius control • game days vs continuous chaos • tools (Litmus, Chaos Mesh, AWS Fault Injection Simulator, Gremlin) • failure injection types (pod kill, network latency, CPU stress, dependency failure) • resilience patterns (circuit breaker, bulkhead, timeout, retry with exponential backoff and jitter, rate limiter, fallback) • idempotency and at-least-once vs exactly-once • graceful degradation strategies. *Lab: install Chaos Mesh on a kind cluster; run a pod-kill experiment on a sample app with 3 replicas; verify SLO impact and recovery time.*
M5	**Module 5 — Incident Response, Postmortems & Capacity** Mapped: **CO5, CO6** · 6 hrs • Incident lifecycle (detect, declare, respond, mitigate, resolve, learn) • incident commander role • communication patterns (status page, Slack channel, customer comms) • blameless postmortem structure (timeline, contributing factors, action items, lessons) • just culture • capacity planning (USE method baseline, growth projections, headroom targets) • load testing (k6, Locust, Gatling) • right-sizing (Kubernetes VPA recommendations, AWS Compute Optimizer) • performance engineering (profiling with pprof, flame graphs) • cost vs performance trade-offs. *Workshop: each team facilitates a mock incident scenario (pre-written) playing different roles; produces a blameless postmortem document the next session.*

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	Site Reliability Engineering	Betsy Beyer et al.	O'Reilly Media	1st Ed., 2016	NA
2	The Site Reliability Workbook	Betsy Beyer et al.	O'Reilly Media	1st Ed., 2018	NA
3	Building Secure and Reliable	Heather Adkins etal.	O'Reilly Media	1st Ed., 2020	NA

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	Systems				
4	Observability Engineering	Charity Majors, Liz Fong-Jones, George Miranda	O'Reilly Media	1st Ed., 2022	978-1492076445
5	Chaos Engineering: System Resilience in Practice	Casey Rosenthal, Nora Jones	O'Reilly Media	1st Ed., 2020	978-1492043867
6	Practical Monitoring	Mike Julian	O'Reilly Media	1st Ed., 2017	978-1491957356
7	Web Operations	John Allspaw	O'Reilly Media	1st Ed., 2010	978-1449377441

Stream C: Applied Cybersecurity

COURSE CODE: 26CA32C3E
COURSE NAME: CLOUD SECURITY & ZERO TRUST
L-T-P-S: 3-0-3-0
PRE-REQ: WAPS

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse the AWS, Azure, and GCP shared-responsibility models and native security services to identify what the cloud provider secures versus what the customer must secure for each service category.	PO1, PO2	4
CO2	Design cloud Identity and Access Management (IAM) using least-privilege roles, attribute-based access, federated identity, and short-lived credentials to eliminate static long-lived secrets.	PO1, PO2	5
CO3	Implement Zero Trust Network Access (ZTNA) replacing perimeter trust with identity-aware proxies, device posture checks, and continuous verification informed by the BeyondCorp and NIST SP 800-207 models.	PO2, PO3	5

31/7/26

HoD-CSA

HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO4	Build secrets management, encryption-at-rest, and encryption-in-transit controls using KMS, HSM, envelope encryption, and key rotation policies appropriate for regulated workloads.	PO2, PO3	5
CO5	Examine multi-account / multi-subscription landing zone architectures (AWS Organisations + SCPs, Azure Management Groups, GCP Organisation Policy) to enforce guardrails at scale.	PO4, PO5	4
CO6	Architect Cloud Security Posture Management (CSPM), Cloud Workload Protection (CWPP), and Cloud Infrastructure Entitlement Management (CIEM) capabilities into a continuous monitoring programme for cloud-native workloads.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Cloud Security Foundations & Shared Responsibility** Mapped: **CO1, CO5** · 6 hrs • Shared responsibility model deep-dive across IaaS / PaaS / SaaS / serverless / managed Kubernetes • common cloud breach patterns (Capital One, Uber, Codecov case studies) • CIS Benchmarks (AWS, Azure, GCP, Kubernetes) • Center for Internet Security Top 18 Controls • AWS Well-Architected Security Pillar • Azure Cloud Adoption Framework Secure methodology • GCP Cloud Architecture Framework Security • native security services overview (AWS Security Hub, GuardDuty, Inspector, Detective, Macie • Azure Defender, Sentinel • GCP Security Command Center, Chronicle). *Workshop: students take a sample 3-tier cloud architecture and produce a shared-responsibility matrix mapping every component to provider/customer accountability + applicable CIS controls.*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M2	**Cloud IAM Deep-Dive** Mapped: **CO2, CO3** · 6 hrs • IAM principals (users, groups, roles, service accounts) • policy languages (AWS IAM JSON, Azure RBAC + ABAC, GCP IAM bindings) • least-privilege policy generation (AWS Access Analyzer, IAM Access Advisor) • permission boundaries • service control policies • OIDC federation (GitHub Actions → AWS, no static AWS keys) • SAML federation with corporate identity providers • just-in-time access (AWS IAM Identity Center temporary elevation) • break-glass account procedures • CloudTrail / Azure Activity Log / GCP Audit Logs as the IAM source of truth • common IAM misconfigurations (wildcard actions, overly permissive trust policies, public S3 bucket policies). *Lab: configure AWS IAM Identity Center + GitHub OIDC federation so a GitHub Actions workflow can deploy infrastructure to AWS without any long-lived credentials in repository secrets; verify in CloudTrail.*
M3	**Zero Trust Architecture** Mapped: **CO3, CO5** · 6 hrs • Why perimeter trust fails (insider threat, ransomware lateral movement, remote work) • Forrester Zero Trust eXtended • Google BeyondCorp evolution and architecture • NIST SP 800-207 Zero Trust Architecture (policy decision point, policy enforcement point, dynamic trust assessment) • identity-aware proxies (Google IAP, Cloudflare Access, Tailscale) • device posture (mobile device management, endpoint compliance) • microsegmentation (zero trust networking with Cilium NetworkPolicy, Calico, AWS VPC + Security Groups, Azure NSGs + ASGs) • ZTNA vs VPN • software-defined perimeter (SDP) • the consumer Zero Trust journey for a typical mid-size org. *Lab: deploy Cloudflare Access (free tier) in front of a sample web app; require Google Workspace SSO + device posture (fresh OS patch level); demonstrate access denial for non-compliant device and successful access otherwise.*

31/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

M4	**Secrets Management & Encryption** Mapped: **CO4, CO6** · 6 hrs • Why static secrets in environment variables are a problem • secrets management platforms (HashiCorp Vault, AWS Secrets Manager, Azure Key Vault, GCP Secret Manager, Doppler) • dynamic secrets (Vault database secrets engine generates short-lived DB credentials) • secret rotation policies and automation • KMS architectures (envelope encryption pattern) • customer-managed keys (CMK) vs provider-managed • HSM-backed keys (CloudHSM, Azure Dedicated HSM) • BYOK (bring your own key) • encryption at rest (S3 SSE-KMS, EBS, RDS, DynamoDB encryption) • encryption in transit (TLS 1.3, mTLS, certificate management with ACM/cert-manager) • key rotation cadence • the encryption-as-feature problem (you still have to manage who can decrypt). *Lab: deploy HashiCorp Vault locally; integrate Vault database secrets engine with PostgreSQL so application gets fresh 1-hour DB credentials per session; rotate the root key and verify continuity.*
M5	**CSPM, CWPP, CIEM & Continuous Monitoring** Mapped: **CO5, CO6** · 6 hrs • Cloud Security Posture Management (CSPM) — continuous configuration scanning vs CIS / NIST / CSA STAR baselines • tools (AWS Security Hub, Azure Defender for Cloud, GCP SCC, Wiz, Prisma Cloud, Lacework, open-source: Prowler, ScoutSuite, Checkov, kubescape) • auto-remediation patterns • Cloud Workload Protection Platforms (CWPP) — runtime threat detection on VMs, containers, serverless (Falco, AWS GuardDuty Runtime Monitoring, Sysdig Secure) • CIEM — entitlement excess, dormant access, toxic combinations (AWS IAM Access Analyzer, Wiz, Sonrai) • landing zones (AWS Control Tower, Azure Landing Zones) • SCPs / Azure Policy / GCP Org Policy as preventive guardrails • continuous compliance dashboards. *Lab: run Prowler against a personal AWS sandbox account; score against CIS AWS Foundations Benchmark; pick top 5 findings and remediate; demonstrate the score improvement on re-run.*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	Amazon Web Services in Action	Andreas Wittig, Michael Wittig	Manning Publications	3rd Ed., 2023	978-1633439160
2	Cloud Security: A Comprehensive Guide	M.Estape,B.Cantrill	O'Reilly Media	2024	NA
3	Cloud Computing Security	K. Hashizume et al.	IEEE (Survey Paper)	2013	NA
4	NIST SP 800-207: Zero Trust Architecture	Scott Rose et al.	NIST	2020	NA
5	Zero Trust Security: An Enterprise Guide	Jason Garbis, Jerry Chapman	Apress	1st Ed., 2021	978-1484278260
6	AWS Well-Architected Framework – Security Pillar	Amazon Web Services	AWS	Latest Online Edition	NA
7	CIS Benchmarks for AWS / Azure / GCP / Kubernetes	Center for Internet Security (CIS)	CIS	Latest Online Edition	NA

31/7/24
HoD-CSA
Dr. R. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Stream D: Blockchain & Fintech

COURSE CODE: 26CA32D3E
COURSE NAME: FINANCIAL MARKETS & QUANTITATIVE METHODS
L-T-P-S: 3-0-3-0
PRE-REQ: SCDS

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse Indian payment system architectures including UPI, IMPS, NEFT, RTGS, and card schemes (Visa, Mastercard, RuPay) to map participant roles, message flows, settlement timing, and economic models.	PO1, PO2	4
CO2	Design double-entry ledger systems in software with immutability, idempotency, and consistency guarantees suitable for production money movement.	PO1, PO2	5
CO3	Implement payment gateway integrations (Razorpay, Stripe, PayU) including signed webhooks, idempotent operations, retries, refunds, and partial captures with proper failure handling.	PO2, PO3	5
CO4	Examine reconciliation systems comparing internal ledger to external statements (acquirer files, bank statements) including break investigation workflows and exception handling.	PO2, PO3	4
CO5	Architect KYC, fraud-detection, and risk-scoring pipelines that meet regulatory requirements while maintaining acceptable customer onboarding and approval rates.	PO4, PO5	5
CO6	Build a working payments slice including merchant onboarding, payment acceptance, ledger booking, settlement to merchant bank, reconciliation, and merchant-facing dashboards.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Payment Systems Landscape** Mapped: **CO1, CO5** · 6 hrs • The four-corner model of card payments (cardholder, issuer, acquirer, merchant)

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

	• card networks (Visa, Mastercard, RuPay, Amex, Discover) • authorisation, clearing, settlement (the acquirer side) • chargeback lifecycle • interchange and merchant discount rate (MDR) economics • Indian payment systems — UPI architecture (NPCI, PSP banks, payer/payee VPA, push vs pull, UPI 2.0/Lite/AutoPay), IMPS, NEFT, RTGS • cross-border (SWIFT, ACH, FedNow, SEPA Instant) • ISO 8583 (card messages) and ISO 20022 (modern XML/JSON-based standard) overview • payment system regulation in India (RBI Payment & Settlement Systems Act, NPCI circulars). *Workshop: students take a sample e-commerce transaction (₹2500 UPI vs ₹2500 credit card vs ₹2500 RTGS) and produce 3 sequence diagrams showing all participants and messages from initiation to settlement; identify economic flows.*
M2	**Double-Entry Ledger Systems** Mapped: **CO2, CO4** · 6 hrs • Why double-entry exists (every debit has a credit • the universe of accounts always balances) • chart of accounts design for fintech (assets, liabilities, equity, revenue, expense, contra accounts) • ledger immutability vs mutability (event-sourced ledgers vs row-update) • transaction journal entries • account balance calculation (snapshot + delta vs running balance) • idempotency keys for money movement (the core safety property) • the SAGA pattern for distributed money movement • databases for ledgers (PostgreSQL with strong constraints, TigerBeetle, single-writer ledgers) • the read-your-write problem at financial scale • precision (use integers for paise, never floats for money). *Lab: design and implement a minimal double-entry ledger in PostgreSQL — accounts table, journal entries table, idempotency keys table; implement deposit, transfer, refund operations with full ACID guarantees; write tests that intentionally produce double-spends and verify they fail.*
M3	**Payment Gateway Integration** Mapped: **CO3, CO6** · 6 hrs • Payment gateway architecture (the acquirer's interface to merchants) • Razorpay / Stripe / PayU API patterns (test mode, sandboxes) • orders / payments / refunds / disputes objects

31/7/26

HoD-CSA

Dr. K. BHARGAVI
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225(

	• checkout flows (hosted page, embedded, headless) • 3D Secure 2 (challenge flows, frictionless) • UPI-specific flows (intent vs collect, deep links to PSP apps) • recurring payments (eMandate, UPI AutoPay) • webhook handling (signature verification — HMAC-SHA256, replay protection, idempotency, retry semantics, deliver-at-least-once) • reconciling webhook absence (polling fallback) • settlement T+1, T+2 cycles to merchant bank. *Lab: integrate Razorpay test mode end-to-end — create order via API, complete payment via test card and test UPI, handle webhook with signature verification, persist to ledger with idempotency, refund via API, verify ledger updates.*
M4	**Reconciliation & Settlement** Mapped: **CO4, CO6** · 6 hrs • Why reconciliation matters (your book vs the bank's book diverging silently is how money disappears) • the recon dataset (internal transactions × external statement) • matching logic (exact match by reference, amount + date fuzzy match, many-to-many splits) • break categories (timing differences, missing on one side, amount mismatch, duplicate) • break ageing • auto-resolution rules vs manual investigation queue • ops dashboards for finance teams • settlement file formats (CSV, MT940, BAI2, ISO 20022 camt.053) • netting vs gross settlement • merchant fee calculation and statement generation • tax handling (GST on MDR, TDS). *Lab: implement a reconciliation engine in Python — given a CSV settlement file from acquirer (provided sample) and the internal ledger from M2, identify matched transactions, breaks by category, and produce a recon report; build a tiny ops UI in Streamlit for break investigation.*
M5	**KYC, Fraud, and Modern Fintech Patterns** Mapped: **CO5, CO6** · 6 hrs • KYC architecture in India (Aadhaar eKYC consent flow, DigiLocker, V-CIP video KYC, RBI Master Directions on KYC) • document verification vendors (Karza, IDfy, HyperVerge, Signzy, Gridlines) • CKYC registry (CERSAI) • periodic re-KYC • sanctions screening (UN, OFAC, FIU-IND lists) • PEP screening

21/7/26
Dr. **HoD-CSA**GAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• transaction monitoring (rule-based + ML scoring) • fraud patterns (account takeover, friendly fraud, BIN attacks, sanctioned wallet, mule accounts) • device fingerprinting • behavioural biometrics • chargeback management workflows. Modern fintech patterns: Banking-as-a-Service (Solid, Karbon, Decentro, Setu), Account Aggregator framework (Sahamati, Setu AA, Finvu), Open Credit Enablement Network (OCEN), embedded finance, Buy Now Pay Later (BNPL), neobanks, India Stack components. *Lab: integrate Setu DigiLocker or HyperVerge sandbox for KYC; implement a tiny fraud-rule engine evaluating 5 risk rules (velocity, geo-mismatch, amount threshold, blacklist, device fingerprint); produce risk score per transaction.*

Reference Books:

- **Hill, J. — *Fintech and the Remaking of Financial Institutions*, Academic Press, 2018.**
- **King, B. — *Bank 4.0: Banking Everywhere, Never at a Bank*, Wiley, 2018.**
- **Gleeson, S. — *The Legal Concept of Money*, Oxford, 2018.**
- **Stripe Documentation + Razorpay Documentation (free, primary references for integration patterns).**
- **RBI Master Directions on KYC + Payment Aggregators + Payment Systems (free, primary).**
- **NPCI UPI Procedural Guidelines (free, primary).**
- **Atomic Accounting & TigerBeetle Documentation (free, ledger architecture wisdom).**

Stream E: Data Science

COURSE CODE: 26CA32E3E
COURSE NAME: BIG DATA & DISTRIBUTED ANALYTICS
L-T-P-S: 3-0-3-0 **PRE-REQ: DEAS**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse the architecture of distributed query engines (Spark, Trino/Presto, Dremio, DuckDB) including planner, optimiser, executor, and shuffle layers to predict performance on representative workloads.	PO1, PO2	3

3/11/26
Dr. K. BHASKAVA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur [illegible] 522502.

HoD-CSA

CO2	Design Apache Spark applications using DataFrame and Structured Streaming APIs with explicit awareness of partitioning, broadcast joins, skew, and Adaptive Query Execution (AQE).	PO1, PO2	3
CO3	Examine columnar storage formats (Parquet, ORC, Arrow), file layout strategies, partitioning, bucketing, and Z-ordering to optimise query patterns on lakehouse storage.	PO2, PO3	4
CO4	Implement OLAP query patterns at scale using analytical databases (ClickHouse, Apache Druid, Apache Pinot, StarRocks) for sub-second analytical queries on billion-row datasets.	PO2, PO3	4
CO5	Architect graph analytics pipelines using property graph databases (Neo4j) and distributed graph engines (GraphX, GraphFrames, NetworkX at scale) for centrality, community detection, and pathfinding queries.	PO4, PO5	5
CO6	Build a benchmark suite comparing distributed query engines on TPC-H or TPC-DS workloads at sample scale, measuring latency, throughput, cost, and operational complexity to guide engine selection.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Distributed Query Engines** Mapped: **CO1, CO6** · 6 hrs • Why distributed (single machine TB-scale limits, cost economics) • MapReduce historical context • Apache Spark architecture (driver, executors, cluster manager YARN/Kubernetes/standalone, RDD foundations, DataFrame/Dataset DSL) • Trino/Presto architecture (coordinator, workers, push-based shuffle) • query lifecycle (parser → analyser → logical optimiser Catalyst → physical plan → execution) • cost-based optimisation • statistics in modern engines • modern federated query engines (Trino, Dremio, Starburst) • embedded analytical engines (DuckDB and the single-node renaissance) • when not to use Spark (the < 1TB working set rule). *Lab: install local Spark + Trino + DuckDB; load TPC-H 1GB into Iceberg; run the same query (`q1` from TPC-H) on all three; compare query plans, execution times, and resource usage; produce a one-page write-up.*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M2	**Module 2 — Spark in Depth** Mapped: **CO2, CO3** · 6 hrs • DataFrame API patterns (reading, transformations, actions) • the lazy evaluation model • narrow vs wide transformations • the shuffle and its cost • partitioning strategies (repartition, coalesce, partitionBy on write) • broadcast joins (when, how big) • sort-merge joins • bucket joins • data skew handling (salting, AQE skew join optimisation) • Adaptive Query Execution (AQE — coalesce shuffle partitions, dynamic skew join, dynamic BHJ) • caching strategies (persist levels, when to cache) • window functions • Structured Streaming (micro-batch model, watermarks, output modes append/update/complete, stateful aggregations, deduplication) • Spark on Kubernetes • Databricks Photon engine overview. *Lab: take a deliberately skewed dataset (one key has 90% of records); demonstrate join performance without AQE; enable AQE skew join optimisation; benchmark improvement; tune partitioning to validate further gains.*
M3	**Storage Formats & Lakehouse Performance** Mapped: **CO3, CO6** · 6 hrs • Why columnar storage (compression, predicate pushdown, projection pushdown) • Parquet internals (row groups, column chunks, pages, statistics, dictionary encoding) • ORC internals • Apache Arrow as columnar in-memory standard • partitioning vs bucketing vs Z-ordering vs liquid clustering (Databricks) • Iceberg / Delta Lake / Hudi performance characteristics • file size tuning (the small-file problem and its many cures: OPTIMIZE, compaction, auto-optimize) • statistics and data skipping • Iceberg hidden partitioning and partition evolution • query patterns that benefit from each layout choice • vectorised execution (Photon, Velox, DuckDB, ClickHouse columnar engine). *Lab: take a 100M-row dataset; create the same data as Parquet partitioned by date, partitioned by date + bucketed by user, and Iceberg with hidden partitioning; benchmark 5 representative queries on each layout; produce a layout decision matrix.*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

M4	**Real-Time OLAP Engines** Mapped: **CO4, CO6** · 6 hrs • Why specialised analytical DBs (sub-second on billion rows where Spark/Trino are seconds-to-minutes) • ClickHouse architecture (MergeTree engine, columnar storage, materialised views, sharding + replication) • Apache Druid (segment-based, real-time + historical, time-series-optimised) • Apache Pinot (LinkedIn-born, low-latency real-time analytics on Kafka) • StarRocks (MPP, MySQL-protocol-compatible) • when each shines (ClickHouse for general analytics, Druid for time-series with high cardinality, Pinot for sub-second user-facing) • ingestion patterns (batch INSERT, real-time Kafka ingestion, materialised view incremental refresh) • the user-facing analytics problem (low latency, high concurrency, fresh data) • hybrid HTAP systems (TiDB, SingleStore) as alternative. *Lab: install ClickHouse via Docker; ingest 100M-row sample dataset; create ReplacingMergeTree and SummingMergeTree variants; build a materialised view for hourly aggregations; demonstrate sub-second queries on the user-facing dashboard scenario.*
M5	**Graph Analytics & Modern Distributed Patterns** Mapped: **CO5, CO6** · 6 hrs • Graph data models (property graph, RDF triple) • Neo4j Cypher query language • common graph algorithms (PageRank, betweenness centrality, community detection — Louvain/Label Propagation, shortest path, cycle detection, weakly connected components) • GraphX and GraphFrames in Spark • distributed graph computation patterns • graph use cases (fraud rings, recommendation, knowledge graphs, supply chain) • the case for/against graph DBs vs relational • modern distributed patterns — vectorised execution everywhere (Velox, DuckDB, ClickHouse), the GPU analytics revolution (BlazingSQL, Theseus, NVIDIA RAPIDS) • cost-performance benchmarks (Snowflake vs BigQuery vs Databricks SQL vs ClickHouse Cloud — published TPC-H/TPC-DS results). *Lab: load a sample graph (e.g., DBLP co-authorship, Twitter ego network) into Neo4j; run PageRank, Louvain community detection, and shortest path queries; compare with Spark GraphFrames implementation of PageRank on the same data.*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Reference Books:

- Karau, H. & Warren, R. — *High Performance Spark*, O'Reilly, 2017 (still relevant on tuning).
- Damji, J. et al. — *Learning Spark* (2nd ed.), O'Reilly, 2020 (current Spark).
- Reis, J. & Housley, M. — *Fundamentals of Data Engineering*, O'Reilly, 2022 (architecture chapters).
- Dean, J. & Ghemawat, S. — *MapReduce: Simplified Data Processing on Large Clusters*, OSDI, 2004 (foundational paper).
- Apache Spark + Trino + ClickHouse + Iceberg documentation (free, primary).
- DuckDB documentation and blog (free, modern single-node analytics primer).
- Damji, J. — *The Path to Photon: Vectorized Spark Engine*, Databricks Engineering Blog (free).

OPEN ELECTIVE - 1 & 2

Business School

COURSE NAME: FINANCIAL ACCOUNTING FOR TECH FOUNDERS
L-T-P-S: 3-0-3-0
PRE-REQ: NIL

CO. No	Course Outcome Description	PO MAPING	BTL
CO1	Apply double-entry bookkeeping principles to record day-to-day business transactions in journals, ledgers, and trial balances.	PO1, PO2	3
CO2	Apply Indian Accounting Standards (Ind AS) and GST rules to prepare basic financial statements: P&L, balance sheet, and cash flow statement.	PO1, PO6	3
CO3	Analyze financial statements using ratio analysis (liquidity, profitability, solvency, efficiency) to assess company health and performance.	PO2, PO4	4
CO4	Examine startup-specific financial concepts: burn rate, runway, gross margin, unit economics, ARR/MRR, and how investors read these from financials.	PO2, PO8	4
CO5	Apply tools like Tally Prime / Zoho Books / QuickBooks (or Excel) to maintain books for a small entity through one quarter of mock	PO5, PO8	3

3/11/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

	transactions.		
CO6	Examine compliance basics: GST returns (GSTR-1, GSTR-3B), TDS, advance tax, ROC filings, and penalties for non-compliance — relevant for startup founders.	PO6, PO10	4

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Accounting Foundations** Mapped: **CO1** · 10 hrs • Double-entry principles • T-accounts • journals and ledgers • trial balance • the accounting equation • debit-credit rules • common transaction patterns • Indian context (rupee-denominated) • cash vs accrual accounting basics *Daily journal-entry practice; weekly quizzes on transaction recording*
M2	**Module 2 — Financial Statements** Mapped: **CO2** · 10 hrs • P&L preparation • balance sheet structure • cash flow statement (direct vs indirect) • Indian Schedule III format • basic Ind AS rules • revenue recognition • depreciation • inventory valuation • intro to GST: input/output, types of supply, GST rates

3/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh 522502.

	Weekly statement preparation exercises with corrected solutions
M3	**Module 3 — Financial Statement Analysis** Mapped: **CO3** · 8 hrs • Liquidity ratios (current, quick) • profitability ratios (gross margin, operating margin, net margin, ROCE, ROE) • solvency ratios (D/E) • efficiency ratios (inventory turnover, AR days) • horizontal and vertical analysis • comparing across firms and across years *Analyze published financials of 3 listed Indian companies (e.g., TCS, Zomato, Reliance)*
M4	**Module 4 — Startup Finance** Mapped: **CO4, CO5** · 10 hrs • Burn rate and runway calculations • gross margin in software vs hardware vs services • unit economics (CAC, LTV, payback period) • ARR/MRR tracking • SaaS metrics dashboards • convertible notes vs SAFE vs priced rounds • cap tables • how investors read founder financials *Build a SaaS-startup financial model in Excel/Sheets across 36 months*
M5	**Module 5 — Compliance & Tools** Mapped: **CO5, CO6** · 7 hrs • GST returns (GSTR-1 monthly, GSTR-3B monthly, GSTR-9 annual) • TDS basics • Advance tax for businesses • ROC filings (DIR-3 KYC, MGT-7, AOC-4)

HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)

	• typical startup penalties • tools: Tally Prime, Zoho Books, QuickBooks for small businesses *Hands-on Tally Prime / Zoho Books for a mock startup (1 quarter of transactions)*

Reference Books:

#	REFERENCE
1	Tulsian, P.C. — Financial Accounting (latest ed.), Pearson India
2	Maheshwari, S.N. — Introduction to Accountancy (12th ed.), Vikas 2018
3	ICAI — Study Material on Ind AS (free, official)
4	Berk, J., DeMarzo, P. — Corporate Finance (5th ed.), Pearson 2020 (selected chapters)
5	Mulford, C., Comiskey, E. — The Financial Numbers Game, Wiley 2002 (for spotting accounting tricks)
6	ICAI — Practical Guide to GST (latest annual edition, free download)

COURSE NAME: MARKETING &BRAND BUILDING

L-T-P-S: 3-0-3-0

PRE-REQ: NIL

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply marketing fundamentals — segmentation, targeting, positioning (STP) — to define a clear go-to-market strategy for a product or service.	PO1, PO2	3
CO2	Examine the 4Ps (Product, Price, Place, Promotion) and 7Ps (adding People, Process, Physical Evidence) frameworks and apply them to product launch decisions.	PO1, PO2	3
CO3	Analyze digital marketing channels (SEO, SEM, social media, content marketing, email, performance marketing) and select appropriate channel mix for given audiences and budgets.	PO2, PO3	4
CO4	Apply brand-building frameworks (Aaker's Brand Identity, Kapferer's Prism, Jobs-to-be-Done) to develop coherent brand positioning and	PO2, PO3	4

31/7/26

HoD-CSA

Dr. K. SRINIVASAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K L Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	messaging.		
CO5	Examine consumer behavior fundamentals — the buying funnel (AIDA, AARRR), customer journey mapping, behavioral economics in marketing, B2B vs B2C dynamics.	PO4, PO5	5
CO6	Architect a complete marketing plan for a real or simulated product, including STP, brand identity, content strategy, channel mix, and measurement plan.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Marketing Foundations & STP** Mapped: **CO1, CO5** · 10 hrs • Marketing definition and evolution (1.0 → 5.0 per Kotler) • STP framework • demographic, psychographic, behavioral, geographic segmentation • targeting strategies (concentrated, differentiated, undifferentiated) • positioning maps • perceptual mapping • competitive positioning *Each student picks a real Indian brand and produces an STP analysis; peer-review across teams*
M2	**Module 2 — The Marketing Mix (4Ps/7Ps)** Mapped: **CO2** · 10 hrs • Product: levels, lifecycle, classifications • Price: cost-plus, value-based, penetration, skimming, dynamic, freemium pricing • Place: distribution channels, omnichannel • Promotion: advertising, PR, sales promotion, direct marketing • Service marketing: 7Ps (People, Process, Physical Evidence) *Apply 7Ps to a service brand (Swiggy, Zomato, Razorpay); develop a launch plan for*

HoD-CSA 31/7/24

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

	a new product
M3	**Module 3 — Digital Marketing & Performance** Mapped: **CO3** · 10 hrs • SEO basics (on-page, off-page, technical) • SEM (Google Ads): bidding, quality score, ad rank • Social media marketing: organic vs paid • performance marketing: CPC, CPM, CPA, ROAS • influencer marketing • content marketing • email automation • marketing analytics: GA4 fundamentals *Run a small ad campaign with mock budget on chosen platform; report results with metrics*
M4	**Module 4 — Brand Strategy** Mapped: **CO4** · 8 hrs • What is a brand • brand vs marketing • brand identity (Aaker, Kapferer Brand Prism) • brand archetypes • brand voice and tone • visual identity systems (logo, color, typography) • naming conventions • brand storytelling • case studies of strong Indian brands (Tata, Tanishq, Amul, Boat, FreshToHome, Mamaearth) *Develop a brand identity for a fictional Indian D2C brand; present brand book*
M5	**Module 5 — Consumer Behavior& Indian Context** Mapped: **CO5, CO6** · 7 hrs • Buying funnels (AIDA, AARRR) • customer journey mapping

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• touchpoints • purchase decision frameworks • behavioral economics in marketing (anchoring, social proof, scarcity, framing) • India-specific consumer dynamics (Bharat vs India, tier 2/3, vernacular content, mobile-first) • D2C wave in India *Design a customer journey map and craft growth-loop strategy*

Reference Books:

#	REFERENCE
1	Kotler, P., Keller, K.L. — Marketing Management (15th ed., Indian context), Pearson 2017
2	Kotler, P. — Marketing 5.0: Technology for Humanity, Wiley 2021, ISBN 978-1119668510
3	Aaker, D. — Aaker on Branding, Morgan James 2014, ISBN 978-1614488323
4	Heath, C., Heath, D. — Made to Stick, Random House 2007, ISBN 978-1400064281
5	Cialdini, R. — Influence: The Psychology of Persuasion (Revised), Harper Business 2021, ISBN 978-0062937650
6	Sridhar, B. — Build, Don't Talk: Things You Need to Know Before You Get Real, Penguin 2022 (Indian D2C context)

COURSE NAME: BEHAVIORAL ECONOMICS & DECISION SCIENCES
L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply foundational behavioral economics concepts (heuristics, biases, bounded rationality, prospect theory) to analyze real-world decision-making patterns.	PO1, PO2	3
CO2	Examine cognitive biases (confirmation, availability, anchoring, framing, sunk cost, hindsight, status quo) and identify their effects in product	PO1, PO2	4

3/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

	design, policy, and personal decisions.		
CO3	Apply nudge theory (Thaler & Sunstein) to design choice architecture that improves outcomes without restricting freedom of choice.	PO2, PO3	3
CO4	Analyze applications of behavioral economics in product design (defaults, friction, loss aversion), public policy (savings programs, organ donation), and behavior change (health, sustainability).	PO2, PO3	4
CO5	Examine ethical considerations in applying behavioral insights — the line between nudging and dark-pattern manipulation, transparency, autonomy, and informed consent.	PO4, PO5	4
CO6	Architect a behavioral intervention proposal for a chosen problem (product, policy, or personal) with measurable hypotheses and an experimental design framework.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Foundations of Behavioral Economics** Mapped: **CO1** · 10 hrs • Classical economics vs behavioral economics • bounded rationality (Simon) • System 1 vs System 2 thinking (Kahneman) • prospect theory (Kahneman & Tversky) • loss aversion (~2x) • endowment effect • status quo bias • mental accounting • framing effects • reference dependence *Read Kahneman's Thinking Fast and Slow (Part I-II); weekly self-experiment journals on cognitive biases*

3/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

M2	**Module 2 — Catalog of Cognitive Biases** Mapped: **CO2** · 10 hrs • Confirmation bias • availability heuristic • anchoring and adjustment • representativeness heuristic • sunk cost fallacy • hindsight bias • planning fallacy • overconfidence • halo effect • survivorship bias • recency bias • gambler's fallacy • cognitive load and decision fatigue *Each student presents one bias with a real-world case study; class discussion*
M3	**Module 3 — Choice Architecture & Nudge Theory** Mapped: **CO3** · 8 hrs • Defaults and opt-in vs opt-out (Save More Tomorrow, organ donation registries) • friction as a design tool • salience and framing • social proof in nudges • precommitment devices • choice overload (paradox of choice) • MINDSPACE framework • SaveAct nudges *Design a nudge for a behavior the student wants to change in their own life; document outcomes*
M4	**Module 4 — Applications: Product, Policy, Behavior Change** Mapped: **CO4** · 10 hrs • Product design: defaults that drive engagement (Netflix autoplay), pricing anchors, social proof in checkout

31/7/24
HoD-CSA

	• Policy: pension auto-enrollment in UK, organ donation in Spain, India's APY • Behavior change: smoking cessation, weight loss, financial literacy, climate behavior • Marketing applications and consumer protection • UX dark patterns *Case studies: India's Jan Dhan Yojana, Direct Benefit Transfer; redesign exercise of a flawed app/policy*
M5	**Module 5 — Ethics & Experimental Design** Mapped: **CO5, CO6** · 7 hrs • Nudge vs sludge vs manipulation (Thaler 2018) • transparency and autonomy • the EAST framework (Easy, Attractive, Social, Timely) • A/B testing fundamentals • behavioral interventions in policy: RCT design (J-PAL approach) • ethical considerations • Indian behavioral units (BIU at NITI Aayog) *Design a behavioral intervention proposal with hypotheses, mechanism, and measurement plan*

Reference Books:

#	REFERENCE
1	Kahneman, D. — Thinking, Fast and Slow, Farrar, Straus and Giroux 2011, ISBN 978-0374533557
2	Thaler, R., Sunstein, C. — Nudge: The Final Edition, Yale 2021, ISBN 978-0143137016
3	Ariely, D. — Predictably Irrational (Revised), Harper Perennial 2010, ISBN 978-0061854545
4	Banerjee, A., Duflo, E. — Poor Economics, PublicAffairs 2011 (RCT-driven economics)
5	Sunstein, C. — Choosing Not to Choose, Oxford 2015
6	World Bank — Mind, Society, and Behavior World Development Report 2015 (free PDF)

31/1/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

COURSE NAME: ORGANIZATIONAL BEHAVIOR & PEOPLE MANAGEMENT
L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply foundational concepts of organizational behavior — individual differences (personality, perception, motivation), group dynamics, and organizational culture — to analyze workplace situations.	PO1, PO2	3
CO2	Examine major motivation theories (Maslow, Herzberg, McClelland, Self-Determination Theory) and apply them to design work environments that engage technology professionals.	PO1, PO2	4
CO3	Analyze team dynamics through models like Tuckman's stages (forming-storming-norming-performing), Lencioni's Five Dysfunctions, and Google's Project Aristotle findings on psychological safety.	PO2, PO3	4
CO4	Apply leadership frameworks — Situational Leadership (Hersey-Blanchard), Transformational vs Transactional leadership, Servant Leadership — to varied workplace contexts.	PO2, PO3	3
CO5	Examine conflict management styles (Thomas-Kilmann), negotiation principles (Fisher & Ury), and difficult conversations frameworks for workplace effectiveness.	PO4, PO5	4
CO6	Examine the Indian workplace context including hierarchy norms, indirect communication, family-business dynamics, gender at work, and cross-cultural collaboration in MNCs and global teams.	PO4, PO5	4

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Individual Behavior in Organizations** Mapped: **CO1, CO2** · 9 hrs • Personality (Big Five, MBTI critique) • perception and attribution • motivation theories: Maslow's hierarchy, Herzberg's two-factor (hygiene + motivators), McClelland's needs (achievement/affiliation/power), Self-

HoD-CSA 3/7/24
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

	Determination Theory (autonomy/competence/relatedness — Deci & Ryan) • intrinsic vs extrinsic motivation • flow (Csíkszentmihályi) *Personality assessments + reflection; case discussions of motivation in tech companies (Google, Microsoft, Indian startups)*
M2	**Module 2 — Group Dynamics & Team Effectiveness** Mapped: **CO3** · 10 hrs • Group vs team distinction • Tuckman's stages (forming-storming-norming-performing-adjourning) • group cohesion • groupthink • social loafing • Lencioni's Five Dysfunctions of a Team • Google's Project Aristotle findings (psychological safety as #1 predictor) • team types: software engineering teams, cross-functional teams, distributed teams *Team simulation exercises; case studies of high-performing tech teams; analysis of team failures (Boeing 737 MAX, Wells Fargo)*
M3	**Module 3 — Leadership** Mapped: **CO4** · 10 hrs • Trait theories vs behavioral theories • Situational Leadership (Hersey-Blanchard) • Transformational leadership (Bass) • Servant Leadership (Greenleaf) • Authentic Leadership • leadership in tech contexts (Camille Fournier's Manager's Path framework) • Indian leaders' styles (Narayana Murthy, Indra Nooyi, Ratan Tata, Falguni Nayar) *Leadership case-study analysis; role-play exercises; self-leadership reflection*
M4	**Module 4 — Conflict, Negotiation & Communication** Mapped: **CO5** · 8 hrs • Sources of workplace conflict • Thomas-Kilmann conflict modes (avoid, accommodate, compete,

31/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	compromise, collaborate) • principled negotiation (Fisher & Ury BATNA, ZOPA, separate people from problem) • difficult conversations framework (Stone, Patton, Heen) • giving and receiving feedback (SBI model) • workplace etiquette across hierarchies *Negotiation simulations (paired exercises); difficult conversation role-plays; feedback practice*
M5	**Module 5 — Indian Workplace + Modern Organizational Issues** Mapped: **CO5, CO6** · 8 hrs • Indian workplace context: hierarchy norms (Hofstede's high power distance), face-saving, indirect communication, family-business dynamics • gender at work in India • remote work and hybrid • AI in HR (selection, evaluation) • diversity and inclusion • psychological safety in Indian engineering teams • D&I beyond optics • mental health at work • toxic workplace patterns *Discussion of recent Indian HR cases (Infosys layoffs, EY workplace pressure case, Indian tech compensation debates); position-paper assignment*

Reference Books:

#	REFERENCE
1	Robbins, S.P., Judge, T.A. — Organizational Behavior (Indian ed., 18th), Pearson 2019, ISBN 978-9353437985
2	Lencioni, P. — The Five Dysfunctions of a Team, Jossey-Bass 2002, ISBN 978-0787960759
3	Edmondson, A. — The Fearless Organization, Wiley 2018, ISBN 978-1119477242
4	Fisher, R., Ury, W., Patton, B. — Getting to Yes (Updated 3rd ed.), Penguin 2011, ISBN 978-0143118756

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

5	Stone, D., Patton, B., Heen, S. — Difficult Conversations (10th anniv.), Penguin 2010, ISBN 978-0143118442
6	Coyle, D. — The Culture Code, Bantam 2018, ISBN 978-0804176989
7	Pillai, R. — Caste Capital: A Study of Indian Workplaces, HarperCollins India 2022 (for honest Indian context)

Fine Arts & Visual Media

COURSE NAME: FILMMAKING & VISUAL STORYTELLING
L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply visual storytelling fundamentals — narrative structure (3-act, hero's journey, save the cat), character development, conflict, theme — to develop a short film concept.	PO1, PO2	3
CO2	Apply cinematography basics — composition (rule of thirds, leading lines, headroom), lighting (3-point, key/fill/back, motivated lighting), camera movement (dolly, pan, tilt, handheld), shot types (wide/medium/close-up) — using a smartphone or DSLR	PO1, PO2	3
CO3	Apply screenwriting conventions including formatting (industry standard 12pt Courier), dialogue craft, scene structure, and visual writing (show, don't tell) to produce a polished short script.	PO2, PO3	3
CO4	Apply editing fundamentals — continuity editing, montage, jump cuts, J/L cuts, pacing — using DaVinci Resolve, Premiere Pro, or CapCut to assemble a coherent narrative from raw footage.	PO2, PO3	3
CO5	Examine sound design fundamentals — diegetic vs non-diegetic sound, music selection, dialogue clarity, ADR, foley basics — and apply them to enhance emotional impact.	PO4, PO5	4
CO6	Architect a complete short film (3-5 minutes) from concept to delivery, demonstrating coherent integration of script, cinematography, editing, and sound design.	PO4, PO5	5

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Visual Storytelling & Screenwriting** Mapped: **CO1, CO3** · 10 hrs • Narrative structures (3-act, hero's journey, save the cat) • character development • conflict (internal vs external) • theme • visual writing (show don't tell) • screenwriting format (12pt Courier, scene headings, action lines, dialogue) • short film vs feature • the logline • treatment writing • Indian cinema's distinct narrative conventions *Each student writes a short logline + treatment + 5-page script; peer reviews*
M2	**Module 2 — Cinematography Foundations** Mapped: **CO2** · 11 hrs • Camera basics (sensor, exposure triangle: aperture/shutter/ISO) • composition (rule of thirds, leading lines, headroom, 180-degree rule) • lighting (3-point: key/fill/back • motivated lighting • high-key vs low-key) • shot types (extreme wide to extreme close-up) • camera movement (static, pan, tilt, handheld, dolly, gimbal) • shooting in available light *Hands-on shooting exercises with smartphone/DSLR; visual essay assignments mimicking famous shots*
M3	**Module 3 — Production Workflow** Mapped: **CO1, CO2** · 8 hrs • Pre-production: storyboarding, shot lists, scheduling, location scouting.

31/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

	casting, gear checklist • production: directing actors, working with crew (DOP, sound, etc.) • on-set protocols • safety and continuity • production reports • managing 1-day shoots vs 3-day shoots *Mock production planning exercise; storyboard for own script; gear test sessions*
M4	**Module 4 — Editing & Post-Production** Mapped: **CO4, CO5** · 10 hrs • Editing software intro (DaVinci Resolve preferred — free • Premiere Pro alternative • CapCut for mobile) • workflow: ingest, organize, rough cut, fine cut, sound mix, color grade, export • continuity editing • montage • jump cuts, J/L cuts • sound design (diegetic vs non-diegetic, music, dialogue) • royalty-free music sources • basic color grading *Edit a provided footage set into a 90-second narrative; sound design exercise; color grade exercise*
M5	**Module 5 — Indian Cinema & Career Pathways** Mapped: **CO5, CO6** · 9 hrs • Indian cinema landscape: Bollywood vs South cinema vs regional • OTT (Netflix, Prime, Hotstar) impact • short film festivals (Mumbai, Kerala, MAMI Film Festival) • YouTube/Instagram Reels as distribution • FTII, SRFTI alumni networks • entry pathways: assistant director, editor's assistant, content creator, advertising • new tools: AI-assisted editing, virtual production *Festival research; final film showcase + critique by industry guest*

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Reference Books:

	REFERENCE
1	McKee, R. — Story: Substance, Structure, Style and the Principles of Screenwriting, Methuen 1999, ISBN 978-0413715609
2	Murch, W. — In the Blink of an Eye (2nd ed.), Silman-James 2001, ISBN 978-1879505629
3	Brown, B. — Cinematography: Theory and Practice (3rd ed.), Routledge 2016, ISBN 978-0415716161
4	Field, S. — Screenplay: The Foundations of Screenwriting (Revised), Delta 2005, ISBN 978-0385339032
5	Truby, J. — The Anatomy of Story, Faber 2008, ISBN 978-0865479937
6	Mascelli, J. — The Five C's of Cinematography (Reissue), Silman-James 1998, ISBN 978-1879505414

COURSE NAME: PHOTOGRAPHY FOR DIGITAL MEDIA

L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply photography fundamentals — exposure triangle (aperture/shutter speed/ISO), focal length, depth of field, white balance — using a DSLR/mirrorless camera or modern smartphone.	PO1, PO2	3
CO2	Apply composition principles (rule of thirds, leading lines, golden ratio, framing, negative space, symmetry, depth) to create visually compelling images.	PO1, PO2	3
CO3	Apply lighting techniques for various conditions — golden hour, blue hour, harsh midday sun, indoor available light, single off-camera flash, studio strobes, smartphone LED panels.	PO2, PO3	3
CO4	Apply post-processing fundamentals using Adobe Lightroom (or Darktable / Snapseed for free alternatives) — exposure correction, color grading, dodging/burning, masking, batch processing.	PO2, PO3	3

31/7/26

HoD-CSA

HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

CO5	Examine photography genres and their conventions — portrait, street, documentary, landscape, food, product, event — and apply genre-appropriate techniques to specific assignments.	PO4, PO5	4
CO6	Architect a coherent photo essay (10-15 images) on a chosen theme demonstrating technical proficiency, compositional rigor, and storytelling depth.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Camera & Exposure** Mapped: **CO1** · 9 hrs • Camera anatomy (DSLR/mirrorless/smartphone) • exposure triangle: aperture (f-stops, depth of field, bokeh), shutter speed (motion blur, freeze, hand-held min speed), ISO (noise, dynamic range) • metering modes • exposure compensation • histogram • white balance • RAW vs JPEG • focusing modes (single, continuous, manual) • back-button focus *Daily shooting exercises with assigned exposure parameters; histogram reading practice*
M2	**Module 2 — Composition & Framing** Mapped: **CO2** · 10 hrs • Rule of thirds • leading lines • framing (natural frames) • negative space • symmetry • depth (foreground-midground-background) • patterns and breaking patterns

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• golden ratio and Fibonacci spiral • subject placement • camera angle (high, low, eye-level) • focal length effects (wide vs telephoto compression) *Daily themed shooting prompts; weekly portfolio reviews; analysis of master photographers (Cartier-Bresson, Raghu Rai, Steve McCurry, Dayanita Singh)*
M3	**Module 3 — Lighting** Mapped: **CO3** · 10 hrs • Natural light: golden hour, blue hour, midday harsh, overcast diffused • window light • available light indoors • introduction to off-camera flash • one-light setups for portraits • reflectors and diffusers • smartphone LED panels • understanding shadow direction • high-key vs low-key lighting *Outdoor lighting walk; one-light portrait session; smartphone lighting hack workshop*
M4	**Module 4 — Post-Processing** Mapped: **CO4** · 8 hrs • Lightroom workflow (Library → Develop → Export) • free alternatives (Darktable, Snapseed for mobile, Capture One starter) • basic adjustments: exposure, contrast, highlights, shadows, whites, blacks, clarity, vibrance, saturation • HSL color adjustments • tone curve • split toning / color grading • masking (subject, sky, brush) • presets • batch processing • export for web vs print *Process a provided RAW file from scratch; develop a personal preset; portfolio editing session*

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.

M5	**Module 5 — Genres & Photo Essay** Mapped: **CO5, CO6** · 8 hrs • Portrait photography (environmental, studio, street) • street photography (decisive moment, candid ethics, public photography law in India) • documentary and photojournalism • landscape (composition, time-of-day, panorama) • food photography (lighting, styling, smartphone vs camera) • product photography for e-commerce • event photography *Weekly genre-specific assignments: photo essay project on a chosen theme*

Reference Books:

#	REFERENCE
1	Peterson, B. — Understanding Exposure (4th ed.), Amphoto 2016, ISBN 978-1607748502
2	Cartier-Bresson, H. — The Mind's Eye: Writings on Photography and Photographers, Aperture 2005, ISBN 978-0893818753
3	Rai, R. — A Vision of India: The Photography of Raghu Rai, Steidl 2014
4	Kelby, S. — The Lightroom Classic Book for Digital Photographers (latest ed.), New Riders
5	Adams, A. — The Camera, The Negative, The Print (trilogy), New York Graphic Society, ISBN 978-0821221846
6	Sontag, S. — On Photography, Picador 2001, ISBN 978-0312420093
7	Bartholomew, P. — Outside In: A Tale of 3 Cities, Penguin India 2018

21/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-52250

Humanities & Liberal Arts

COURSE NAME: CREATIVE WRITING & NARRATIVE CRAFT

L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply fundamental craft elements of fiction — character, setting, plot, conflict, point-of-view, voice, dialogue, scene vs summary — to write coherent short narratives.	PO1, PO2	3
CO2	Apply fundamental craft elements of creative non-fiction — personal essay, memoir, literary journalism, travel writing — to produce work grounded in lived experience and observation.	PO1, PO2	3
CO3	Examine narrative structures in fiction and non-fiction (3-act, hero's journey, in medias res, frame narrative, vignette, mosaic) and apply structures appropriate to specific stories.	PO2, PO3	4
CO4	Apply revision techniques — global revision (structure, voice), line editing (sentence-level), proofreading — across multiple drafts to substantially improve prose.	PO2, PO3	3
CO5	Examine major Indian and global literary voices — across fiction, non-fiction, and essays — and identify craft moves that can be studied and applied.	PO4, PO5	4
CO6	Architect a substantial original creative work (short story, personal essay collection, or hybrid piece) of 3000-5000 words demonstrating range, voice, and craft.	PO4, PO5	5

HoD-CSA 3/1/26

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Fiction Foundations** Mapped: **CO1, CO3** · 10 hrs • What makes a story story? — character desire and obstacle • scene vs summary • show don't tell (specifically) • concrete sensory detail • point-of-view (1st, close 3rd, omniscient, 2nd) • free indirect discourse • voice and style • dialogue craft (subtext, attribution, beats) • pacing and tension • the short story arc *Daily writing prompts (200-500 words); weekly peer workshops on submitted drafts*
M2	**Module 2 — Non-Fiction Foundations** Mapped: **CO2** · 8 hrs • Personal essay structure (Montaigne tradition, modern essayists like Vivian Gornick, Phillip Lopate) • memoir vs autobiography • literary journalism (New Journalism: Tom Wolfe, Joan Didion • Indian: Aatish Taseer, Suketu Mehta, Pankaj Mishra) • travel writing • the lyric essay • truth in non-fiction • ethics of writing about real people *Personal essay drafts; analyze non-fiction craft in published essays*

3/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

M3	**Module 3 — Narrative Structure** Mapped: **CO3** · 8 hrs • Aristotelian 3-act • hero's journey (Campbell, Vogler) • in medias res • frame narrative • vignette structure • mosaic / fractured timeline • epistolary • multiple-POV • structure as meaning (Anne Lamott, Charles Baxter on form) • short story structures (Hemingway iceberg, Cheever's quiet revelation, Indian: Mahasweta Devi's stark realism) *Re-structure exercise on own draft; analyze structure of master stories*
M4	**Module 4 — Revision** Mapped: **CO4** · 8 hrs • What is revision (vs editing) • global revision (does this story work?) • structural revision (does the order serve the story?) • voice revision (is this voice consistent?) • line editing (sentence music, rhythm, word choice) • cutting (Strunk: 'omit needless words') • the writing residency mindset • reading drafts aloud • rest-and-return strategy • getting and giving useful feedback *Submit a complete draft; receive structured peer feedback; produce a fundamentally revised version*
M5	**Module 5 — Indian Literary Voices** Mapped: **CO5, CO6** · 9 hrs • Indian literary canon: Tagore (poetry, short stories), Premchand (Hindi-Urdu humanism), Mahasweta Devi (Adivasi voices), R.K. Narayan (small-town India), Salman Rushdie (magic realism), Arundhati Roy (lyric prose), Jhumpa Lahiri (diaspora), Vikram Seth (formal range), Aravind Adiga (satirical

3/7/26
HoD-CSA

Dr. K. RAVANA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	realism), Mirza Ghalib and Faiz Ahmed Faiz (poetry) • regional Indian writers across languages • Indian-English non-fiction (Pankaj Mishra, Suketu Mehta, Sonia Faleiro, Snigdha Poonam) *Read 4-6 short pieces by Indian authors; craft analysis essays; final project drafting*

Reference Books:

#	
1	Lamott, A. — Bird by Bird: Some Instructions on Writing and Life, Pantheon 1995, ISBN 978-0385480017
2	King, S. — On Writing: A Memoir of the Craft (10th anniv.), Scribner 2010, ISBN 978-1439193631
3	Burroway, J., Stuckey-French, E. — Writing Fiction (10th ed.), University of Chicago 2019, ISBN 978-0226616698
4	Gornick, V. — The Situation and the Story, FSG 2002, ISBN 978-0374528607
5	Strunk, W., White, E.B. — The Elements of Style (4th ed.), Pearson 2000
6	Lopate, P. — The Art of the Personal Essay (Anchor Reissue), Anchor 1995
7	Mehta, S. — Maximum City (anniversary ed.), Vintage 2014 (Indian non-fiction reference)
8	Roy, A. — The God of Small Things (Reissue), Random House India 2017

COURSE CODE: 26IE3201E
COURSE NAME: TERM PAPER
L-T-P-S: 0-0-9-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPPING	BTL
CO1	Apply systematic literature search techniques (Google Scholar, ACM Digital Library, IEEE Xplore, arXiv, Semantic Scholar) and evaluate the quality of academic sources.	P01, P05	3
CO2	Apply citation management (Zotero, Mendeley) and academic writing tools (LaTeX with Overleaf, BibTeX) to produce well-cited, properly	P03, P05	3

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-52250

	formatted papers.		
CO3	Analyze a research problem in the chosen specialization stream — formulate a research question, identify the gap in existing work, scope the contribution feasibly.	PO2, PO3	4
CO4	Analyze methodology choices for the contribution type — survey/review (PRISMA), empirical study (experimental design), comparative analysis (benchmark methodology), engineering case study.	PO2, PO4	4
CO5	Evaluate research ethics — plagiarism prevention (Turnitin / iThenticate), authorship credit, AI-assistance disclosure, data privacy, IRB requirements (where applicable).	PO6, PO10	5
CO6	Evaluate own and peer work through structured peer review, presentation, and revision cycles to produce a publication-standard paper.	PO4, PO9	5

Syllabus:

Module No.	Syllabus
M1	**Phase 1 — Topic Selection & Faculty Mentor Pairing** Mapped: **CO3 (BTL-4)** · 6 hrs · Gate: Topic approval • Choose specialization stream (S1-S5) — must align with your declared track • Identify 3 candidate problems within the stream • One-page brief per candidate: problem, why it matters, what already exists, what would be new • Pair with a faculty mentor (each faculty supervises 4-6 students) • Mentor approval of one final topic by end of Week 2 • Topic must be NEW for the student — not a rehash of a course project • Topic must be FEASIBLE in 12 weeks (one student, ~4 hrs/week) *Approved topic by end of Week 2 in the form of a 1-page "Topic + scope + intended contribution" document, signed by mentor.*

21/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M2	**Phase 2 — Literature Search & Synthesis** Mapped: **CO1, CO2 (BTL-3)** · 12 hrs · Gate: Literature matrix • Search strategy: keywords, Boolean operators, filters • Sources: Google Scholar, ACM Digital Library, IEEE Xplore, arXiv, Semantic Scholar • Forward and backward citation tracking • Source quality: peer-reviewed venues vs preprints vs blog posts • Citation manager setup (Zotero recommended; Mendeley acceptable) • Reading strategy: title → abstract → introduction → conclusion → method → results • Annotated literature matrix: each source × theme × contribution × limitation • 15-25 sources minimum; 30+ recommended for survey-type papers *Annotated literature matrix (spreadsheet) with 15-25 sources by end of Week 4. Zotero library exported and committed.*

M3	**Phase 3 — Research Question & Methodology Design**
	
	Mapped: **CO3, CO4 (BTL-4)** · 8 hrs · Gate: Methodology approval
	
	• Refine the research question based on literature gaps found
	• Pick contribution type:
	  ○ *Systematic review:* follow PRISMA 2020 protocol
	  ○ *Comparative analysis:* define metrics, baselines, evaluation protocol
	  ○ *Empirical study:* hypothesis, variables, experimental design
	  ○ *Engineering case study:* system description, design rationale, evaluation
	• Write a 2-page methodology section (often called "Approach" in CS papers)
	• Faculty mentor approval before proceeding to data collection / experiments
	
	Approved methodology document by end of Week 6. Includes refined research question, contribution type, and detailed approach.

M4	**Phase 4 — Execution & Data Collection** Mapped: **CO4 (BTL-4)** · 16 hrs · Gate: Results / data • Execute the methodology from Phase 3 • For surveys: read & categorize the literature systematically; build taxonomy • For comparative: implement / configure the baselines; run benchmarks • For empirical: collect data per the experimental design • For engineering: build the system, document design decisions • Data hygiene: version control results; reproducible scripts; raw data preserved

31/7/24

HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applicat
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

	• Mid-trimester check-in with mentor — adjust scope if needed *Results / data / artifact by end of Week 9. Reproducibility: another student should be able to re-run the analysis from the committed materials.*
M5	**Phase 5 — Drafting & Peer Review** Mapped: **CO2, CO6 (BTL-3/5)** · 12 hrs · Gate: Reviewed draft • IMRaD structure for empirical/comparative; modified for review papers • Standard sections: Abstract, Introduction, Related Work, Approach/Method, Results, Discussion, Limitations, Conclusion, References • LaTeX template (IEEE Conference, ACM Conference, or similar) on Overleaf • BibTeX with Zotero export • Figure/table standards: numbered, captioned, referenced in text, source cited • Peer review session: pair with classmate from the same stream; structured rubric • Address reviewer feedback in writing; revise sections accordingly *Reviewed draft (8-12 pages, double-column or 12-15 pages single-column) by end of Week 11. Includes peer review feedback document and revision response.*
M6	**Phase 6 — Revision, Plagiarism Check & Final Submission** Mapped: **CO5, CO6 (BTL-5)** · 6 hrs · Gate: Final paper accepted • Revise based on peer review and mentor feedback • Plagiarism check via Turnitin or iThenticate; institutional threshold (typically < 15%) • AI-assistance disclosure: explicit statement of which tools were used and how (per institute policy and emerging publisher norms) • Final formatting check: citations, captions, page count, references list • Submission to institutional repository (or selected venue if mentor recommends submission) • 10-minute presentation: research question, approach, key finding, limitations, future work • Q&A defense with 2-faculty panel *Final paper submitted by Week 12. Plagiarism report + AI disclosure committed. Presentation delivered. Optional: submission to a student conference (e.g., IEEE student venues, ACM Compute India).*

31/3/26
HoD-CSA
Dr. K. SAGAYAM
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	*Writing for Computer Science*	Justin Zobel	Springer	3rd Ed, 2014	978-1447166382
2	*The Craft of Research*	Wayne C. Booth, Gregory G. Colomb, Joseph M. Williams	University of Chicago Press	4th Ed, 2016	978-0226239736
3	*How to Write a Better Thesis*	David Evans, Paul Gruba, Justin Zobel	Springer	3rd Ed, 2014	978-3319042855
4	*Doing a Successful Research Project*	Martin Brett Davies	Palgrave Macmillan	2nd Ed, 2014	978-1137306425
5	*The LaTeX Companion*	Frank Mittelbach, Michel Goossens	Addison-Wesley	2nd Ed, 2004	978-1137306425
6	*PRISMA 2020 statement* (free)	Page et al.	BMJ Open Science	2021	—
7	*The Elements of Style*	William Strunk Jr., E. B. White	Pearson	4th Ed, 1999	978-0205309023

COURSE CODE: 26CA3325E
COURSE NAME: DEVSECOPS
L-T-P-S: 2-1-3-0 **PRE-REQ: DFSE**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply shift-left security principles — secure-by-default IDE settings, pre-commit hooks, secret scanning, dependency scanning — to catch vulnerabilities before code is merged.	PO1, PO2	3

31/1/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO2	Apply SAST (Semgrep, CodeQL, SonarQube), DAST (OWASP ZAP), and IaC scanning (Checkov, tfsec) within CI/CD pipelines.	PO1, PO2	3
CO3	Analyze software supply chain risks — SBOM generation (CycloneDX, SPDX), artifact signing (Sigstore, cosign), SLSA framework levels — and implement supply-chain controls.	PO2, PO3	4
CO4	Analyze policy-as-code patterns with Open Policy Agent (OPA) and Conftest to enforce security guardrails on Kubernetes manifests, Terraform, and CI workflows	PO2, PO3	4
CO5	Evaluate threat modeling methodologies (STRIDE, attack trees) and integrate threat models into design reviews and PR templates.	PO4, PO5	5
CO6	valuate runtime security controls — Kubernetes admission webhooks, Falco, distroless / minimal images, RBAC hardening — and propose a balanced controls portfolio.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Shift-Left Security & Secrets Management** Mapped: **CO1 (BTL-3)** · 8 hrs · Project Stage 1 — Pre-commit gates • The cost-of-fix curve: 1× in dev → 100× in prod • Secure defaults: IDE plugins (CodeQL, Snyk), .gitignore hygiene • Pre-commit framework: hooks for linting, formatting, secret scanning • Secret scanning: gitleaks, trufflehog, detect-secrets • What to do when secrets leak: rotate immediately, audit access, post-mortem • Secret management at runtime: Vault, AWS Secrets Manager, sealed-secrets • Environment-variable hygiene; avoiding secrets in build args • Dependency vulnerability scanning: npm audit, pip-audit, Trivy fs scan *SecurePipeline gains pre-commit hooks (gitleaks, dep scan, format). A test commit with a deliberately-leaked AWS key is blocked locally.*

31/7/25

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

M2	**Module 2 — SAST, DAST & IaC Scanning in CI/CD** Mapped: **CO2 (BTL-3)** · 10 hrs · Project Stage 2 — Pipeline gates • Static Application Security Testing (SAST): how it differs from linting • Tools: Semgrep (OSS), CodeQL (GitHub Advanced Security), SonarQube • Writing custom Semgrep rules for organization-specific patterns • Dynamic Application Security Testing (DAST): OWASP ZAP, Nuclei • Container image scanning: Trivy, Grype, Clair • IaC scanning: Checkov for Terraform, tfsec, Kubesec for k8s manifests • SCA (Software Composition Analysis): finding vulnerable dependencies — OSS Index, Snyk • Wiring all of these as gates in GitHub Actions *A pull request with a deliberate vulnerable dependency, a SQLi anti-pattern, and a misconfigured Terraform resource is blocked by 3 distinct CI gates.*
M3	**Module 3 — Software Supply Chain Security** Mapped: **CO3 (BTL-4)** · 8 hrs · Project Stage 3 — Signed & SBOM-d • The supply chain threat model: SolarWinds, log4shell, xz-utils — what happened • SLSA (Supply-chain Levels for Software Artifacts) framework: levels 1-4 • SBOM (Software Bill of Materials): CycloneDX, SPDX formats • Generating SBOMs: Syft, Trivy SBOM, Anchore tools • Sigstore + cosign: signing container images and binaries • Verifying signatures at deploy time (cosign verify) • OIDC-backed keyless signing for CI • Provenance attestations (in-toto, SLSA provenance) *Every container image SecurePipeline produces is signed with cosign and accompanied by a CycloneDX SBOM. EKS only deploys signature-verified images.*
M4	**Module 4 — Policy-as-Code & Threat Modeling** Mapped: **CO4, CO5 (BTL-4/5)** · 12 hrs · Project Stage 4 — Guardrails • Policy-as-code rationale: write rules once, enforce everywhere • Open Policy Agent (OPA): Rego language fundamentals

31/7/26

HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• Conftest for testing Kubernetes manifests, Terraform plans, Dockerfiles • OPA Gatekeeper for live admission control on Kubernetes • Common policies: no privileged containers, no :latest tags, required labels, image signature verification • Threat modeling methodologies: STRIDE, PASTA, attack trees • STRIDE deep-dive: Spoofing, Tampering, Repudiation, Info-disclosure, DoS, Elevation of privilege • Threat models as PR-template artifacts; design reviews *SecurePipeline has 5 OPA policies enforced in CI (conftest) + admission control (Gatekeeper). A STRIDE threat model is committed for one major component*
M5	**Module 5 — Runtime Security & Hardening** Mapped: **CO6 (BTL-5)** · 10 hrs · Project Stage 5 — Hardened runtime • Container hardening: distroless / scratch / Wolfi base images • Read-only root filesystem; non-root user; dropped capabilities • Pod security standards: restricted, baseline, privileged • Network policies (Calico / Cilium): default deny + explicit allow • Service mesh and mTLS: Istio, Linkerd briefly • Runtime threat detection: Falco rules, eBPF basics • Kubernetes RBAC hardening; audit logging • Cost vs control trade-offs: pick a portfolio *All SecurePipeline pods run as non-root with read-only filesystem and dropped capabilities. Falco watches for suspicious syscalls. RBAC follows least-privilege. A penetration attempt on the cluster triggers detection.*

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	*Securing DevOps: Security in the Cloud*	Julien Vehent	Manning	1st Ed, 2018	978-1617294136
2	*Container Security*	Liz Rice	O'Reilly	1st Ed, 2020	978-1492056706
3	*Kubernetes Security and Observability*	Brendan Creane, Amit Gupta	O'Reilly	1st Ed, 2021	978-1098107109

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

4	*Threat Modeling: Designing for Security*	Adam Shostack	Wiley	1st Ed, 2014	978-1118809990
5	*The DevSecOps Handbook*	Patrick Debois et al.	IT Revolution Press	2nd Ed, 2021	978-1942788133
6	*Practical Cloud Security*	Chris Dotson	O'Reilly	2nd Ed, 2023	978-1098148171
7	*OWASP Application Security Verification Standard (ASVS)* v4 (free)	OWASP Foundation	OWASP	v4, 2023	—

PROFESSIONAL ELECTIVE IV

Stream A: Applied AI & GenAI

COURSE CODE: 26CA33A4E
COURSE NAME: NATURAL LANGUAGE PROCESSING & LLM FOUNDATIONS
L-T-P-S: 3-0-3-0 **PRE-REQ: NIL**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse classical NLP foundations (tokenisation, embeddings, parsing, NER) to choose appropriate techniques for tasks where modern LLMs are overkill or unsuitable.	PO1, PO2	4
CO2	Examine transformer-based language model architectures (encoder-only BERT, decoder-only GPT, encoder-decoder T5) and pretraining objectives to compare their fitness for downstream tasks.	PO1, PO2	4
CO3	Design pretraining and fine-tuning pipelines including masked language modelling, causal language modelling, and instruction tuning for a small language model.	PO2, PO3	5
CO4	Implement alignment techniques (supervised fine-tuning, RLHF, DPO, constitutional AI) at small scale to shape model behaviour and reduce	PO2, PO3	5

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	harmful outputs.		
CO5	Evaluate language models using perplexity, MMLU, HELM, BIG-bench, and task-specific benchmarks to make informed model selection and reporting decisions	PO4, PO5	4
CO6	Architect a small-scale language model end-to-end including tokeniser training, pretraining on a domain corpus, instruction tuning, alignment, evaluation, and serving.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	Module 1 — Classical NLP & Tokenisation Mapped: CO1, CO5 · 6 hrs NLP pipeline (tokenisation, lemmatisation, POS, parsing, NER) regex and finite-state methods word embeddings (Word2Vec, GloVe, FastText) subword tokenisation (BPE, WordPiece, SentencePiece, Tiktoken) training a custom tokeniser multilingual tokenisation (Indian scripts) information extraction semantic similarity text classification baselines (TF-IDF + logistic regression, fastText) for cost-effective deployment.
M2	Module 2 — Modern Post-Training: SFT, DPO, RLHF Alignment Science Mapped: CO2, CO3 (BTL-4/5) · 6 hrs · Project Stage 2 Brief recap (assumes GENA foundations on instruction tuning, RAG, and basic SFT) Supervised Fine-Tuning at scale: dataset curation, quality filtering, contamination avoidance Direct Preference Optimisation (DPO): theory, advantages over RLHF, IPO/KTO variants Reinforcement Learning from Human Feedback: PPO mechanics, reward hacking, KL

29/7/26

HoD-CSA

Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	constraints Constitutional AI & RLAIF: scalable oversight without human labels Online vs offline alignment trade-offs Process reward models vs outcome reward models Recent open post-training recipes: Tülu 3, Llama 3 alignment, Qwen alignment
M3	Module 3 — Pretraining & Fine-Tuning Mapped: CO3, CO5 · 6 hrs Data curation for pretraining (Common Crawl, The Pile, RedPajama quality vs quantity) deduplication toxicity filtering pretraining at small scale (nanoGPT pattern) learning rate schedules mixed-precision and gradient accumulation full fine-tuning vs parameter-efficient methods (LoRA, QLoRA, prefix tuning, prompt tuning) instruction tuning datasets (Alpaca, Dolly, OpenAssistant) chat templates.
M4	Module 4 — Alignment & Safety Mapped: CO4, CO5 · 6 hrs Why alignment matters (helpful, harmless, honest) supervised fine-tuning (SFT) on instruction data reward modelling RLHF (PPO) high-level and challenges DPO as RLHF alternative (theory and TRL implementation) constitutional AI jailbreaks and prompt injection

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	red-teaming refusal training system prompts vs hardcoded refusals the alignment tax introduction to interpretability (attention probing, activation steering).
M5	Module 5 — Evaluation & Production Considerations Mapped: CO5, CO6 · 6 hrs Perplexity and what it does/does not measure standard benchmarks (MMLU, HellaSwag, GSM8K, HumanEval, MBPP, ARC, BBH, HELM, BIG-bench) LLM-as-judge methodology and its biases eval harness usage (lm-evaluation-harness) custom evals for application-specific tasks contamination concerns serving small LMs (vLLM, TGI, llama.cpp) quantisation (GPTQ, AWQ, GGUF, INT8/INT4) inference cost economics on-device LLMs.

Reference Books:

Sno	Title	Author(s)
1	Speech and Language Processing (3rd ed., free draft), Stanford	Jurafsky, D. & Martin, J. H.
2	Natural Language Processing with Transformers	Tunstall, von Werra, Wolf
3	Understanding Deep Learning	Prince, S.
4	Let's build GPT lecture series and nanoGPT codebase	Karpathy, A.

31/7/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

5	BERT (NAACL 2019), Brown et al. — GPT-3 (NeurIPS 2020), Touvron et al. — LLaMA papers (primary sources).	Devlin et al.
6	Hugging Face NLP Course	—
7	primary sources on alignment methodology.	Anthropic Alignment papers

Stream B: Cloud Native & SRE

COURSE CODE: 26CA33B4E
COURSE NAME: SERVERLESS & EDGE COMPUTING
L-T-P-S: 3-0-3-0 **PRE-REQ: CNIG**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse service mesh data plane and control plane architectures (Istio, Linkerd, Cilium) to evaluate latency, resource overhead, and operational complexity trade-offs.	PO1, PO2	4
CO2	Design service mesh traffic policies including mTLS, authorisation policies, traffic shifting, fault injection, and circuit breaking to enforce production-grade security and reliability without application code changes.	PO1, PO2	5
CO3	Examine event-driven architectural patterns (event notification, event-carried state transfer, event sourcing, CQRS) to choose the right pattern for each business domain.	PO2, PO3	4
CO4	Implement event streaming systems using Apache Kafka including partitioning, consumer groups, exactly-once semantics, schema evolution, and stream processing topologies	PO2, PO3	5
CO5	Compare event-driven communication backbones (Kafka, NATS, RabbitMQ, Redpanda, AWS EventBridge) to make justified choices based on throughput, ordering, durability, and operational requirements.	PO4, PO5	4
CO6	Architect a microservices system combining service mesh for synchronous communication and event streaming for asynchronous workflows with explicit consistency boundaries and failure handling.	PO4, PO5	5

31/7/26
HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Service Mesh Architecture Deep-Dive** Mapped: **CO1, CO2** · 6 hrs • Service mesh problem statement (cross-cutting concerns: mTLS, retries, observability, traffic shaping) • sidecar proxy pattern (Envoy as data plane) • control plane responsibilities • Istio architecture (istiod, ingress/egress gateways, sidecar injection) • Linkerd architecture (Rust-based linkerd-proxy, lightweight design) • Cilium service mesh (eBPF-based, sidecarless) • ambient mesh in Istio • performance benchmarks across meshes • when not to use a service mesh.
M2	***Module 2 — Mesh Traffic Management & Security*** ***Mapped: CO2, CO5 · 6 hrs*** • *Istio VirtualService and DestinationRule (request routing, canary, blue-green, traffic mirroring)* • *Linkerd traffic split with SMI* • *subset routing* • *fault injection (delays, aborts)* • *retry budgets and timeouts* • *circuit breaking (consecutive errors, max ejection)* • *mTLS strict mode* • *authorisation policies (allow/deny by source, namespace, JWT claims)* • *rate limiting at the mesh* • *egress controls* • *multi-cluster mesh patterns* • *gateway API (replacing Ingress)*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M3	***Module 3 — Event-Driven Patterns*** *Mapped:* ***CO3, CO6*** *· 6 hrs* • *Synchronous vs asynchronous communication trade-offs* • *event types (event notification, event-carried state transfer, event sourcing)* • *CQRS (command query responsibility segregation)* • *event sourcing (event store, projections, snapshots, replay)* • *choreography vs orchestration* • *saga pattern for distributed transactions (compensation, pivot transaction)* • *domain events vs integration events* • *event modelling workshops* • *eventual consistency and its implications* • *outbox pattern for reliable event publishing.*
M4	***Module 4 — Apache Kafka Deep-Dive*** *Mapped:* ***CO4, CO5*** *· 6 hrs* • *Kafka architecture (brokers, topics, partitions, replicas, ISR)* • *producer semantics (acks, idempotence, transactions, exactly-once)* • *consumer groups and rebalancing* • *offset management* • *partition assignment strategies* • *schema registry (Avro, Protobuf, JSON Schema)* • *schema evolution rules* • *Kafka Connect for integration* • *Kafka Streams and KSQL for stream processing* • *topic compaction* • *retention policies* • *capacity planning* • *KRaft mode (replacing ZooKeeper)* • *managed Kafka comparison (Confluent Cloud, MSK, Redpanda Cloud).*
M5	***Module 5 — Stream Processing & Alternative Brokers*** ***Mapped: CO4, CO5 · 6 hrs*** • *Stream processing fundamentals (windowing — tumbling, hopping, session* • *watermarks* • *late data handling)* • *Kafka Streams DSL (KStream, KTable)* • *stateful operations (joins, aggregations)* • *checkpointing*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	• *Apache Flink as alternative* • *alternative brokers — NATS (lightweight pub-sub, JetStream for persistence), RabbitMQ (AMQP, exchanges, queues), Redpanda (Kafka API drop-in, no JVM), AWS EventBridge / SNS-SQS / Kinesis comparison* • *choosing a broker by use case* • *event-driven testing strategies (test containers, contract tests with Pact, schema-driven testing).*

Reference Books:

Sno	Title	Author(s)
1	Istio: Up and Running, O'Reilly 2019.	Calcote, L. & Butcher, Z.
2	The Enterprise Path to Service Mesh Architectures, O'Reilly	Posta, C.
3	Designing Event-Driven Systems, O'Reilly 2018 (free, Confluent-published; canonical Kafka)	Stopford, B.
4	Implementing Domain-Driven Design, Addison-Wesley	Vernon, V.
5	CQRS Documents (free PDF, foundational on CQRS).	Young, G.
6	Building Microservices	Newman, S.
7	Confluent + Apache Kafka official documentation.	-----

29/7/26

HoD-CSA

Dr. K. BHARGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Stream C: Applied Cybersecurity

COURSE CODE: 26CA33C4E
COURSE NAME: SECURITY GOVERNANCE, RISK &COMPLIANCE
L-T-P-S: 3-0-3-0 **PRE-REQ: WAPS**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse major security frameworks (ISO 27001, NIST CSF 2.0, CIS Controls v8, SOC 2 Trust Services Criteria) to map their controls and identify the most appropriate framework for a given organisational context.	PO1, PO2	4
CO2	Design risk management programmes using NIST RMF, ISO 31000, and FAIR (quantitative) methodologies to translate technical vulnerabilities into business risk language executives can act on.	PO1, PO2	5
CO3	Examine sector-specific compliance regimes including PCI DSS v4.0, HIPAA, India CERT-In Directions, and RBI Cyber Security Framework to identify obligations applicable to a regulated workload.	PO2, PO3	4
CO4	Implement security policies, standards, and procedures (the policy hierarchy) that balance regulatory completeness with practical operability for a small-to-mid-size organisation.	PO2, PO3	5
CO5	Architect compliance automation pipelines using continuous-control-monitoring platforms (Drata, Vanta, Sprinto, Tugboat Logic) to replace manual evidence collection with engineering-driven attestation	PO4, PO5	5
CO6	Build third-party risk management (TPRM) programmes including vendor due diligence questionnaires, SOC 2 review processes, and contract security requirements that materially reduce supply-chain risk	PO4, PO5	5

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

Syllabus:

Module No.	Syllabus
M1	**Module 1 — GRC Foundations & Framework Landscape** Mapped: **CO1, CO4** · 6 hrs • GRC as a discipline (governance, risk, compliance distinguished) • the security programme operating model • the CISO function and reporting lines • ISO 27001:2022 structure (Annex A controls, ISMS lifecycle, Statement of Applicability) • ISO 27002:2022 implementation guidance • NIST Cybersecurity Framework 2.0 (Govern, Identify, Protect, Detect, Respond, Recover) • CIS Controls v8 (Implementation Groups 1/2/3) • SOC 2 Type I vs Type II, Trust Services Criteria (Security, Availability, Processing Integrity, Confidentiality, Privacy) • when to choose which framework (geography, customer base, sector, scale).
M2	***Module 2 — Risk Management Programmes*** ***Mapped: CO2, CO4 · 6 hrs*** • *Risk fundamentals (asset, threat, vulnerability, likelihood, impact)* • *qualitative risk assessment (NIST SP 800-30 5x5 matrices) and limitations* • *quantitative risk with FAIR (Loss Event Frequency × Loss Magnitude)* • *risk appetite, tolerance, and capacity* • *risk register hygiene* • *risk treatment options (mitigate, transfer, avoid, accept) and risk acceptance documentation* • *OKRs for risk reduction* • *communicating risk to executives (BLUF, expected loss in INR/USD)* • *risk vs control language* • *the difference between vulnerabilities, risks, and incidents.*
M3	***Module 3 — Event-Driven Patterns*** *Mapped: CO3, CO4 · 6 hrs* • *PCI DSS v4.0 deep-dive (12 requirements, scoping, network segmentation.*

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	customised approach, penetration testing requirements, SAQ types, Report on Compliance) • *HIPAA (Privacy Rule, Security Rule, Breach Notification Rule)* • *India CERT-In Directions of April 2022 (incident reporting within 6 hours, log retention 180 days, KYC for VPN/cloud providers — current status and operational impact)* • *RBI Cyber Security Framework for banks and NBFCs* • *SEBI Cybersecurity and Cyber Resilience Framework* • *sectoral DPDP Rules when notified* • *cross-border transfer compliance.*
M4	***Module 4 — Policies, Procedures & Compliance Automation*** *Mapped:* ***CO4, CO5*** *· 6 hrs* • *The policy hierarchy (policy → standard → procedure → guideline)* • *essential policy set for SOC 2 (Information Security Policy, Access Control, Change Management, Incident Response, Risk Management, Acceptable Use, BCP/DR, Vendor Management, Data Classification, Cryptography, Software Development)* • *writing policies that engineers will actually follow* • *policy lifecycle (annual review, exception process, attestation)* • *compliance automation platforms (Drata, Vanta, Sprinto, Secureframe) — control library, evidence collection, integrations with AWS/Okta/GitHub* • *mapping a single technical control to multiple frameworks (SOC 2 + ISO 27001 + PCI)* • *continuous control monitoring* • *policy-as-code (OPA) for technical controls.*
M5	***Module 5 — Stream Processing & Alternative Brokers*** • *Audit lifecycle (readiness assessment, gap analysis, remediation, audit fieldwork, report)* • *evidence quality (screenshots vs system-generated reports vs configuration exports)* • *auditor interaction skills* • *the difference between SOC 2 Type I (point in time) and Type II (operational effectiveness over 6-12 months)* • *third-party risk management programmes (vendor inventory, tiering by criticality, due diligence questionnaires — SIG, CAIQ, custom* • *SOC 2 review process* • *contract security clauses* • *ongoing monitoring* • *vendor offboarding)* • *security awareness programmes (phishing simulation, role-based training, security champions)* • *KPIs for security programmes (MTTD, MTTR, control coverage, audit findings closed, risk register age).*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

Reference Books:

Sno	Title	Author(s)
1	IT Governance: An International Guide to Data Security and ISO	Calder, A. & Watkins, S
2	The Cybersecurity Mindset, IT Governance	Hayslip, G.
3	Measuring and Managing Information Risk:	Freund, J. & Jones, J.
4	ISO/IEC 27001:2022 + ISO/IEC 27002:2022	------
5	NIST CSF 2.0 + NIST SP 800-30 + NIST SP 800-53r5 + NIST SP 800-171	-------
6	AICPA Trust Services Criteria for SOC 2	-------
7	RBI Cyber Security Framework	------

Stream D: Blockchain & Fintech

COURSE CODE: 26CA33D4E
COURSE NAME: REGTECH & COMPLIANCE AUTOMATION
L-T-P-S: 3-0-3-0
PRE-REQ: SCDS

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse the Indian regulatory landscape for financial services (PMLA 2002, RBI Master Directions on KYC/AML, FIU-IND reporting, SEBI obligations) to identify obligations applicable to a fintech, bank, or crypto VASP.	PO1, PO2	4

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO2	Design risk-based KYC and Customer Due Diligence (CDD) workflows including simplified, standard, and enhanced due diligence tiers based on customer risk classification.	PO1, PO2	5
CO3	Implement transaction monitoring systems combining rule-based scenarios (velocity, structuring, threshold) with ML-based anomaly detection, generating alerts with low false-positive rates.	PO2, PO3	5
CO4	Examine sanctions screening, PEP screening, and adverse-media screening pipelines including list management, fuzzy matching, and false-positive workflows	PO2, PO3	4
CO5	Architect Suspicious Transaction Report (STR) and Cash Transaction Report (CTR) generation systems compliant with FIU-IND XML schemas with proper audit trails.	PO4, PO5	5
CO6	Build a comprehensive RegTech platform integrating KYC, transaction monitoring, sanctions screening, regulatory reporting, and audit trail across the customer lifecycle.	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Indian RegTech Landscape** Mapped: **CO1, CO5** · 6 hrs • Why RegTech (compliance is the #2 cost in financial services after IT, growing 15% YoY) • Prevention of Money Laundering Act (PMLA) 2002 and PMLA Rules • FATF (Financial Action Task Force) recommendations and India's mutual evaluation • RBI Master Directions on KYC (4 May 2016, with subsequent updates) • RBI Master Directions on Information Technology Governance • SEBI obligations for intermediaries • IRDAI for insurance • PFRDA for pensions • FIU-IND (Financial Intelligence Unit India) and FINGate XML schemas (STR, CTR, NTR, CCR) • Crypto regulations (PMLA extension to VASPs in March 2023, the Travel Rule for crypto) • India CERT-In Directions overlay • sectoral DPDP rules when notified.

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundat
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225

M2	***Module 2 — Risk-Based KYC & Customer Due Diligence*** ***Mapped: CO2, CO5 · 6 hrs*** • *Customer risk classification (low/medium/high) and the risk-based approach (RBA) per FATF* • *simplified due diligence (SDD) — when permitted under RBI directions* • *standard customer due diligence (CDD)* • *enhanced due diligence (EDD) — when required (PEPs, high-risk jurisdictions, complex structures)* • *Ultimate Beneficial Owner (UBO) identification (the 25% rule and complications)* • *ongoing CDD and periodic re-KYC (annual for high-risk, every 8/10 years for others)* • *CKYC (Central KYC Records Registry) and the CKYC ID* • *V-CIP (Video Customer Identification Process) — RBI guidelines and operational reality* • *Aadhaar eKYC under Aadhaar Act* • *offline KYC (XML)* • *the consent architecture (DigiLocker, AA framework)* • *India Stack components for KYC (Aadhaar, eSign, DigiLocker).*
M3	***Module 3 — Transaction Monitoring*** ***Mapped: CO3, CO5 · 6 hrs*** • *The transaction monitoring problem (millions of transactions, very few suspicious, very high false positive cost)* • *rule-based scenarios — velocity (X transactions in Y time), structuring (split deposits to evade ₹10L threshold), threshold (transactions > limits), pattern (round-amounts, dormant-then-active, geographic anomaly, merchant category anomaly)* • *rule tuning (lowering thresholds increases recall, increases false positives, increases ops cost)* • *the alert lifecycle (open, in-review, escalated, closed-true-positive, closed-false-positive)* • *machine learning approaches (supervised models on historical confirmed cases, unsupervised anomaly detection, network analysis for mule detection)* • *model risk management for ML in compliance* • *explainability requirements* • *the Suspicious Activity Report (SAR / STR) process.*
M4	***Module 4 — Sanctions, PEP & Adverse Media Screening*** *Mapped:* ***CO4, CO5*** *· 6 hrs* • *Sanctions lists (UN Consolidated List, OFAC SDN, EU Consolidated, UK HMT, India's Ministry of External Affairs UAPA list)* • *list aggregators and update cadences* • *the screening problem (matching customer name against millions of records with name variations, transliteration, and aliases)*

8/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• *fuzzy matching algorithms (Levenshtein, Jaro-Winkler, Soundex, Metaphone, vector similarity for transliterated names)* • *blocking key strategies* • *false positives vs false negatives trade-offs* • *PEP (Politically Exposed Persons) databases (Dow Jones Risk & Compliance, Refinitiv World-Check, ComplyAdvantage, open-source: OpenSanctions)* • *RCA (Relatives and Close Associates)* • *adverse media / negative news screening using NLP* • *the screening pipeline integrated into onboarding and continuous monitoring* • *List of Designated Individuals/Entities (LDIE) under UAPA* • *crypto address screening (Chainalysis, TRM Labs, Elliptic for blockchain analytics).*
M5	***Module 5 — Reporting, Audit Trail & GRC Tooling*** ***Mapped: CO5, CO6 · 6 hrs*** • *FIU-IND FINGate portal and XML schemas (STR, CTR for ₹10L+ cash, NTR for non-profit, CCR for cross-border wire)* • *STR XML field-by-field* • *submission timelines (within 7 days of identification)* • *the audit trail requirement (every compliance decision must be reviewable years later) — immutable event log design* • *the regulator inspection scenario (auditors arrive, demand 2 years of decisions on a sample of customers — show me the evidence)* • *compliance case management (alerts, investigations, escalation, evidence linking, final disposition)* • *RegTech platforms (open-source: Compliance.ai concepts, Apache Ranger for data* • *commercial: ComplyAdvantage, Refinitiv, Actico, Featurespace, Trulioo)* • *GST integration via APIs (e-invoicing IRP, GSTN return data for AML signal)* • *regulatory horizon scanning* • *whistleblower mechanisms.*

Reference Books:

Sno	Title	Author(s)
1	RegTech: A Comprehensive Guide	Bali, S
2	Anti-Money Laundering Compliance: A Practical Guide	Lyons, R. & Schiltz, A

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

3	ACAMS Today articles + ACAMS study guides for CAMS certification	—
4	FATF Recommendations + India Mutual Evaluation Report	—
5	RBI Master Directions on KYC, AML, IT Governance	—
6	FIU-IND FINGate XML schema documentation	—
7	OpenSanctions documentation + datase	—

Stream E: Data Science

COURSE CODE: 26CA33E4E
COURSE NAME: EXPERIMENTATION & CAUSAL ANALYTICS
L-T-P-S: 3-0-3-0
PRE-REQ: DEAS

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Analyse the difference between interpretability (intrinsically transparent models) and post-hoc explainability to choose the appropriate approach for regulatory, debugging, and trust scenarios.	PO1, PO2	4
CO2	Design feature attribution methods (LIME, SHAP, Integrated Gradients, attention rollout, GradCAM) appropriate to model architecture (tree, linear, deep, transformer, vision) producing faithful and actionable explanations.	PO1, PO2	5
CO3	Examine algorithmic fairness frameworks including demographic parity, equal opportunity, equalised odds, and individual fairness to evaluate model decisions across protected groups.	PO2, PO3	4
CO4	Implement bias detection and mitigation pipelines using IBM AI Fairness 360, Microsoft Fairlearn, and Google What-If Tool across pre-processing, in-processing, and post-processing approaches.	PO2, PO3	5

HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO5	Architect causal analytics workflows distinguishing predictive vs causal questions, identifying confounders, and applying observational causal methods to make actionable business or policy recommendations.	PO4, PO5	5
CO6	Build accountability artefacts including model cards, datasheets, fairness audits, and explainability reports compliant with emerging AI regulation (EU AI Act, NITI Aayog Responsible AI principles, Indian DPDP-aligned algorithmic transparency).	PO4, PO5	5

Syllabus:

Module No.	Syllabus
M1	**Module 1 — Why Explainability & The Interpretability Spectrum** Mapped: **CO1, CO6** · 6 hrs • Why explainability matters now (high-stakes decisions: credit, hiring, healthcare, criminal justice • regulatory pressure — EU AI Act risk tiers, US algorithmic accountability proposals, India NITI Aayog Responsible AI • debugging benefits independent of regulation • user trust) • the interpretability vs accuracy myth (modern XGBoost + SHAP often delivers both) • the interpretability spectrum — intrinsically interpretable models (linear regression, logistic regression, decision trees, GAMs/EBM, falling rule lists, RuleFit) vs post-hoc explainability on opaque models • the right model for each problem • the cost of explainability (some methods are computationally expensive at inference time) • when explainability isn't enough (the 'recourse' problem — telling someone why they were denied is different from giving them an actionable path).
M2	***Module 2 — Feature Attribution Methods*** ***Mapped: CO2, CO5 · 6 hrs*** • *Local vs global explanations* • *permutation importance and its drawbacks (correlated features)* • *LIME (local interpretable model-agnostic explanations) — algorithm, kernel, sampling instability* • *SHAP (Shapley values from cooperative game theory) — TreeSHAP for trees, KernelSHAP for any model, DeepSHAP for neural networks*

31/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicati
Koneru Lakshmaiah Education Founda
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

	• *SHAP summary plots, dependence plots, force plots, beeswarm* • *Integrated Gradients for deep models (axiomatic justification)* • *Grad-CAM and Grad-CAM++ for CNNs* • *attention visualisation in transformers (and the case that attention is NOT explanation)* • *attention rollout* • *the faithfulness problem (do explanations actually reflect the model?)* • *explainability evaluation metrics (faithfulness, stability, robustness).*
M3	***Module 3 — Algorithmic Fairness*** ***Mapped: CO3, CO4 · 6 hrs*** • *Why algorithmic bias is hard (datasets reflect history* • *the world is unfair, models learn that)* • *types of harm — allocational vs representational* • *fairness definitions (demographic parity, equal opportunity, equalised odds, predictive parity, counterfactual fairness, individual fairness)* • *the fairness impossibility theorem (Chouldechova, Kleinberg — multiple fairness criteria cannot generally hold simultaneously)* • *audit case studies (COMPAS, Apple Card credit decisions, Amazon hiring tool, Tay, Indian caste/religion considerations)* • *pre-processing methods (reweighing, sampling, learned representations)* • *in-processing methods (adversarial debiasing, fairness constraints in optimisation)* • *post-processing methods (Reject Option, threshold optimisation per group)* • *the legal landscape — disparate treatment vs disparate impact (US 4/5 rule)* • *EU AI Act protected categories* • *India's emerging frameworks.*
M4	***Module 4 — Counterfactual & Causal Analytics*** *Mapped:* ***CO5, CO2*** *· 6 hrs* • *Counterfactual explanations (DiCE — what minimal change would flip the decision)* • *actionability constraints (immutable features like age cannot be changed)* • *plausibility constraints (counterfactuals on the data manifold)* • *recourse — the practical question 'what should I do to get approved'* • *counterfactual fairness as a fairness criterion* • *causal fairness* • *the difference between predictive and causal questions in business analytics (will this user churn vs will this intervention prevent churn)* • *confounding in business data (selection bias, survivor bias, simultaneity)* • *when prediction is enough vs when causal is required* • *causal discovery basics (PC algorithm, FCI, NOTEARS for DAG learning from observational data) overview*

HoD-CSA 21/7/26

Dr. K [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• *introducing causal thinking to a data science team.*
M5	***Module 5 — Accountability Artefacts & Production XAI*** ***Mapped: CO6, CO1 · 6 hrs*** • *Model cards (Mitchell et al., Google) — what to put in them* • *datasheets for datasets (Gebru et al.)* • *FactSheets (IBM)* • *the EU AI Act risk tiers (unacceptable / high / limited / minimal) and conformity requirements* • *the EU AI Act's annex with high-risk systems* • *NITI Aayog Responsible AI principles (Indian)* • *fairness audits (algorithmic auditing protocols, second-party vs third-party)* • *explainability in regulated domains (healthcare FDA Predetermined Change Control Plans, RBI fair-lending circulars, SEBI algorithmic trading disclosures)* • *productionising XAI (explanation latency budgets, caching, pre-computed explanations, on-demand SHAP serving)* • *explanation drift over time* • *user interfaces for explanations (when to show, how to phrase, the recourse vs reason distinction)* • *LLM explainability frontier (mechanistic interpretability — Anthropic's circuits, sparse autoencoders, OpenAI superposition research) — current state.*

Reference Books:

Sno	Title	Author(s)
1	Interpretable Machine Learning	Molnar, C.
2	Explainable Artificial Intelligence	Barredo Arrieta, A. et al.
3	Model Cards for Model Reporting	Mitchell, M. et al.

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

4	Datasheets for Datasets	Gebru, T. et al
5	Fairness and Machine Learning	Barocas, S., Hardt, M., Narayanan, A
6	The Book of Why, Basic Books	Pearl, J. & Mackenzie, D.
7	Microsoft Fairlearn documentation	—

COURSE CODE: 26IE3030E
COURSE NAME: CAPSTONE PROJECT
L-T-P-S: 0-0-18-0 **PRE-REQ: TRPO**

CO. No	Course Outcome Description	PO MAPPING	BTL
CO1	Apply requirements engineering, system design, and architecture decision-making to scope a 12-week buildable product aligned with the chosen specialization stream.	PO1, PO2	3
CO2	Apply full-stack engineering practices — version control hygiene, branching strategy, code review, automated testing, CI/CD — across the build phase.	PO1, PO2	3
CO3	Analyze cloud deployment options, scaling characteristics, and cost trade-offs to deploy the product on a managed cloud platform with infrastructure-as-code.	PO2, PO3	4
CO4	Analyze security posture — OWASP Top 10, dependency vulnerabilities, secrets management, authentication/authorization — and apply remediations before launch.	PO2, PO3	4
CO5	Evaluate observability, performance, and reliability through structured logging, metrics, traces, SLOs, and load testing.	PO4, PO5	5
CO6	Evaluate the product through user testing, demo presentation, technical defense, and a published artifact (live URL + open-source repo + technical write-up).	PO4, PO5	5

31/7/26
HoD-CSA

Dr. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Syllabus:

Module No.	Syllabus
M1	**Phase 1 — Proposal, Scoping & Mentor Pairing** Mapped: **CO1 (BTL-3)** · 12 hrs · Gate: Approved proposal • Stream alignment: capstone must be in the declared specialization stream (S1-S5) • Problem statement: who is the user, what pain, what success looks like • Scope guardrails: must be deployable in 8 weeks of build, with 2 weeks of polish • Solo or 2-person team — no larger; if 2-person, a clear work-split must be presented • Capstone differs from term paper: this is engineering output, not research output • Capstone differs from course projects: must integrate ≥ 3 stream-relevant subsystems • Mentor pairing (each faculty supervises 4-5 capstones, or 2-3 if 2-person teams) • Proposal document: problem, users, scope, architecture sketch, build plan, eval plan *Approved 5-page proposal by end of Week 2 with: problem statement, target user, scope guardrails, draft architecture diagram, week-by-week build plan, evaluation criteria. Signed off by mentor + second faculty reviewer.*
M2	**Phase 2 — Architecture, Design & Repo Setup** Mapped: **CO1, CO2 (BTL-3)** · 12 hrs · Gate: Architecture approved • Architecture decision records (ADRs): capture every significant choice with rationale • Tech stack lock: language, framework, database, hosting, observability — all finalized • Database schema: ERD or document model; migration tooling chosen • API surface: REST or GraphQL or gRPC; documented (OpenAPI / SDL) • Authentication strategy: JWT, OAuth, magic links, etc. • Repo setup: GitHub repo with README, LICENSE, branch protection, CI scaffolding • Project management: GitHub Projects board with all weekly milestones loaded • Threat model (lightweight STRIDE walk-through) *Approved architecture document by end of Week 4: ADRs, ER diagram, API spec (or GraphQL schema), CI scaffold green, threat model recorded.*

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M3	**Phase 3 — Core Build (Sprint 1)** Mapped: **CO2 (BTL-3)** · 36 hrs · Gate: Working core flow • Build the happy-path end-to-end vertical slice first — frontend, API, database • Auth integrated from day 1 (do not bolt on later) • Branching strategy: feature branches → PR → review → merge to main • Code review even on solo capstones: faculty mentor reviews ≥ 2 PRs/week • Automated tests: unit (≥ 60% coverage on business logic) + ≥ 5 integration tests • CI: every push runs lint + tests; PR requires green CI to merge • Deploy preview environments per branch (Vercel / Netlify / Render free tier) • Daily commits expected; weekly demo of progress to mentor *Working core flow demoed by end of Week 7: user can sign in, complete the primary user journey end-to-end. Test coverage and CI green.*
M4	**Phase 4 — Cloud Deploy & Infrastructure-as-Code** Mapped: **CO3 (BTL-4)** · 24 hrs · Gate: Production deployment • Choose cloud target — managed PaaS (Render, Railway, Fly.io) for simpler stacks; AWS/GCP for more involved ones • Containerize with Docker; multi-stage builds; minimal images • IaC: Terraform or Pulumi for infrastructure (mandatory — clicking in console is not acceptable) • Database: managed Postgres (Neon, Supabase, RDS); migrations via Flyway/Prisma/Alembic • Secrets: cloud secret manager (AWS Secrets Manager, GCP Secret Manager) — never in repo • Domain: free subdomain (or custom if budget permits) with TLS • CI/CD: deploy on merge to main; rollback strategy documented • Cost dashboard: weekly cost tracking; budget cap alerts *Live production URL by end of Week 9. IaC repo committed; deploy works on merge to main; cost tracking + budget alerts configured. Free tier preferred (target: < ₹500 total cloud spend).*
M5	**Phase 5 — Security Hardening & Observability** Mapped: **CO4, CO5 (BTL-4/5)** · 18 hrs · Gate: Security & observability sign-off • OWASP Top 10 self-audit: checklist with evidence (code reference / mitigation) for each item • Dependency scan: pip-audit / npm audit / Snyk free tier — zero criticals before launch

HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	• SAST: Semgrep CI on every PR (uses CYBF / DSOP knowledge) • Secrets scan: gitleaks pre-commit + CI (zero leaks in repo history) • Authentication hardening: rate limiting, password requirements, account lockout • Authorization tests: explicit IDOR test cases in the test suite • Structured logging (JSON logs); request correlation IDs • Metrics: Prometheus-format /metrics or Sentry/DataDog free tier • Tracing: OpenTelemetry instrumentation for at least the critical user journey • SLO definition: 1-3 user-facing SLOs (e.g., "99% of page loads < 1.5s") • Load test: k6 or Locust hitting prod-like staging at 10x expected load *Security audit report + observability stack live by end of Week 11: OWASP checklist filled, zero critical CVEs, structured logging in production, metrics dashboard, 1-3 SLOs measured, load test report*
M6	**Phase 6 — User Testing, Final Demo & Defense** Mapped: **CO6 (BTL-5)** · 18 hrs · Gate: Capstone accepted • User testing: ≥ 5 real users (peers, faculty, friends/family) complete tasks; record observations • Address top 3 UX issues found; iterate • Technical write-up: 5-8 page document covering problem, design, build, eval, lessons learned • README quality: setup, architecture diagram, screenshots/GIFs, demo video link • Demo video: 3-minute recorded walkthrough showing the core flow • Live demo session: 15-minute presentation to a 3-faculty panel + open Q&A • Technical defense: panel asks deep architecture questions; student defends decisions • Optional: open-source release with proper licensing; HN/Reddit launch (extra credit) *Final accepted capstone by Week 12: live URL, public repo, technical write-up, demo video, user testing report, defense passed. Project committed to institutional showcase site.*

Reference Books:

Sno	Title	Author(S)	Publisher	Edition / Year	ISBN
1	*The Pragmatic Programmer*	Andrew Hunt,	Addison-	20th Anniversary	978-

3/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

.		David Thomas	Wesley	Ed. 2019	0135957059
2	*Designing Data-Intensive Applications*	Martin Kleppmann	O'Reilly	1st Ed, 2017	978-1449373320
3	*The Phoenix Project*	Gene Kim, Kevin Behr, George Spafford	IT Revolution Press	5th Anniversary Ed, 2018	978-1942788294
4	*Site Reliability Engineering* (free online)	Beyer et al. (Google)	O'Reilly	1st Ed, 2016	978-1491929124
5	*The Web Application Hacker's Handbook*	Stuttard, Pinto	Wiley	2nd Ed, 2011	978-1118026472
6	*Refactoring: Improving the Design of Existing Code*	Martin Fowler	Addison-Wesley	2nd Ed, 2018	978-0134757599
7	*Software Engineering at Google* (free online)	Winters, Manshreck, Wright	O'Reilly	1st Ed, 2020	978-1492082798

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

ANNEXURE-II

Y26 MCA PROGRAM STRUCTURE

Detailed Structure of the Program highlighting all the courses and their credits

Course Code	COURSE TITLE	CAT	L	T	P	S	Cr	CH	PRE-REQ	New/ Revised/Retained	Stakeholder feedback based on which change was proposed	Focused on Employability/Entrepreneurship/ Skill Development or Career Advancement	Justification (Detailed Justification on how the course content maps to employability/entrepreneurship/skill category.)
26CA5101E	Python Programming - 1	PCC	3	0	6	0	4	9	NIL	Newly Introduced Course	Faculty	Employability	Equips students with fundamental programming concepts, fosters problem-solving skills.
26CA5102E	Linux & OS Administration	PCC	3	0	3	0	3	6	NIL	Newly Introduced Course	Faculty	Employability	Enhance knowledge on Internal working of devices and administration of Linux
26CA5103E	Database & Distributed DB Admin	PCC	3	0	3	0	3	6	NIL	Newly Introduced Course	Faculty	Employability	Enhance the knowledge on SQL Queries for employability
26CA5104E	Cloud Admin & Cloud DevOps	PCC	3	0	3	0	3	6	NIL	Newly Introduced Course	Industry peers	Employability	Provides students with practical skills and knowledge in the realm of DevOps

Dr. K. BHAVANA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

										uced Course			that can plan tasks, write code for both frontends and backends, and securely connect to external data sources or tools in real-time
26CA6216E	Blockchain & Smart Contract Prog	PCC	3	0	3	0	3	6	FSD2	Newly Introduced Course	Faculty	Skill development	Enhances the students to build secure, self-executing digital programs that run on decentralized networks without middlemen.
26CA6217E	Defensive Security	PCC	3	0	3	0	3	6	OFSC	Newly Introduced Course	Faculty	Employability	Equip students to build secure infrastructure, block unauthorized access, detect live attacks in real-time, and minimize damage when security breaches occur.
26IE6205E	Capstone Project - 1	PRI	0	0	12	6	5	18	TINT	Newly Introduced Course	Faculty	Skill development	Able to build the projects by analysing the problems in existing system.
26CA6318E	Web3 & dApp Development	PCC	3	0	3	0	3	6	BCSC	Newly Introduced Course	Faculty	Employability	Equips students to build software on blockchains on Ethereum
26CA6319E	DevSecOps	PCC	3	0	3	0	3	6	DFSC	Newly Introduced Course	Academic Peers	Employability	Enhances the student in Development, Security, and Operations and to build security into every step of making software
26CA6320E	Data Science for Scale	PCC	3	0	3	0	3	6	DSAV	Newly Introduced Course	Faculty	Employability	Equips the student how to build, optimize, and run data models and

Dr. K. BHAGAWAN
HoD-CSA
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM.
Guntur District, Andhra Pradesh-522502.

													systems that handle massive amounts of information efficiently
26IE6306E	Capstone Project - 2	PRI	0	0	12	6	5	18	CPS1	Newly Introduced Course	Faculty	Skill development	Able to build the projects by analysing the problems in existing system.
26CA61A1E	Conversational AI & Speech Systems	PE-1	3	0	3	0	3	6	NIL	Newly Introduced Course	Faculty	Employability	Equips the students to build computer programs that can listen to, understand, and talk with humans using natural voice or text
26CA61B1E	Microservices, Service Mesh & Distributed Systems	PE-1	3	0	3	0	3	6	NIL	Newly Introduced Course	Industry peers	Employability	Equip students to build large teams build fast apps that do not break all at once if one small part stops working
26CA61C1E	Advanced Threat Hunting & Adversary Emulation	PE-1	3	0	3	0	3	6	NIL	Newly Introduced Course	Faculty	Entrepreneurship	Equips the students to uncover hidden breaches and test whether defensive tools, logs, and response teams can stop real-world, multi-stage attacks
26CA61D1E	DeFi Protocols & Tokenomics Engineering	PE-1	3	0	3	0	3	6	NIL	Newly Introduced Course	Faculty	Employability	Enhance students capability to build open, self-running financial systems on blockchains

HoD-CSA 31/7/26
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

26CA61E1E	Analytics Engineering & Modern Data Platforms	PE-1	3	0	3	0	3	6	NIL	Newly Introduced Course	Industry peers	Employability	Equips to build fast data pipelines, use cloud warehouses, and make reliable reports for business growth.
26CA61A2E	Computer Vision & Multimodal AI	PE-2	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	Equips to build machines to see, hear, read, and reason across different types of data at the same time
26CA61B2E	Event-Driven & Stream Processing Architectures	PE-2	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	helps to build systems that remove delays, track live changes, and process endless data feeds on the fly
26CA61C2E	Privacy Engineering & Data Protection (DPDP, GDPR)	PE-2	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	helps to understand laws like India's Digital Personal Data Protection Act and Europe's GDPR
26CA61D2E	Fintech Apps — Payments, Trading, Lending	PE-2	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Entrepreneurship	Equips to build apps for payments, trading, and lending make money management fast and easy
26CA61E2E	Causal Analytics & Experimentation	PE-2	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	Equips to build apps that avoid false conclusions from random patterns, and make smart choices based on real cause-and-effect

Dr. K. [illegible] 31/7/26
HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

26CA62A3E	Recommender Systems & Personalization at Scale	PE-3	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	Enhances the ability to design algorithms and massive data pipelines that predict user preference
26CA62B3E	SRE & Advanced Observability	PE-3	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	helps to track metrics, logs, and traces to find system problems fast that reduces downtime, cuts manual work, and keeps applications running smoothly for users.
26CA62C3E	Cloud Native Security & Zero Trust Architecture	PE-3	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	Equips to protect modern apps built on microservices and containers
26CA62D3E	RegTech & Compliance Automation	PE-3	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Skill development	to use advanced tools like artificial intelligence and big data to help businesses manage complex laws and rules efficiently and automatically
26CA62E3E	Business Forecasting & Demand Analytics	PE-3	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	to Enhance and optimize inventory, plan production, manage supply chains, and allocate budgets effectively to minimize waste and maximize profitability.

HoD-CSA

31/7/26

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

26CA63A 4E	Intelligent Document Processing & RPA	PE-4	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Entrepreneurship	Equips to uses artificial intelligence to understand document content, while RPA acts as the digital worker that moves that data into other software systems
26CA63B 4E	Serverless, Edge & FaaS Computing	PE-4	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	to run code without managing servers to cut infrastructure costs, scale instantly based on demand, and lower latency by running tasks closer to users
26CA63C 4E	Security Governance, Risk & Compliance (GRC)	PE-4	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	Equips to align an organization's IT and security actions with business goals.
26CA63D 4E	Enterprise Distributed Ledger & Layer-2 Solutions	PE-4	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	Equips to speed up transactions and lower costs while keeping data safe and private across multiple companies.
26CA63E 4E	Customer Growth & Behavioural Analytics	PE-4	3	0	3	0	3	6	PEC1	Newly Introduced Course	Faculty	Employability	to turn user action data into insights that increase sales, stop customers from leaving, and improve the user experience.

Percentage of Syllabus Revision = **100% (48 out of 48)**
Percentage of Courses focusing on Employability= **70.84% (34 out of 48)**
Percentage of Courses focusing on Entrepreneurship= **8.33% (4 out of 48)**
Percentage of Courses focusing on Skill Development = **20.83% (10 out of 48)**

Dr. K. BHAGAVAN
HoD-CSA 31/7/26
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Master of Computer Applications

Y26 Batch

(DEEMED TO BE UNIVERSITY)

A.Y. 2026-27

Syllabus

HoD-CSA

Dr. K BHAGAVAN 31/1/26
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Y26 MCA - Trimester (PBL) Structure

Sl No	TRI	Course Code	CAT	COURSE TITLE	ACRO	L	T	P	S	Cr	CH	PRE-REQ
1	T01	26CA5101E	PCC	Python Programming - 1	PYP1	3	0	6	0	4	9	NIL
2	T01	26CA5102E	PCC	Linux & OS Administration	LXOS	3	0	3	0	3	6	NIL
3	T01	26CA5103E	PCC	Database & Distributed DB Admin	DBDA	3	0	3	0	3	6	NIL
4	T01	26CA5104E	PCC	Cloud Admin & Cloud DevOps	CDVO	3	0	3	0	3	6	NIL
5	T01	26UC5103E	HAS	Professional Communication & Mathematical Methods	PCMM	0	0	3	0	0	3	NIL
6	T02	26CA5205E	PCC	Python Programming - 2	PYP2	3	0	3	0	3	6	PYP1
7	T02	26CA5206E	PCC	Full Stack Development - 1	FSD1	3	0	6	0	4	9	PYP1
8	T02	26CA5207E	PCC	AI & Machine Learning	AIML	3	0	6	0	4	9	PYP1
9	T02	26CA5208E	PCC	Offensive Security	OFSC	3	0	3	0	3	6	LXOS
10	T03	26CA5309E	PCC	Full Stack Development - 2	FSD2	3	0	6	0	4	9	FSD1
11	T03	26CA5310E	PCC	Applied DL, LLM Eng & RAG	ADRG	3	0	0	6	3	9	AIML
12	T03	26CA5311E	PCC	HPC Admin & Cluster Mgmt	HPCM	3	0	3	0	3	6	LXOS
13	T03	26CA5312E	PCC	Adaptive SE with MLOps	ASEM	3	0	3	0	3	6	AIML
14	1S	26IE5004E	PRI	Technical Internship	TINT	0	0	12	0	4	12	NIL

Dr. K. BHARGAVAN
HoD-CSA 3/4/26
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

15	T04	26CA6113E	PCC	GenAI API & AI Platform Eng	GAIP	3	0	0	6	3	9	ADRG
16	T04	26CA6114E	PCC	Data Science, Analytics & Viz	DSAV	3	0	3	0	3	6	AIML
17	T04	PE1	PEC	Professional Elective - 1	PEC1	3	0	3	0	3	6	NIL
18	T04	PE2	PEC	Professional Elective - 2	PEC2	3	0	3	0	3	6	PEC1
19	T05	26CA6215E	PCC	Agentic AI + Full Stack + MCP	AAFM	3	0	3	0	3	6	GAIP
20	T05	26CA6216E	PCC	Blockchain & Smart Contract Prog	BCSC	3	0	3	0	3	6	FSD2
21	T05	26CA6217E	PCC	Defensive Security	DFSC	3	0	3	0	3	6	OFSC
22	T05	PE3	PEC	Professional Elective - 3	PEC3	3	0	3	0	3	6	PEC1
23	T05	26IE6205E	PRI	Capstone Project – 1	CPS1	0	0	12	6	5	18	TINT
24	T06	26CA6318E	PCC	Web3 & dApp Development	W3DA	3	0	3	0	3	6	BCSC
25	T06	26CA6319E	PCC	DevSecOps	DVSO	3	0	3	0	3	6	DFSC
26	T06	26CA6320E	PCC	Data Science for Scale	DSSC	3	0	3	0	3	6	DSAV
27	T06	PE4	PEC	Professional Elective - 4	PEC4	3	0	3	0	3	6	PEC1
28	T06	26IE6306E	PRI	Capstone Project – 2	CPS2	0	0	12	6	5	18	CPS1

	Total Credits		**90**	

List of Professional Electives

Slno	TRI	COURSE CODE	CAT	COURSE TITLE	ACRO	L	T	P	S	Cr	CH	PRE-PREQ
1	TO4	26CA61A1E	PE-1	Conversational AI & Speech Systems	CASS	3	0	3	0	3	6	AIML
2	TO4	26CA61B1E	PE-1	Microservices, Service Mesh & Distributed Systems	CDVO	3	0	3	0	3	6	CDVO

31/7/26

HoD-CSE

Dr. K. [illegible]AVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

3	TO 4	26CA61C1E	PE-1	Advanced Threat Hunting & Adversary Emulation	OFSC	3	0	3	0	3	6	OFSC
4	TO 4	26CA61D1E	PE-1	DeFi Protocols & Tokenomics Engineering	PYPS	3	0	3	0	3	6	PYPS
5	T04	26CA61E1E	PE-1	Analytics Engineering & Modern Data Platforms	PYPS	3	0	3	0	3	6	PYPS
6	T04	26CA61A2E	PE-2	Computer Vision & Multimodal AI	ADRG	3	0	3	0	3	6	ADRG
7	T04	26CA61B2E	PE-2	Event-Driven & Stream Processing Architectures	CDVO	3	0	3	0	3	6	CDVO
8	T04	26CA61C2E	PE-2	Privacy Engineering & Data Protection (DPDP, GDPR)	DBDA	3	0	3	0	3	6	DBDA
9	T04	26CA61D2E	PE-2	Fintech Apps — Payments, Trading, Lending	FSPD	3	0	3	0	3	6	FSPD
10	T04	26CA61E2E	PE-2	Causal Analytics & Experimentation	AIML	3	0	3	0	3	6	AIML
11	T05	26CA62A3E	PE-3	Recommender Systems & Personalization at Scale	AIML	3	0	3	0	3	6	AIML
12	T05	26CA62B3E	PE-3	SRE & Advanced Observability	CDVO	3	0	3	0	3	6	CDVO
13	T05	26CA62C3E	PE-3	Cloud Native Security & Zero Trust Architecture	CDVO	3	0	3	0	3	6	CDVO
14	T05	26CA62D3E	PE-3	RegTech & Compliance Automation	DBDA	3	0	3	0	3	6	DBDA
15	T05	26CA62E3E	PE-3	Business Forecasting & Demand Analytics	AIML	3	0	3	0	3	6	AIML
16	T06	26CA63A4E	PE-4	Intelligent Document Processing & RPA	ADRG	3	0	3	0	3	6	ADRG

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

HoD-CSA 31/7/26

17	T06	26CA63B4E	PE-4	Serverless, Edge & FaaS Computing	CDVO	3	0	3	0	3	6	CDVO
18	T06	26CA63C4E	PE-4	Security Governance, Risk & Compliance (GRC)	OFSC	3	0	3	0	3	6	OFSC
19	T06	26CA63D4E	PE-4	Enterprise Distributed Ledger & Layer-2 Solutions	BCSC	3	0	3	0	3	6	BCSC
20	T06	26CA63E4E	PE-4	Customer Growth & Behavioral Analytics	DSAV	3	0	3	0	3	6	DSAV

SYLLABUS

PYTHON PROGRAMMING I — FOUNDATIONS & OOP

COURSE CODE: 26CA5101E
COURSE NAME : Python Programming - 1
L-T-P-S :3-0-6-0
PRE-REQ : NIL

CO. No	Course Outcome Description	PO	BTL
CO1	Apply Python language fundamentals — types, control flow, functions, and standard data structures (list / tuple / dict / set) — to solve algorithmic problems and build small command-line utilities.	PO1,PO4	3
CO2	Apply object-oriented design principles — classes, encapsulation, inheritance, polymorphism, dunder methods, and dataclasses — to model domain entities for application-domain problems.	PO1,PO3	3
CO3	Analyze module structure, packaging, virtual environments (venv / poetry), and dependency management to design maintainable, distributable Python applications.	PO2,PO3	4
CO4	Analyze exception handling pathways, file I/O patterns (text / JSON / CSV / SQLite), and standard library usage to build robust scripts and reliable persistence layers.	PO2,PO4	4
CO5	Evaluate code quality and design choices to refactor solutions using idiomatic Python — comprehensions, generators, decorators, context managers — for readability, performance, and Pythonic style.	PO2,PO6	5

31/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO6	Evaluate the integration of testing (pytest), version control (Git / GitHub), and AI-assisted coding (GitHub Copilot, Claude Code) into a complete project deliverable, including CI workflows on GitHub Actions.	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Python Fundamentals & Algorithmic Thinking** Python interpreter, REPL, scripts vs notebooks. Numeric / string / boolean types, operators, casting. Control flow: if, for, while, match. Functions, parameters (positional / keyword / *args / **kwargs), return values, scope, closures. Data structures: list, tuple, dict, set; mutability vs immutability; common operations and complexity. Algorithmic problem-solving with these primitives.
M2	**Object-Oriented Programming** Classes and instances. __init__, instance vs class attributes, methods, self. Encapsulation; properties and descriptors. Inheritance, MRO, super(); multiple inheritance pitfalls. Polymorphism via duck typing and abstract base classes (abc). Magic methods (__str__, __repr__, __eq__, __hash__, __lt__, __add__). @dataclass, frozen dataclasses, slots. Composition over inheritance.
M3	**Modules, Packages & Project Structure** Module system, import mechanics, __name__ == "__main__". Package layout — flat vs src/; __init__.py; namespace packages. Virtual environments (venv); pip; requirements.txt vs pyproject.toml; introduction to poetry / uv. Distribution: building wheels, publishing to TestPyPI. Project layout best practices.
M4	**Exceptions, File I/O & Standard Library** Exception hierarchy, try/except/else/finally, custom exceptions, exception chaining. File I/O — text vs binary, encoding, context managers. Structured data: json, csv, sqlite3. Standard library tour — os, pathlib, datetime, collections, itertools, functools, logging, argparse. Robust CLI applications.
M5	**Idiomatic Python — Pythonic Style** List / dict / set comprehensions and generator expressions. Generators and the yield statement; lazy evaluation. Iterators and the iterator protocol. Decorators — function and class decorators; functools.wraps. Context managers (with); writing custom managers via __enter__/__exit__ or contextlib. Type hints and mypy for static checking. Code quality: black, ruff, pylint.
M6	**Testing, Git & AI-Assisted Coding** Test-driven development with pytest — fixtures, parameterization, coverage. Mocking with unittest.mock. Git fundamentals — branches, merges, rebases; GitHub workflow with PRs and reviews. CI on GitHub Actions — lint, test, coverage gates. AI pair programming — GitHub Copilot patterns; Claude Code agentic mode; prompting for refactor / debug / docs. Project polish: README, license, badges.

Reference Books:

Dr. K. BHAGAVAN
HoD-CSA
HOD-Computer Science & Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

1. Learning Python (5th ed),Mark Lutz · O'Reilly · 2013, ISBN: 978-1-4493-5573-9 · Coverage: M1–M2 foundations
2. Fluent Python (2nd ed) — Clear, Concise, and Effective Programming,Luciano Ramalho · O'Reilly · 2022,ISBN: 978-1-4920-5635-5 · Coverage: M2 OOP + M5 idioms — gold-standard for Pythonic style
3. Python Cookbook (3rd ed),David Beazley & Brian K. Jones · O'Reilly · 2013,ISBN: 978-1-4493-4037-7 · Coverage: Practical recipes throughout M3–M5
4. The Python Standard Library by Example,Doug Hellmann · Addison-Wesley · 2017,ISBN: 978-0-13-429105-5 · Coverage: M4 stdlib tour
5. Effective Python (2nd ed) — 90 Specific Ways to Write Better Python,Brett Slatkin · Addison-Wesley · 2019,ISBN: 978-0-13-485398-7 · Coverage: M5 idioms · companion to Fluent Python
6. Test-Driven Development with Python (2nd ed),Harry Percival · O'Reilly · 2017,ISBN: 978-1-4919-5870-4 · Coverage: M6 testing & CI
7. Robust Python — Write Clean and Maintainable Code,Patrick Viafore · O'Reilly · 2021,ISBN: 978-1-4920-9051-9 · Coverage: M5–M6 type hints, mypy, code quality

LINUX & OPERATING SYSTEM ADMINISTRATION

COURSE CODE: 26CA5102E
COURSE NAME : Linux & OS Administration
L-T-P-S :3-0-3-0
PRE-REQ : NIL

CO. No	Course Outcome Description	PO	BTL
CO1	Apply Linux command-line tools, file system semantics, and process management to perform day-to-day system administration tasks confidently.	PO1,PO4	3
CO2	Apply shell scripting (Bash) with control flow, functions, and standard utilities (awk, sed, grep, find, xargs) to automate operational tasks.	PO1,PO4	3
CO3	Analyze user, group, and permission models (DAC + ACLs + capabilities + SELinux) to design least-privilege access for multi-tenant systems.	PO2,PO3	4
CO4	Analyze service management with systemd, networking with ip/ss/firewalld, and storage with LVM/RAID to diagnose runtime failures.	PO1,PO2	4
CO5	Evaluate hardening choices (kernel parameters, SSH policy, file integrity monitoring, audit logging) to design a CIS-benchmark-aligned baseline.	PO2,PO6	5
CO6	Evaluate fleet automation (Ansible) and observability (logs, metrics, traces) to design and ship a managed Linux infrastructure deliverable.	PO2,PO4,PO5	5

Syllabus:

HoD-CSA 31/1/26
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Module No.	Syllabus
M1	**Linux Foundations & CLI Mastery** Distributions and the Linux ecosystem; kernel vs userspace. File system hierarchy (FHS); inodes, links, mount points. Essential commands — file ops, text processing (cat, grep, sed, awk, cut, sort, uniq, tr), I/O redirection, pipes. Process management — ps, top/htop, kill signals, foreground/background, job control. Package management — apt, dnf, snap.
M2	**Bash Scripting & Automation** Shell variables, parameter expansion, command substitution. Control flow — if/case/for/while. Functions, scope, return values. Robust scripts — set -euo pipefail; signal traps; here-docs. Text-processing pipelines with awk and sed for log analysis. Cron and systemd timers for scheduled jobs. Error handling, logging conventions, exit codes.
M3	**Users, Permissions & Security Model** User and group management — useradd, usermod, /etc/passwd, /etc/shadow. Discretionary Access Control — chmod, chown, umask; setuid, setgid, sticky bit. POSIX ACLs (getfacl/setfacl). Linux capabilities. SELinux contexts and modes (enforcing, permissive). sudo policy via /etc/sudoers and sudoers.d. PAM modules and authentication chains.
M4	**Services, Networking & Storage** systemd — units, targets, dependencies, journalctl. Networking — ip addr/route/link, ss, nmcli, network namespaces, firewalld/nftables, DNS resolution. Storage — partitioning (parted, fdisk), filesystems (ext4, xfs, btrfs), LVM (PV/VG/LV, snapshots), software RAID basics. Network filesystems (NFS, SMB).
M5	**Hardening & Compliance** Kernel hardening — sysctl tuning, namespaces. SSH hardening — key-only auth, ciphers, rate limiting, fail2ban. File integrity monitoring (AIDE). Audit framework (auditd, ausearch, aureport). Log management — rsyslog, journald persistence, log rotation. CIS Benchmark walk-through; running an OpenSCAP scan and remediating findings.
M6	**Fleet Automation & Observability** Configuration management with Ansible — inventories, plays, roles, idempotence. Vault for secrets. Observability — metrics with Prometheus + node_exporter; logs with Loki; dashboards with Grafana. Alerting basics. Containers as a deployment unit — Podman/Docker fundamentals. Project polish: README, runbook, architecture diagram.

Reference Books:

1. UNIX and Linux System Administration Handbook (5th ed),Evi Nemeth, Garth Snyder, Trent Hein, Ben Whaley, Dan Mackin · Addison-Wesley · 2017,ISBN: 978-0-13-427755-4 · Coverage: All modules — gold-standard SA reference
2. The Linux Command Line (2nd ed),William Shotts · No Starch Press · 2019,ISBN: 978-1-59327-952-3 · Coverage: M1–M2 foundations

Dr. K. [illegible] GAVAN
HOD-Computer [illegible] Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.
HoD-CSA 31/7/22

3. Pro Bash Programming (2nd ed),Chris F.A. Johnson, Jayant Varma · Apress · 2015,ISBN: 978-1-4842-0122-1 · Coverage: M2 scripting depth
4. SELinux System Administration (3rd ed),Sven Vermeulen · Packt · 2020,ISBN: 978-1-80056-700-3 · Coverage: M3
5. systemd for Administrators,Lennart Poettering (online series) + man pages,Coverage: M4 systemd internals
6. Ansible: Up and Running (3rd ed),Bas Meijer, Lorin Hochstein, René Moser · O'Reilly · 2022,ISBN: 978-1-09-810915-2 · Coverage: M6
7. Practical Monitoring,Mike Julian · O'Reilly · 2017,ISBN: 978-1-4919-5735-6 · Coverage: M6 observability

DATABASE & DISTRIBUTED DATABASE ADMINISTRATION

COURSE CODE: 26CA5103E
COURSE NAME : Database & Distributed DB Admin
L-T-P-S :3-0-3-0
PRE-REQ :NIL

CO. No	Course Outcome Description	PO	BTL
CO1	Apply relational design principles (normalization, ER modelling, SQL DDL/DML) to build well-structured PostgreSQL schemas for application domains.	PO1,PO3	3
CO2	Analyze SQL execution plans (EXPLAIN ANALYZE), index strategies (B-tree, hash, GiST, GIN), and statistics to optimize slow queries.	PO2,PO4	4
CO3	Apply NoSQL document design (MongoDB) — schema patterns, embedding vs referencing, indexing, aggregation pipelines — to model semi-structured data.	PO1,PO3	3
CO4	Analyze ACID semantics, isolation levels, MVCC, and locking to diagnose concurrency issues in transactional workloads.	PO1,PO2	4
CO5	Evaluate distribution strategies — replication, sharding, partitioning, CAP trade-offs — to design highly-available database architectures.	PO2,PO3	5
CO6	Evaluate operational practices — backup/restore, point-in-time recovery, monitoring, security — to ship a production-ready database deployment.	PO2,PO5,PO6	5

Syllabus:

HoD-CSA 31/7/26
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Module No.	Syllabus
M1	**Relational Modelling & SQL** ER modelling, normalization (1NF–BCNF, 4NF). PostgreSQL DDL: CREATE TABLE, constraints, foreign keys, CHECK. DML: SELECT, JOIN (inner/left/right/full), GROUP BY, HAVING, window functions, CTEs (recursive). Set operations. Subqueries — correlated vs non-correlated.
M2	**Query Optimization & Indexing** PostgreSQL planner internals; EXPLAIN ANALYZE / BUFFERS reading. Index types — B-tree, hash, GiST, GIN, BRIN. Covering indexes. Statistics, ANALYZE, autovacuum. Common anti-patterns — N+1 queries, missing indexes, bad cardinality estimates. pg_stat_statements for hot-query identification.
M3	**NoSQL & MongoDB** CAP theorem; document, wide-column, key-value, graph models. MongoDB CRUD; document schema patterns — embedded docs, references, polymorphic, bucket. Indexing in MongoDB — single, compound, multi-key, text, geo. Aggregation pipeline — $match, $group, $project, $lookup, $unwind, $facet. Schema validation; transactions in MongoDB
M4	**Transactions & Concurrency** ACID properties. Isolation levels — Read Uncommitted, Read Committed, Repeatable Read, Serializable. Anomalies — dirty read, non-repeatable read, phantom read. PostgreSQL MVCC; locks (row, table, advisory). Deadlock detection & resolution. Two-phase commit (2PC). Optimistic vs pessimistic concurrency.
M5	**Replication & Sharding** Replication patterns — sync vs async, single-leader, multi-leader, leaderless. PostgreSQL streaming replication; logical replication. MongoDB replica sets — primary election, oplog. Sharding — hash, range, geo. MongoDB sharded clusters. CAP and PACELC trade-offs. Failover strategies. Read replicas + connection pooling.
M6	**Operations: Backup, Monitoring, Security** Backup strategies — pg_basebackup, pg_dump, WAL archiving; mongodump, snapshot. Point-in-time recovery. Monitoring — pg_stat_*, mongostat, Prometheus exporters. Database security — RLS, GRANT/REVOKE, encryption at rest & in transit, audit logging. Secrets & connection strings management. Migrations with Flyway / Liquibase.

Reference Books:,

1. Database System Concepts (7th ed),Silberschatz, Korth, Sudarshan · McGraw-Hill · 2019,ISBN: 978-1-260-08450-4 · Theory foundation
2. PostgreSQL: Up and Running (3rd ed),Regina Obe, Leo Hsu · O'Reilly · 2017,ISBN: 978-1-4919-6341-8 · M1–M2 PG ops
3. SQL Performance Explained,Markus Winand · self-published · 2012,ISBN: 978-3-9503078-2-5 · M2 — gold-standard for indexing
4. MongoDB: The Definitive Guide (3rd ed),Shannon Bradshaw, Eoin Brazil, Kristina Chodorow · O'Reilly · 2019,ISBN: 978-1-4919-5446-1 · M3

Dr. K. BHAGAVAN
HoD-CSA 3/1/26
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Andhra Pradesh-522

5. Designing Data-Intensive Applications,Martin Kleppmann · O'Reilly · 2017,ISBN: 978-1-4493-7332-0 · M5 distribution theory
6. PostgreSQL High Performance Cookbook,Chitij Chauhan, Dinesh Kumar · Packt · 2017,ISBN: 978-1-78539-998-1

CLOUD ADMINISTRATION & CLOUD DEVOPS

COURSE CODE: 26CA5104E
COURSE NAME : Cloud Admin & Cloud DevOps
L-T-P-S :3-0-3-0
PRE-REQ : NIL

CO. No	Course Outcome Description	PO	BTL
CO1	Apply cloud computing fundamentals and AWS core services (EC2, S3, VPC, IAM, RDS) to provision a multi-tier application.	PO1,PO4	3
CO2	Apply Infrastructure as Code with Terraform — providers, modules, state, workspaces — to manage cloud resources reproducibly.	PO1,PO4	3
CO3	Analyze CI/CD pipeline patterns (GitHub Actions, GitLab CI) to design end-to-end deployment workflows with quality gates.	PO2,PO4	4
CO4	Apply containerization (Docker) and orchestration fundamentals (ECS, Kubernetes basics) to deploy scalable applications.	PO1,PO4	3
CO5	Evaluate observability strategies — logging, metrics, tracing, alerting — to design production-grade monitoring.	PO2,PO3	5
CO6	Evaluate cost, security, and reliability trade-offs in cloud architectures to design well-architected workloads.	PO2,PO3,PO6	5

Syllabus:

Module No.	Syllabus

HoD-CSA 31/7/24
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M1	**Cloud Fundamentals & AWS Core** Cloud service models (IaaS/PaaS/SaaS); deployment models. AWS global infrastructure — regions, AZs, edge locations. IAM — users, groups, roles, policies, MFA. EC2, EBS, AMIs. VPC — subnets, route tables, IGW, NAT. S3 — buckets, ACLs, lifecycle.
M2	**Infrastructure as Code with Terraform** Terraform basics — providers, resources, data sources. State management — local vs remote (S3 + DynamoDB lock). Variables, outputs, locals. Modules — composition, registry. Workspaces. Lifecycle rules. Plan/apply/destroy flow. Drift detection.
M3	**CI/CD Pipelines** GitHub Actions — workflows, jobs, steps, runners. Reusable workflows. Secrets and OIDC. Matrix builds. Deployment strategies — rolling, blue/green, canary. Pipeline patterns — build, test, scan, deploy. GitOps (ArgoCD intro). Approval gates.
M4	**Containers & Orchestration** Docker — images, layers, multi-stage builds, networking, volumes, compose. Container registries (ECR, Docker Hub). AWS ECS — task definitions, services, ALB integration. Kubernetes basics — pods, deployments, services, ingress. EKS overview.
M5	**Observability & SRE Patterns** Three pillars — logs, metrics, traces. CloudWatch metrics + Logs Insights. Prometheus + Grafana. Distributed tracing (X-Ray, OpenTelemetry). SLOs, SLIs, error budgets. Alerting fatigue management. Incident response basics.
M6	**Well-Architected: Cost, Security, Reliability** AWS Well-Architected Framework — six pillars. Cost optimization — right-sizing, reserved/spot, S3 storage classes. Security — encryption (KMS), Secrets Manager, GuardDuty, Security Hub. Reliability — multi-AZ, auto-scaling, disaster recovery (RPO/RTO). Compliance basics.

Reference Books:

1. AWS Certified Solutions Architect Study Guide (Associate, 4th ed),Ben Piper, David Clinton · Sybex · 2023,ISBN: 978-1-119-98262-0 · Maps to entire course
2. Terraform: Up & Running (3rd ed),Yevgeniy Brikman · O'Reilly · 2022,ISBN: 978-1-098-11674-7 · M2
3. Continuous Delivery,Jez Humble, David Farley · Addison-Wesley · 2010,ISBN: 978-0-321-60191-9 · M3 — classic
4. Docker Deep Dive,Nigel Poulton · self-published · 2023,ISBN: 978-1-916-71607-7 · M4
5. Site Reliability Engineering,Beyer, Jones, Petoff, Murphy · O'Reilly · 2016,ISBN: 978-1-491-92912-4 · Free online · M5
6. AWS Well-Architected Framework whitepaper,AWS,aws.amazon.com/architecture/well-architected · M6

HoD-CSA 3/1/26

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Pradesh-522502

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

PROFESSIONAL COMMUNICATION & MATHEMATICAL METHODS

COURSE CODE: 26UC5103E

COURSE NAME : Professional Communications & mathematical methods

L-T-P-S :0-0-3-0

PRE-REQ : NIL

CO. No	Course Outcome Description	PO	BTL
CO1	Apply technical writing principles — clarity, structure, audience-awareness — to produce design docs, READMEs, runbooks, and incident reports of professional quality.	PO1,PO3	3
CO2	Apply spoken communication skills — technical presentation, code walkthrough, lightning talks, stakeholder briefings — including handling Q&A and interview-style explanations.	PO1,PO5	3
CO3	Apply linear algebra fundamentals — vectors, matrices, eigenvalues, SVD — needed across ML, computer graphics, cryptography, and HPC.	PO1,PO2	3
CO4	Apply probability & statistics fundamentals — random variables, distributions, expectation, variance, conditional probability, Bayes' theorem — needed for ML and analytics.	PO1,PO2	3
CO5	Analyze discrete mathematics topics — sets, logic, graphs, combinatorics, number theory — needed for algorithms, cryptography, and database theory.	PO1,PO2	4
CO6	Apply soft-skill workplace competencies — meeting hygiene, written async communication, async-first collaboration, conflict resolution, code-review etiquette.	PO1,PO2,PO5	3

Syllabus:

Module No.	Syllabus

HoD-CSA 31/1/26
Dr. K
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M1	**Technical Writing Workshop** Technical writing principles — clarity, brevity, audience modeling. Documentation taxonomy (Diátaxis framework — tutorials, how-tos, reference, explanation). Writing READMEs that actually get read. Architecture decision records (ADRs). Runbook structure. Incident-report writing (5 Whys, blameless post-mortems). Style guides — Google Developer Documentation Style Guide, Microsoft Manual of Style. Tools — Markdown, Mermaid for inline diagrams, Excalidraw for sketches. Prose anti-patterns — passive voice, undefined acronyms, missing caveats.
M2	**Linear Algebra for CS** Vectors and vector operations. Matrices as linear maps. Matrix multiplication; transpose; inverse. Determinants and trace. Linear independence, basis, rank. Eigenvalues & eigenvectors — geometric intuition. Singular Value Decomposition (SVD) — full, reduced, truncated. PCA derivation from SVD. Application to ML — embeddings, recommender systems. Computational efficiency — sparse vs dense, BLAS levels. Practical: NumPy for all of the above.
M3	**Probability & Statistics** Random variables — discrete vs continuous. Common distributions — Bernoulli, Binomial, Poisson, Normal, Exponential. Expectation, variance, covariance, correlation. Joint, marginal, conditional probability. Bayes' theorem with worked examples (medical-test paradox). Law of large numbers; central limit theorem. Maximum likelihood estimation. Applications — A/B testing, naive Bayes classifier, MCMC intro. Practical: scipy.stats.
M4	**Spoken Communication Workshop** Technical presentation skills — slide hygiene (one idea per slide, no walls of text), the "tell-them-three-times" structure. Code walkthrough — narrating intent, signposting transitions, fielding questions. Lightning talks (5 min) — picking one idea, ruthlessly cutting. Stakeholder briefings — the BLUF (Bottom Line Up Front) structure for executives. Interview-style explanations — STAR (Situation/Task/Action/Result) for behaviorals; whiteboard explanation for system-design. Handling difficult questions — "I don't know but here's how I'd find out." Indian-accent neutralization is NOT a goal — clarity and confidence are.
M5	**Discrete Mathematics** Sets, relations, functions. Propositional & predicate logic — truth tables, equivalence, normal forms. Graph theory — directed/undirected, walks/paths/cycles, connectivity, trees, planarity, coloring. Combinatorics — permutations, combinations, pigeonhole, generating functions intro. Number theory — modular arithmetic, GCD, primes, RSA-relevant theorems (Euler's totient, Fermat's little theorem, Chinese Remainder). Big-O recap with formal proof discipline. Applications — graph algorithms, cryptography (RSA worked end-to-end), DB normalization theory.
M6	**Workplace Skills Workshop** Async-first collaboration — when to chat vs ticket vs doc vs meet. Meeting hygiene — agenda before meeting, decision log, action items with owners. Written async — Slack/Teams etiquette, thread discipline, status updates, blocker escalation. Code-review etiquette — what to say (and not say), how to receive feedback. Conflict resolution — non-violent communication, "disagree and commit." 1:1s with managers — what to bring, how to ask for feedback. Negotiation basics —

HoD-CSA 3/7/26

Dr. K. Bhagavan
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	anchoring, BATNA. Personal brand — GitHub profile, LinkedIn essentials, public writing.

Reference Books:

1. The Elements of Style (4th ed),William Strunk Jr., E.B. White · Pearson · 1999,ISBN: 978-0-205-30902-3 · Coverage: M1 — classic style guide
2. Docs for Developers — An Engineer's Field Guide to Technical Writing,Bhatti, Corleissen, Lambourne, Nunez, Waters · Apress · 2021,ISBN: 978-1-4842-7216-3 · Coverage: M1 modern tech writing
3. Linear Algebra Done Right (4th ed),Sheldon Axler · Springer · 2024,ISBN: 978-3-031-41025-3 · Coverage: M2 — free PDF on author's site
4. Introduction to Probability (2nd ed),Joseph K. Blitzstein, Jessica Hwang · CRC Press · 2019,ISBN: 978-1-138-36991-7 · Coverage: M3
5. Discrete Mathematics and Its Applications (8th ed),Kenneth H. Rosen · McGraw-Hill · 2019 (Indian edition),ISBN: 978-9-389-53884-0 · Coverage: M5 — gold-standard Indian-market text
6. The Pyramid Principle (3rd ed),Barbara Minto · Pearson · 2009,ISBN: 978-0-273-71051-6 · Coverage: M1, M4 — the executive-communication classic

PYTHON PROGRAMMING II — DSA & PROBLEM SOLVING

COURSE CODE: 26CA5205E
COURSE NAME : Python Programming - 2
L-T-P-S :3-0-3-0
PRE-REQ : PYP1

CO. No	Course Outcome Description	PO	BTL
CO1	Apply linear data structures (arrays, linked lists, stacks, queues, deques) implemented in Python to solve common engineering problems.	PO1,PO4	3
CO2	Analyze tree structures (BST, AVL, heaps, tries) and graph representations (adjacency list/matrix) to choose the right structure per problem.	PO2,PO4	4
CO3	Analyze sorting and searching algorithms — quicksort, mergesort, heapsort, binary search variants — for correctness and asymptotic complexity.	PO1,PO2	4
CO4	Analyze graph algorithms (BFS, DFS, Dijkstra, Bellman-Ford, MST, topological sort) and their real-world applications.	PO2,PO4	4

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO5	Evaluate dynamic programming, greedy, and backtracking strategies to design efficient solutions for combinatorial problems.	PO1,PO2	5
CO6	Evaluate algorithm performance via empirical benchmarking and profile-driven optimization in Python.	PO1,PO2,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Linear Data Structures** Arrays vs lists; amortized analysis. Linked lists — singly, doubly, circular. Stacks, queues, deques. Hash tables — open addressing, chaining. Big-O fundamentals; worst/avg/amortized analysis. collections.deque, heapq, bisect in stdlib.
M2	**Trees & Heaps** Binary trees, traversals (pre/in/post/level). BST operations and balancing. AVL trees, Red-Black overview. Heaps and priority queues. Tries for prefix problems. Segment trees and Fenwick trees (intro).
M3	**Sorting & Searching** Comparison sorts — quicksort, mergesort, heapsort, Timsort (Python's). Linear sorts — counting, radix, bucket. Stability and in-place. Binary search variants. External sorting. Lower bounds.
M4	**Graph Algorithms** Graph representations. BFS, DFS & applications. Topological sort. Single-source shortest path — Dijkstra, Bellman-Ford. All-pairs — Floyd-Warshall. MST — Kruskal, Prim. Union-Find with path compression. Graph problems on real datasets.
M5	**DP, Greedy & Backtracking** Recursion + memoization → DP. Classical DPs — knapsack, LIS, LCS, edit distance, matrix chain. Greedy — interval scheduling, Huffman. Backtracking — N-queens, sudoku, subset sum. State-space pruning.
M6	**Benchmarking & Optimization** Profiling with cProfile, line_profiler, memory_profiler. timeit microbenchmarks. Big-O empirical verification. NumPy vectorization for speedups. Cython / C-extensions intro. LeetCode strategy.

Reference Books:

1. Introduction to Algorithms (4th ed),Cormen, Leiserson, Rivest, Stein · MIT Press · 2022,ISBN: 978-0-262-04630-5 · CLRS — gold standard reference
2. Algorithms (4th ed),Robert Sedgewick, Kevin Wayne · Addison-Wesley · 2011,ISBN: 978-0-321-57351-3

Dr. K. BHAGAVAN
HoD-CSA 31/7/26
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

3. Problem Solving with Algorithms and Data Structures Using Python,Bradley Miller, David Ranum · Franklin, Beedle · 2011,ISBN: 978-1-59028-257-1
4. Cracking the Coding Interview (6th ed),Gayle Laakmann McDowell · CareerCup · 2015,ISBN: 978-0-9847828-5-7 · placement prep
5. Elements of Programming Interviews in Python,Aziz, Lee, Prakash · CreateSpace · 2016,ISBN: 978-1-5375-1394-2
6. Competitive Programming (4th ed),Steven Halim, Felix Halim · Lulu · 2020

FULL STACK DEVELOPMENT I — PYTHON WEB & DATABASE

COURSE CODE: 26CA5206E
COURSE NAME : Full Stack Development - 1
L-T-P-S :3-0-6-0
PRE-REQ : PYP1

CO. No	Course Outcome Description	PO	BTL
CO1	Apply HTTP fundamentals, HTML5, CSS3, and modern JavaScript (ES6+) to build accessible, responsive frontends.	PO1,PO4	3
CO2	Apply Python web frameworks (Flask & Django) — routing, views, templates, forms, sessions — to build full-featured web applications.	PO1,PO4	3
CO3	Analyze ORM patterns (SQLAlchemy & Django ORM) and database migrations to design clean data layers.	PO2,PO3	4
CO4	Apply REST API design principles (resources, methods, status codes, versioning) to build well-documented backend APIs.	PO1,PO4	3
CO5	Analyze authentication & authorization patterns (sessions, JWT, OAuth) to implement secure user flows.	PO2,PO4	4
CO6	Evaluate deployment strategies (gunicorn + nginx, Docker, cloud PaaS) to ship a production-ready web application.	PO2,PO4,PO5	5

Syllabus:

Module No.	Syllabus
M1	**Web Foundations & Frontend** HTTP/HTTPS, request-response cycle, methods, status codes. HTML5 semantic tags, forms, accessibility (ARIA). CSS3 — flexbox, grid, responsive design, custom properties. JavaScript ES6+ — let/const, arrow functions, destructuring, modules, async/await, fetch. Browser DevTools.
M2	**Flask — Lightweight Web Framework**

HoD-CSA 31/7/26
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	Flask app factory, blueprints, routing. Jinja2 templating, template inheritance. WTForms for form handling. Sessions and flash messages. Configuration management. Logging. Flask extensions overview (Flask-Login, Flask-Migrate).
M3	**Database & ORM** SQLAlchemy core vs ORM. Models, relationships (1:1, 1:N, N:N). Sessions, query API. Migrations with Alembic / Flask-Migrate. Connection pooling. Indexing strategy from the ORM. Transactions and rollbacks.
M4	**Django — Batteries-Included Framework** Django project structure, apps. Django ORM — models, querysets, managers. Class-based views vs function views. Forms framework. Admin interface. Middleware. Django REST Framework basics — serializers, viewsets, routers.
M5	**Authentication & Authorization** Session-based auth (Flask-Login, Django auth). Password hashing (bcrypt, argon2). JWT — structure, signing, verification. OAuth2 / OpenID Connect — flows, providers (Google, GitHub). Role-based access control. CSRF protection. Rate limiting.
M6	**Deployment & Production** WSGI/ASGI servers (gunicorn, uvicorn). nginx reverse proxy, static files, SSL/TLS. Docker for web apps. Environment configuration, secrets. Deploying to PaaS (Render, Railway, Fly.io). Logging & monitoring. Background jobs with Celery + Redis intro.

Reference Books:

1. Flask Web Development (2nd ed),Miguel Grinberg · O'Reilly · 2018,ISBN: 978-1-491-99173-2 · Definitive Flask reference
2. Django for Professionals,William S. Vincent · self-published · 2022,ISBN: 978-1-7355159-1-7
3. Two Scoops of Django 3.x,Daniel Roy Greenfeld, Audrey Roy Greenfeld · self-published · 2020,ISBN: 978-0-692-91572-9 · Best practices
4. Essential SQLAlchemy (2nd ed),Jason Myers, Rick Copeland · O'Reilly · 2015,ISBN: 978-1-491-91646-9
5. Eloquent JavaScript (4th ed),Marijn Haverbeke · No Starch Press · 2024,ISBN: 978-1-7185-0357-0 · Free online
6. RESTful Web APIs,Leonard Richardson, Mike Amundsen · O'Reilly · 2013,ISBN: 978-1-449-35806-8

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

COURSE CODE: 26CA5207E
COURSE NAME : AI & Machine Learning
L-T-P-S : 3-0-6-0
PRE-REQ : PYP1

CO. No	Course Outcome Description	PO	BTL

HoD-CSA

Dr. K. B...
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Andhra Pradesh-522502.

CO1	Apply pandas, NumPy, and scikit-learn for end-to-end ML workflows on tabular data.	PO1,PO2	3
CO2	Apply supervised learning algorithms (regression, classification) to build predictive models.	PO1,PO4	3
CO3	Analyze unsupervised techniques (clustering, dimensionality reduction) to discover structure in data.	PO2,PO4	4
CO4	Analyze model evaluation metrics, cross-validation, and bias-variance trade-offs to diagnose model behavior.	PO2,PO3	4
CO5	Evaluate feature engineering, hyperparameter tuning, and ensemble methods for high-performance models.	PO2,PO3	5
CO6	Evaluate ML deployment patterns (REST API, batch scoring, monitoring) for production-ready solutions.	PO2,PO4,PO5	5

Syllabus:

Module No.	Syllabus
M1	**ML Foundations & Tooling** ML taxonomy — supervised, unsupervised, reinforcement. NumPy, pandas mastery. Matplotlib + Seaborn for EDA. scikit-learn API conventions. Train/val/test split strategies. Data leakage pitfalls.
M2	**Supervised Learning** Linear & logistic regression. Decision trees. Random forest. Gradient boosting (XGBoost, LightGBM). Support vector machines. K-NN. Loss functions, regularization (L1/L2). Class imbalance handling.
M3	**Unsupervised Learning** K-means, hierarchical clustering, DBSCAN. PCA, t-SNE, UMAP. Anomaly detection — Isolation Forest. Association rules. Choosing K, evaluating clusters. Topic modelling intro (LDA).
M4	**Evaluation & Model Diagnostics** Classification metrics — precision, recall, F1, ROC-AUC, PR curves. Regression metrics — RMSE, MAE, R^2. Cross-validation — k-fold, stratified, time-series. Bias-variance, learning curves. Confusion matrix interpretation.
M5	**Feature Eng & Hyperparameter Tuning** Feature engineering — encoding, scaling, polynomials, interaction terms. Feature importance — SHAP, permutation. Pipelines & ColumnTransformer. Grid search, random search, Bayesian optimization (Optuna). Stacking and blending.
M6	**Deployment & Monitoring** Model serialization (pickle, joblib, ONNX). FastAPI for ML APIs. Batch vs online inference. Model registry (MLflow). Data drift & model drift detection. A/B testing. Latency & throughput optimization.

Reference Books:

HoD-CSA 31/1/26

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed),Aurélien Géron · O'Reilly · 2022,ISBN: 978-1-098-12597-8 · Maps to entire course
2. Pattern Recognition and Machine Learning,Christopher Bishop · Springer · 2006,ISBN: 978-0-387-31073-2 · Theory foundation
3. The Elements of Statistical Learning (2nd ed),Hastie, Tibshirani, Friedman · Springer · 2009,ISBN: 978-0-387-84857-0 · Free PDF
4. Python Machine Learning (3rd ed),Sebastian Raschka · Packt · 2019,ISBN: 978-1-78995-575-0
5. Approaching (Almost) Any Machine Learning Problem,Abhishek Thakur · self-published · 2020,ISBN: 978-93-5419-211-3 · Practical Kaggle wisdom
6. Interpretable Machine Learning,Christoph Molnar · 2nd ed online · 2022,christophm.github.io/interpretable-ml-book · Free

OFFENSIVE SECURITY & ETHICAL HACKING

COURSE CODE: 26CA5208E
COURSE NAME : Offensive Security
L-T-P-S :3-0-3-0
PRE-REQ : LXOS

CO. No	Course Outcome Description	PO	BTL
CO1	Apply offensive security methodology (PTES, OWASP Testing Guide) to plan and scope pentesting engagements.	PO1,PO4	3
CO2	Apply reconnaissance techniques — passive (OSINT) and active (port scanning, banner grabbing) — to map target attack surface.	PO1,PO4	3
CO3	Analyze OWASP Top 10 web vulnerabilities (injection, XSS, IDOR, SSRF, broken auth) to identify and exploit them in test environments.	PO2,PO4	4
CO4	Analyze network and system exploitation techniques — Metasploit, Mimikatz, privilege escalation paths — for authorized testing.	PO2,PO4	4
CO5	Evaluate post-exploitation strategies — persistence, lateral movement, exfiltration — and the corresponding defensive controls.	PO2,PO6	5
CO6	Evaluate findings, write professional pentest reports with risk ratings, and present remediation guidance to stakeholders.	PO2,PO5,PO6	5

Syllabus:

Module No.	Syllabus

Dr. K. BHAGAVAN 3/12/24
HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M1	**Pentesting Methodology & Legal** Hacker mindset, ethics, legal framework (IT Act 2000 in India, CFAA in US). Engagement scoping, rules of engagement (RoE). PTES phases. OWASP Testing Guide. Report deliverables. Kali Linux setup. Lab environment (VirtualBox + Kali + DVWA + Metasploitable + HTB).
M2	**Reconnaissance & Scanning** Passive recon — Google dorking, Shodan, theHarvester, Maltego. Active recon — Nmap (TCP, UDP, OS, version), masscan. Service enumeration. Banner grabbing. DNS recon — dig, dnsenum. Subdomain enumeration — Sublist3r, Amass. Web fingerprinting — whatweb, wappalyzer.
M3	**Web Application Pentesting** OWASP Top 10 deep-dive. SQL injection (manual + sqlmap). XSS — reflected, stored, DOM. CSRF. IDOR. SSRF. XXE. Broken authentication. Burp Suite mastery — proxy, repeater, intruder, scanner. Mobile-app OWASP top 10 intro.
M4	**Network & System Exploitation** Metasploit Framework — modules, payloads, Meterpreter. Vulnerability exploitation. Buffer overflows (intro). Wireless attacks (deauth, evil twin) — Aircrack-ng. Linux privilege escalation — SUID, sudo misconfig, kernel exploits. Windows privesc — Mimikatz, token manipulation.
M5	**Post-Exploitation & Lateral Movement** Persistence techniques — scheduled tasks, registry, services. Credential dumping (Mimikatz, secretsdump). Lateral movement — pass-the-hash, pass-the-ticket, PsExec. Tunneling and pivoting (chisel, ssh tunnels). Active Directory attacks — Kerberoasting, AS-REP roasting (intro). Defensive controls — EDR, behavioral detection.
M6	**Reporting & Remediation** Risk ratings — CVSS 3.1, OWASP risk rating. Executive summary writing. Technical findings format. Remediation prioritization. Retest methodology. Bug bounty platforms (HackerOne, Bugcrowd) — disclosure flow. Final report writing for RedTeamLab.

Reference Books:

1. The Web Application Hacker's Handbook (2nd ed),Dafydd Stuttard, Marcus Pinto · Wiley · 2011,ISBN: 978-1-118-02647-2 · Bible of webapp pentesting
2. The Hacker Playbook 3,Peter Kim · Secure Planet · 2018,ISBN: 978-1-980-90975-9
3. Penetration Testing — A Hands-On Introduction to Hacking,Georgia Weidman · No Starch Press · 2014,ISBN: 978-1-59327-564-8
4. Real-World Bug Hunting,Peter Yaworski · No Starch Press · 2019,ISBN: 978-1-59327-861-8
5. Red Team Field Manual (RTFM),Ben Clark · self-published · 2014,ISBN: 978-1-49414-617-3
6. OWASP Testing Guide v4.2,OWASP Foundation,owasp.org/www-project-web-security-testing-guide ·

31/7/26
HoD-CSA
Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

FULL STACK DEVELOPMENT II — FASTAPI · MONGODB · REACT

COURSE CODE: 26CA5309E
COURSE NAME : Full Stack Development - 2
L-T-P-S :3-0-6-0
PRE-REQ : FSD1

CO. No	Course Outcome Description	PO	BTL
CO1	Apply FastAPI with Pydantic, async/await, dependency injection, and OpenAPI auto-docs to build modern Python APIs.	PO1,PO4	3
CO2	Analyze async Python patterns (asyncio, async DB drivers) for high-throughput services.	PO2,PO4	4
CO3	Apply MongoDB advanced features — aggregation, transactions, change streams, indexing.	PO1,PO2	3
CO4	Apply React with hooks and modern state management (TanStack Query) to build SPAs.	PO1,PO4	3
CO5	Analyze WebSocket and SSE patterns to design real-time collaborative apps.	PO2,PO3	4
CO6	Evaluate frontend deployment (CDN, edge, SSR with Next.js) to ship a production SPA.	PO2,PO4,PO5	5

Syllabus:

Module No.	Syllabus
M1	**FastAPI Fundamentals** FastAPI path operations, query params, request bodies. Pydantic models, validation, settings. Dependency injection. Background tasks. OpenAPI auto-docs. CORS. File uploads.
M2	**Async Python Deep Dive** asyncio fundamentals — event loop, coroutines, tasks. async/await. Concurrency primitives — gather, semaphores. Async HTTP (httpx). Async DB drivers (motor, asyncpg).
M3	**MongoDB for Full Stack** Motor (async). Aggregation pipeline mastery — $match, $lookup, $facet, $bucket. Indexing strategy. Change streams for real-time. Transactions in replica sets. GridFS.
M4	**React Modern Patterns** React 18 — hooks, suspense, concurrent features. State — useReducer, Context, Zustand or Redux Toolkit. TanStack Query for server state. React Router v6. React Hook Form + Zod.

HoD-CSA 31/3/26
Dr. K. [illegible] CAVA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
[illegible] Pradesh-522502.

M5	**Real-Time & WebSockets** WebSocket protocol. FastAPI WebSocket endpoints. SSE. Real-time patterns — pub/sub, presence, optimistic UI. Redis pub/sub. CRDT intro for conflict resolution.
M6	**Modern Deployment & SSR** Next.js basics — App Router, server components, server actions. SSR vs SSG vs ISR. Vercel deploy. CDN strategy. Image optimization. Core Web Vitals.

Reference Books:

1. FastAPI: Modern Python Web Development,Bill Lubanovic · O'Reilly · 2023,ISBN: 978-1-098-13509-0
2. Asyncio in Python,Caleb Hattingh · O'Reilly · 2020,ISBN: 978-1-492-07533-2
3. MongoDB: The Definitive Guide (3rd ed),Bradshaw, Brazil, Chodorow · O'Reilly · 2019,ISBN: 978-1-4919-5446-1,Learning React (2nd ed)
4. Alex Banks, Eve Porcello · O'Reilly · 2020,ISBN: 978-1-4920-5172-5
5. Real-Time Web Application Development,Rami Vemula · Apress · 2017,ISBN: 978-1-4842-3270-6
6. Next.js — Beginner's Guide to Production,Vercel docs + Lee Robinson,nextjs.org/learn · Free

APPLIED DEEP LEARNING, LLM ENGINEERING & RAG

COURSE CODE: 26CA5310E
COURSE NAME : Applied DL, LLM Eng & RAG
L-T-P-S :3-0-0-6
PRE-REQ : AIML

CO. No	Course Outcome Description	PO	BTL
CO1	Apply PyTorch with autograd, optimizers, training loops to build neural networks for vision and text.	PO1,PO2	3
CO2	Analyze transformer architectures and pretrained models from Hugging Face for downstream tasks.	PO2,PO3	4
CO3	Apply LLM API patterns — prompts, function-calling, structured output — using OpenAI, Anthropic, and OSS LLMs.	PO1,PO4	3
CO4	Analyze RAG architecture — chunking, embeddings, vector search, reranking — for accurate Q&A.	PO2,PO3	4
CO5	Evaluate RAG quality with frameworks (RAGAs, TruLens) and design retrieval improvements based on metrics.	PO2,PO4	5
CO6	Evaluate fine-tuning options (LoRA, instruction tuning) vs prompting trade-offs for production LLM systems.	PO2,PO4,PO5	5

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522 502

Syllabus:

Module No.	Syllabus
M1	**PyTorch & DL Foundations** PyTorch tensors, autograd, optimizers. MLPs, CNNs, RNNs from scratch. Training loops with validation. GPU acceleration. PyTorch Lightning intro. Reproducibility.
M2	**Transformers & Hugging Face** Attention mechanism. Transformer architecture. BERT, GPT, T5. Hugging Face Transformers — pipelines, AutoModel, tokenizers. Fine-tuning. Quantization basics.
M3	**LLM API Patterns** OpenAI & Anthropic APIs. System prompts. Few-shot. Function-calling / tool use. Structured outputs (JSON mode). Streaming. Cost optimization. Self-hosted LLMs (Ollama, vLLM).
M4	**RAG Architecture** RAG overview. Document loading (PDF, HTML, Markdown). Chunking strategies. Embeddings — OpenAI, Cohere, sentence-transformers. Vector DBs — FAISS, Chroma, pgvector. Hybrid search (BM25 + dense). Reranking.
M5	**Evaluation & Improvement** RAG eval — context relevance, answer faithfulness. Frameworks — RAGAs, TruLens, LangSmith. LLM-as-judge. Test-set construction. A/B prompts. Hallucination detection.
M6	**Fine-Tuning & Production** When to fine-tune vs prompt. LoRA / QLoRA. Instruction tuning. Production patterns — caching, fallback chains, prompt versioning, observability. Cost & latency budgets.

Reference Books:

1. Deep Learning with PyTorch (2nd ed),Eli Stevens, Luca Antiga, Thomas Viehmann · Manning · 2020,ISBN: 978-1-617-29526-3
2. Natural Language Processing with Transformers, Lewis Tunstall, Leandro von Werra, Thomas Wolf · O'Reilly · 2022,ISBN: 978-1-098-13679-0
3. Hands-On Large Language Models,Jay Alammar, Maarten Grootendorst · O'Reilly · 2024,ISBN: 978-1-098-15096-9
4. Deep Learning,Goodfellow, Bengio, Courville · MIT Press · 2016,ISBN: 978-0-262-03561-3 · Free online
5. Building LLMs for Production,Louis-François Bouchard, Louie Peters · Towards AI · 2024,ISBN: 978-1-961-69100-3
6. Transformers for Natural Language Processing (3rd ed),Denis Rothman · Packt · 2024,ISBN: 978-1-83508-871-0

HIGH PERFORMANCE COMPUTING & CLUSTER MANAGEMENT

Dr. K. BHA... HoD-CSA 31/1/22
HOD-Computer Science & Engineering
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

COURSE CODE: 26CA5311E
COURSE NAME : HPC Admin & Cluster Mgmt
L-T-P-S :3-0-3-0
PRE-REQ : LXOS

CO. No	Course Outcome Description	PO	BTL
CO1	Apply HPC architecture concepts (interconnects, NUMA, parallel filesystems) to understand cluster design.	PO1,PO3	3
CO2	Apply parallel programming with OpenMP and MPI to write multi-threaded and multi-process code.	PO1,PO2	3
CO3	Apply GPU programming with CUDA + PyTorch for accelerated numerical workloads.	PO1,PO2	3
CO4	Analyze SLURM workload manager — partitions, jobs, queues, accounting — to operate shared HPC clusters.	PO1,PO2	4
CO5	Analyze containerized HPC (Singularity/Apptainer, Kubernetes for ML) for portable scientific workloads.	PO2,PO4	4
CO6	Evaluate scaling efficiency (strong vs weak scaling), performance bottlenecks, and tuning strategies for HPC apps.	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**HPC Architecture & Cluster Setup** HPC vs cloud computing. Cluster topology — head node, compute nodes, interconnect (InfiniBand, RoCE). NUMA architecture. Parallel filesystems — Lustre, GPFS, BeeGFS. CPU/GPU/memory profiles. Top500 supercomputers overview.
M2	**Parallel Programming — OpenMP & MPI** Parallelism — task vs data. OpenMP — pragmas, parallel for, reduction, sections. MPI — point-to-point (Send/Recv), collective (Bcast/Reduce/Allgather), communicators. Hybrid OpenMP+MPI. Race conditions, deadlocks. Profiling parallel code (gprof, Intel VTune).
M3	**GPU Computing — CUDA & PyTorch** GPU architecture — SMs, warps, memory hierarchy. CUDA programming — kernels, threads, blocks. cuBLAS, cuDNN. PyTorch on GPU — tensor ops, mixed precision (AMP). Distributed PyTorch (DDP). Multi-GPU training basics.
M4	**SLURM & Job Scheduling** SLURM architecture — slurmctld, slurmd, slurmdbd. Partitions, QOS, accounts. sbatch, srun, squeue, sacct. Job arrays, dependencies, pre-emption. Resource limits. Fair-share scheduling. Operator daily ops.
M5	**Containerized HPC & K8s for ML** Singularity / Apptainer — container building, MPI integration. Kubernetes basics — pods, deployments, services. Kubeflow / KubeRay for ML on K8s. NVIDIA GPU operator. Volcano scheduler. SLURM ↔ K8s comparison for ML workloads.

31/7/26
HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M6	**Performance Engineering** Roofline model. Strong vs weak scaling. Amdahl's law. I/O optimization (parallel I/O, MPI-IO). Memory hierarchy tuning. Profiling tools — perf, Nsight Systems. Energy efficiency (PUE, FLOP/W). Production runbook.

Reference Books:

1. Introduction to High Performance Computing for Scientists and Engineers,Georg Hager, Gerhard Wellein · CRC Press · 2010,ISBN: 978-1-4398-1192-4
2. Using OpenMP — Portable Shared Memory Parallel Programming,Chapman, Jost, Van Der Pas · MIT Press · 2007,ISBN: 978-0-262-53302-7
3. Using MPI (3rd ed),Gropp, Lusk, Skjellum · MIT Press · 2014,ISBN: 978-0-262-52739-2
4. CUDA C Programming,John Cheng, Max Grossman, Ty McKercher · Wrox · 2014,ISBN: 978-1-118-73932-7
5. Programming Massively Parallel Processors (4th ed),David Kirk, Wen-mei Hwu · Morgan Kaufmann · 2022,ISBN: 978-0-323-91231-0
6. Slurm Workload Manager Documentation,SchedMD,slurm.schedmd.com · Free

ADAPTIVE SOFTWARE ENGINEERING WITH MLOPS

COURSE CODE: 26CA5312E
COURSE NAME : Adaptive SE with MLOps
L-T-P-S :3-0-3-0
PRE-REQ : AIML

CO. No	Course Outcome Description	PO	BTL
CO1	Apply software engineering practices — clean architecture, design patterns, SOLID — to ML codebases.	PO1,PO3	3
CO2	Apply experiment tracking, model versioning, and reproducibility patterns (MLflow, DVC, Weights & Biases).	PO1,PO4	3
CO3	Analyze CI/CD for ML — testing models, data validation, model deployment pipelines.	PO2,PO3	4
CO4	Apply feature stores, model registries, and inference servers for production ML.	PO1,PO3	3
CO5	Evaluate model monitoring strategies — performance, data drift, concept drift — and automated retraining.	PO2,PO3	5
CO6	Evaluate AI agents and self-improving systems for adaptive software design.	PO2,PO3,PO4	5

Syllabus:

Module No.	Syllabus

Dr. K. BHAGAVAN
HoD-CSA 31/7/26
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Andhra Pradesh-522502.

M1	**SE for ML — Clean Architecture** Why SE matters for ML. Anti-patterns — notebook-only, glue code. SOLID for ML. Hexagonal/clean architecture. Domain modeling. Dependency injection. Type hints + Pydantic for ML APIs. Testing strategies for ML — unit, integration, property-based.
M2	**Experiment Tracking & Versioning** MLflow — runs, metrics, params, artifacts. Tracking server setup. Model registry — staging, production. DVC for data versioning. Weights & Biases — experiment dashboards. Reproducibility — Docker, lock files, deterministic seeds.
M3	**CI/CD for ML** CI for ML — model tests (smoke, robustness, fairness), data validation (Great Expectations). CD for ML — automated training pipelines, deployment gates. GitHub Actions for ML workflows. Pipeline orchestration (Airflow, Prefect, Dagster). Argo Workflows for K8s ML.
M4	**Inference & Feature Stores** Inference servers — TorchServe, Triton, BentoML. Batch vs online inference. Feature stores — Feast, Tecton — online vs offline features. Embedding stores — pgvector, Pinecone. Model latency optimization — quantization, batching, GPU vs CPU.
M5	**Monitoring & Drift** Model performance monitoring — accuracy, latency, throughput. Data drift — PSI, KS test, Wasserstein. Concept drift detection. Automated retraining triggers. Alerting strategies. Tools — Evidently, Arize, WhyLabs.
M6	**Adaptive Systems & AI Agents** Adaptive software — feedback loops, MAPE-K. AI-augmented development — Copilot, Claude Code. Self-improving systems. Agent frameworks — LangChain, AutoGen, Claude Agent SDK. Human-in-the-loop patterns. Production agent deployment.

Reference Books:

1. Designing Machine Learning Systems,Chip Huyen · O'Reilly · 2022,ISBN: 978-1-098-10796-3
2. Building Machine Learning Powered Applications,Emmanuel Ameisen · O'Reilly · 2020,ISBN: 978-1-492-04511-3
3. Machine Learning Engineering,Andriy Burkov · True Positive · 2020,ISBN: 978-1-9994-9385-0
4. Introducing MLOps,Mark Treveil et al. · O'Reilly · 2020,ISBN: 978-1-492-08328-3
5. Practical MLOps,Noah Gift, Alfredo Deza · O'Reilly · 2021,ISBN: 978-1-098-10301-9
6. Software Engineering at Google,Titus Winters et al. · O'Reilly · 2020,ISBN: 978-1-492-08279-8

GenAI API & AI Platform Eng

3/7/26
HoD-CSA
Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

COURSE CODE: 26CA6113E
COURSE NAME : GenAI API & AI Platform Eng
L-T-P-S :3-0-0-6
PRE-REQ : ADRG

CO. No	Course Outcome Description	PO	BTL
CO1	Apply production LLM API patterns — prompt versioning, prompt caching, batched inference, async + streaming, idempotency, retry with backoff — to build resilient GenAI services.	PO1,PO4	3
CO2	Analyze multi-model orchestration patterns — model routing by cost/latency/quality, fallback chains, OSS vs commercial trade-offs, specialized models (code, embeddings, vision) — to design optimal cost-quality envelopes.	PO2,PO3	4
CO3	Analyze GenAI safety threats and defenses — prompt injection, jailbreaks, data exfiltration, content moderation, PII detection & redaction, adversarial prompts — to harden production systems.	PO2,PO5	4
CO4	Apply LLM observability — token usage, latency P50/P95/P99, cache hit rates, error rates, cost attribution per tenant — and eval-driven development with golden datasets.	PO1,PO2	3
CO5	Evaluate AI gateway architectures (LiteLLM, Portkey, Kong AI Gateway) and multi-tenant patterns — quotas, rate limits per tenant, audit logging, encryption — for production-grade GenAI platforms.	PO4,PO6	5
CO6	Evaluate compliance and governance frameworks — NIST AI RMF, EU AI Act, India's DPDP Act 2023, model cards, system cards, red-teaming reports — for responsible deployment of GenAI systems.	PO2,PO5,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Production LLM API Patterns** Anatomy of an LLM API call — auth, request shape, streaming, response semantics. Provider deep-dives: Anthropic Messages API (system, tool_use, prompt caching), OpenAI Chat Completions + Responses API, Google Gemini API. Prompt versioning & templates with Jinja2 / LangChain Hub. Prompt caching — when it pays off, cache breakpoints. Batched inference — Anthropic batch API, OpenAI batch endpoint (50% cost). Async + streaming with Server-Sent Events. Token counting (tiktoken, anthropic.count_tokens) + rate-limit-aware retry with exponential backoff & jitter. Idempotency keys. Webhook patterns for long-running jobs.
M2	**Multi-Model Orchestration** Why multi-model — cost, latency, quality, redundancy. Model routing strategies: rule-based (task type), classifier-based, embedding-similarity, learned router. Fallback chains — primary —> cheap-model fallback —> cached response. OSS models — Llama 3, Mistral, Mixtral, Qwen — running with vLLM, Ollama, Text Generation Inference. Quantization — INT8, INT4, AWQ, GPTQ. Specialized models — Voyage AI & OpenAI embeddings, Cohere Rerank, Anthropic claude-3-5-haiku for cheap

HoD-CSA 3/1/26
Dr. K. BHARGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	classification. Cost-quality envelope plots. Pareto frontiers. Real-world routers (Martian, NotDiamond) for inspiration.
M3	**GenAI Safety Engineering** Threat model for LLM apps — OWASP Top 10 for LLM Applications. Prompt injection — direct, indirect (via retrieval/tool outputs). Jailbreaks — DAN, Skeleton Key, multi-turn manipulation. Defenses — input sanitization, prompt isolation, instruction-following classifiers (Anthropic prompt-shield, Lakera). Content moderation pipelines — OpenAI moderation, AWS Bedrock Guardrails. PII detection & redaction (Microsoft Presidio, AWS Comprehend). Output filtering — toxicity, copyright, regulated information. Constitutional AI principles. Adversarial robustness eval. Red-team checklist.
M4	**Observability for GenAI** LLM telemetry pillars — input/output, tokens, latency, cost, model, prompt version, user/tenant, trace ID. Tracing standards — OpenTelemetry GenAI semantic conventions. Tools — LangSmith, Langfuse (self-hosted), Helicone, Arize Phoenix. Eval-driven dev — golden datasets, regression tests in CI, LLM-as-judge with reference answers. A/B testing prompts — chi-square / t-test on win rate. Drift detection on input distributions. Error categorization — model errors vs system errors vs user errors.
M5	**AI Gateways & Multi-Tenancy** AI gateway pattern — single ingress for all LLM traffic. Open-source: LiteLLM (100+ providers), Portkey, Kong AI Gateway. Virtual keys — per-tenant API keys mapped to provider keys. Multi-tenant data model — tenants, users, projects, prompts, usage. Per-tenant quotas — token-based, request-based, dollar-based. Rate limiting strategies — fixed window, sliding window, token bucket. Audit logging schema. Encryption — TLS in transit, AES-256 at rest, BYOK with KMS. SOC 2 / ISO 27001 considerations. Tenant isolation models — shared, dedicated namespace, dedicated infra.
M6	**Compliance & Governance** NIST AI Risk Management Framework — Map / Measure / Manage / Govern. EU AI Act — risk tiers (unacceptable, high-risk, limited-risk, minimal-risk), conformity assessments. India's DPDP Act 2023 — consent, data fiduciary obligations, cross-border transfers. Bias auditing tools (AIF360, Fairlearn). Model cards (Mitchell et al.) and system cards (Anthropic). Red-team reports — methodology, finding-severity scoring, disclosure. Incident response for AI — playbook structure, post-mortem template. Regulatory horizon — UK AI Bill, China's interim Generative AI measures.

Reference Books:

1. Building LLMs for Production Louis-François Bouchard, Louie Peters · Towards AI · 2024 ISBN: 978-1-961-69100-3
2. Generative AI on AWS Chris Fregly, Antje Barth, Shelbee Eigenbrode · O'Reilly · 2023 ISBN: 978-1-098-15922-1
3. Designing Machine Learning Systems Chip Huyen · O'Reilly · 2022 ISBN: 978-1-098-10796-3

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

4. AI Snake Oil — What Artificial Intelligence Can Do, What It Can't, and How to Tell the Difference Arvind Narayanan, Sayash Kapoor · Princeton · 2024 ISBN: 978-0-691-24913-3
5. Hands-On Large Language Models Jay Alammar, Maarten Grootendorst · O'Reilly · 2024 ISBN: 978-1-098-15096-9
6. Prompt Engineering for LLMs John Berryman, Albert Ziegler · O'Reilly · 2024 ISBN: 978-1-098-15643-5
7. The Alignment Problem Brian Christian · W. W. Norton · 2020 ISBN: 978-0-393-86833-3

Data Science, Analytics & Viz

COURSE CODE: 26CA6114E
COURSE NAME : Data Science, Analytics & Viz
L-T-P-S :3-0-3-0
PRE-REQ : AIML

CO. No	Course Outcome Description	PO	BTL
CO1	Apply the analytics workflow — define metric, collect data, wrangle, analyze, visualize, present — to deliver a complete analytics deliverable for a real stakeholder question.	PO1,PO2	3
CO2	Apply advanced SQL (window functions, CTEs, recursive queries, time-series rollups) and pandas (merge, melt, pivot, groupby/agg, multi-index) to wrangle large analytical datasets.	PO1,PO4	3
CO3	Analyze data using inferential statistics — descriptive measures, hypothesis testing (t-test, chi-square, ANOVA), A/B-test design (sample size & power), regression for inference — to draw evidence-based conclusions.	PO2,PO4	4
CO4	Analyze visualization grammar (the layered grammar of graphics) and modern tools (Tableau, Power BI, Plotly Express, Plotly Dash) to design accurate and informative charts.	PO2,PO4	4
CO5	Evaluate dashboard design choices — information hierarchy, interactivity, drill-downs, narrative flow, mobile responsiveness — to ship a stakeholder-grade decision tool.	PO2,PO5	5
CO6	Evaluate the analytics-to-action loop — turning insights into hypotheses, hypotheses into experiments, experiments into product decisions — and the common anti-patterns (vanity metrics, p-hacking, Simpson's paradox).	PO2,PO4,PO6	5

Syllabus:

HoD-CSA 31/7/26
Dr. K. B. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Module No.	Syllabus
M1	**Analytics Workflow & Metric Design** Analytics lifecycle — question, data, analysis, presentation. Defining a "north-star metric" — input vs output, leading vs lagging. Stakeholder mapping. Data sources — REST APIs, file dumps, CDC streams, SaaS platforms, government open data. Jupyter Lab as analyst's workbench. pandas advanced — multi-index, merge (one-to-many, many-to-many), melt vs pivot, groupby/agg with custom aggregators, transform vs apply, window functions via rolling(), datetime handling, categorical dtype. Working with messy real data — schema drift, partial duplicates, encoding quirks, mixed units. Polars as a faster alternative.
M2	**SQL for Analytics** SQL window functions deep-dive — ROW_NUMBER, RANK, DENSE_RANK, LAG/LEAD, FIRST_VALUE/LAST_VALUE, NTH_VALUE, NTILE, framing clauses (ROWS vs RANGE). Common Table Expressions — basic, recursive (path queries, hierarchies, generating series). CASE expressions for in-query feature engineering. Pivot/unpivot patterns (CROSSTAB on PG, FILTER aggregates). Time-series rollups with date_trunc, generate_series, gap-filling. Funnel analysis with COUNT FILTER. Cohort analysis from event tables. Sessionization with LAG. Interview-grade SQL problems — top-N per group, running deltas, distinct counts with windows.
M3	**Statistics for Analytics** Descriptive statistics — mean, median, mode, variance, percentiles, IQR, skew, kurtosis. Distributions — Normal, Poisson, Binomial, log-normal; how to recognize them in real data. Sampling and the central limit theorem. Hypothesis testing framework — null/alternative, p-value, Type I/II error, statistical vs practical significance. Specific tests — one/two-sample t-test, paired t-test, chi-square independence, ANOVA. Multiple-comparison corrections (Bonferroni, BH). A/B-test design — sample-size calc, statistical power, MDE, sequential testing pitfalls. Regression for inference — coefficient interpretation, multicollinearity, heteroscedasticity. Causality basics — confounders, Simpson's paradox, intro to DAGs. Bootstrapping for non-parametric inference.
M4	**Visualization Grammar** Layered Grammar of Graphics (Wilkinson, Wickham). The 5 atomic elements — data, aesthetics, geoms, scales, facets. Choosing chart types by question: trend (line), comparison (bar/dot-plot), distribution (histogram, density, violin, ECDF), composition (stacked bar, treemap), relationship (scatter, hexbin), spatial (choropleth, dot-density). Color theory — sequential, diverging, qualitative palettes (ColorBrewer, viridis), accessibility (CVD-safe palettes). Edward Tufte principles — data-ink ratio, chartjunk, small multiples. Common malpractice — truncated y-axis, dual axes, 3D pies. Tableau Desktop deep-dive — calc fields, parameters, LOD expressions, dashboard actions. Power BI fundamentals — DAX measures vs columns, time-intelligence functions. Plotly Express + Plotly Dash for code-first dashboards.
M5	**Dashboard Design** Information hierarchy — F-pattern reading, focal points, KPI cards. Filter patterns — global filters, cross-filters, parameter actions, slicers. Drill-down design — drill-through, drill-up, hierarchical filters. Storytelling with data — Knaflic's framework:

Dr. HoD-CSA 31/1/26
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	context, meaning, action. Stephen Few's dashboard taxonomy — strategic, analytical, operational. Mobile-first dashboards — touch targets, breakpoints, vertical scrolling. Performance optimization — data-source extracts, in-memory mode, query consolidation, Tableau Hyper, Power BI aggregations. Accessibility — alt text for charts, color contrast, screen-reader labels.S
M6	**Analytics-to-Action** Insight → hypothesis → experiment loop. Working with PMs — decision documents, "what would change your mind?" Experiment design — control/treatment, randomization, blocking, blinding. Effect-size estimation. Cost-benefit framing for product changes. Stakeholder presentation skills — Pyramid Principle (Minto), executive summaries, the 30-second elevator brief. Anti-patterns — vanity metrics, p-hacking, HARKing, survivorship bias, Simpson's paradox in disaggregated reports. Anti-fragility in analytics — when not to recommend an action.

Reference Books:

1. Storytelling with Data Cole Nussbaumer Knaflic · Wiley · 2015 ISBN: 978-1-119-00225-3
2. Practical Statistics for Data Scientists (2nd ed) Peter Bruce, Andrew Bruce, Peter Gedeck · O'Reilly · 2020
3. ISBN: 978-1-492-07294-2
4. Trustworthy Online Controlled Experiments Ron Kohavi, Diane Tang, Ya Xu · Cambridge · 2020 ISBN: 978-1-108-72426-5
5. SQL for Smarties (5th ed) Joe Celko · Morgan Kaufmann · 2014 ISBN: 978-0-12-800761-7
6. Information Dashboard Design (2nd ed) Stephen Few · Analytics Press · 2013 ISBN: 978-1-938377-00-6
7. The Visual Display of Quantitative Information (2nd ed) Edward Tufte · Graphics Press · 2001 ISBN: 978-0-961-39214-7
8. Python for Data Analysis (3rd ed) Wes McKinney · O'Reilly · 2022 ISBN: 978-1-098-10403-0

Agentic AI + Full Stack + MCPs

COURSE CODE: 26CA6215E
COURSE NAME : Agentic AI + Full Stack + MCP
L-T-P-S :3-0-3-0
PRE-REQ : GAIP

CO. No	Course Outcome Description	PO	BTL

Dr. K. BHARGAVI
HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502
3/7/26

CO1	Apply core agent patterns — ReAct, function-calling, planner/executor split, reflection, self-critique — using modern frameworks (Anthropic Agent SDK, LangGraph, AutoGen, CrewAI) to solve multi-step tasks.	PO1,PO4	3
CO2	Analyze the Model Context Protocol (MCP) — server architecture, transports (stdio, HTTP+SSE), tools, resources, prompts — to design composable agent capabilities.	PO2,PO4	4
CO3	Apply full-stack engineering for agentic apps — streaming UIs, tool-call rendering, async background jobs, persistent agent state, conversation memory — using FastAPI + React + WebSocket/SSE.	PO1,PO4	3
CO4	Analyze multi-agent orchestration patterns — supervisor/worker, peer-to-peer, hierarchical teams, swarms — and their cost/latency/correctness trade-offs.	PO2,PO5	4
CO5	Evaluate agent safety and reliability — sandboxing, capability scoping, tool-call auditing, prompt injection through tools, infinite-loop prevention, human-in-the-loop checkpoints.	PO4,PO6	5
CO6	Evaluate agent observability and eval — trace completeness, step-by-step replay, LLM-as-judge for reasoning chains, agentic-eval frameworks (LangSmith, Braintrust, AgentBench).	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Agent Foundations & Patterns** What an "agent" actually is — perception —> reasoning —> action —> memory loop. Levels of autonomy (LLM as classifier, single tool-call, ReAct, full agent loop). Function-calling fundamentals — tool schemas, parameter validation, parallel tool calls, multi-turn tool use. ReAct (Reason+Act) prompting. Planner/Executor architectures. Reflection & self-critique loops. Anthropic Agent SDK — building blocks, tool definitions, conversation lifecycle. LangGraph state machines. AutoGen and CrewAI for multi-agent. Pitfalls — runaway loops, hallucinated tool args, lost context.
M2	**Model Context Protocol (MCP)** MCP overview — open protocol for connecting LLM agents to external tools, resources, and prompt templates. Architecture — host (Claude Desktop, Cursor, your app), client, server. Transports — stdio for local servers, HTTP+SSE for remote. Server capabilities — tools (callable functions), resources (read-only data), prompts (reusable templates). Building MCP servers in Python and TypeScript. Server discovery and capability negotiation. Authentication patterns for remote MCP servers. The MCP ecosystem — popular community servers (filesystem, GitHub, Slack, Postgres, Puppeteer). Composing multiple MCP servers in a single agent.

Dr. [illegible] HoD-CSA
HOD-Computer Science and Applications
Koneru Laksmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

M3	**Full-Stack for Agentic Apps** Streaming UI patterns — Server-Sent Events for token streams, WebSocket for bi-directional state, optimistic UI for tool results. Rendering tool calls and tool results in chat — Anthropic chat UI patterns, OpenAI assistants UI. Persistent agent state — conversation history, scratchpads, vector memory. Background async jobs — long-running agents with Celery, Inngest, or Temporal. Resumable conversations after disconnect. Multi-modal inputs — file uploads, images, voice (Whisper). React patterns for agent UX — loading skeletons, tool-call cards, edit-and-rerun, branching conversations.
M4	**Multi-Agent Orchestration** When to go multi-agent — task decomposition, expertise specialization, parallelism. Patterns — supervisor + workers (LangGraph), peer-to-peer round-robin (AutoGen), hierarchical teams (CrewAI), swarms (OpenAI Swarm). Inter-agent communication — message passing, shared scratchpad, blackboard architecture. Conflict resolution. Resource budgets per agent. Cost-latency trade-offs of multi-agent vs single-agent. Real-world examples — Devin (software engineering), Open Hands, manus.im, replit-agent. When NOT to go multi-agent (most of the time).
M5	**Agent Safety & Reliability** Threat model for agents — what's worse than a chatbot? Tool-mediated harm — destructive shell commands, data exfiltration via API tools, accidental email sends. Capability scoping — least-privilege tool access, allow/deny lists, dry-run mode. Sandboxing strategies — containers, Firecracker microVMs, e2b, Modal, Anthropic Code Execution. Prompt injection through tools — when an agent reads a webpage and follows malicious instructions. Audit logging for every tool call. Infinite-loop detection — max steps, no-progress detection, budget caps. Human-in-the-loop — confirmation gates for destructive ops, escalation patterns. Anthropic Constitutional AI principles for agents.
M6	**Agent Observability & Eval** What's hard about evaluating agents — partial credit, multi-step plans, non-deterministic paths. Trace-based observability — LangSmith, Braintrust, Langfuse for agent traces. Step-by-step replay. LLM-as-judge for reasoning chains. Agent benchmarks — AgentBench, WebArena, SWE-bench, OSWorld, HumanEval-Pack. Building custom eval suites — golden tasks, success criteria, partial-credit rubrics. CI for agents — automatic eval on every code change. A/B testing agent versions. Cost-quality trade-off plots. Going to production — feature flags, gradual rollout, kill switches.

Reference Books:

1. Building Generative AI Services with FastAPI Ali Parandeh · O'Reilly · 2025 ISBN: 978-1-098-16025-8

31/7/26
HoD-CSA
Dr. K
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

2. AI Engineering — Building Applications with Foundation Models Chip Huyen · O'Reilly · 2025 ISBN: 978-1-098-16630-4
3. Anthropic Documentation — Building Effective Agents Anthropic Engineering · 2024 anthropic.com/engineering/building-effective-agents
4. Multi-Agent Systems — A Modern Approach (3rd ed) Gerhard Weiss (ed.) · MIT Press · 2024 ISBN: 978-0-262-04895-8
5. Hands-On Generative AI with Transformers and Diffusion Models Omar Sanseviero, Pedro Cuenca et al. · O'Reilly · 2024 ISBN: 978-1-098-14924-6
6. The MCP Specification Anthropic + community · 2024+ modelcontextprotocol.io
7. The Alignment Problem Brian Christian · W.W. Norton · 2020 ISBN: 978-0-393-86833-3

Blockchain & Smart Contract Prog

COURSE CODE: 26CA6216E
COURSE NAME : Blockchain & Smart Contract Prog
L-T-P-S :3-0-3-0
PRE-REQ : FSD2

CO. No	Course Outcome Description	PO	BTL
CO1	Apply blockchain fundamentals — distributed ledgers, consensus mechanisms (PoW, PoS, PBFT), Merkle trees, hash functions, asymmetric cryptography & signatures — to reason about transactional integrity in decentralized systems.	PO1,PO4	3
CO2	Analyze the EVM execution model — gas, opcodes, storage layout, calldata, message calls, delegatecall — to write efficient and correct smart-contract bytecode.	PO2,PO3	4
CO3	Apply Solidity (and Vyper) language idioms — types, mappings, structs, modifiers, events, errors, libraries — to build smart contracts following community style guides.	PO1,PO4	3
CO4	Apply Hardhat / Foundry development frameworks for testing (forge test), gas profiling, fuzzing, and local-network deployment with Anvil.	PO1,PO4	3
CO5	Analyze ERC token standards (ERC-20, ERC-721, ERC-1155, ERC-4626) and DeFi primitives (AMM, lending, staking, yield) to compose production protocols.	PO2,PO4	4
CO6	Evaluate smart-contract security — reentrancy, integer over/underflow, oracle manipulation, signature replay, MEV, access-control flaws — and apply audit methodologies (Slither, Mythril, manual review, formal verification basics).	PO1,PO4,PO6	5

Syllabus:

31/7/26
Dr. [illegible] HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Module No.	Syllabus
M1	**Blockchain Foundations & Cryptography** Distributed ledger fundamentals — what blockchains actually solve. Hash functions (SHA-256, Keccak-256), pre-image & collision resistance. Merkle trees & Patricia tries (used in Ethereum state). Asymmetric cryptography — secp256k1 elliptic curve, public/private keypairs, ECDSA signatures, address derivation. Bitcoin UTXO model vs Ethereum account model. Consensus algorithms — Proof of Work, Proof of Stake (Ethereum 2.0), PBFT family, BFT vs CFT. The blockchain trilemma. Layer-1 vs Layer-2; rollups (optimistic vs zk).
M2	**Ethereum Virtual Machine (EVM)** EVM architecture — stack, memory, storage, calldata. Gas model — opcode costs, storage refund (post-EIP-3529), gas limits. Transaction lifecycle — mempool, block inclusion, finality. Calls — message calls, delegatecall (proxy patterns), staticcall, callcode. ABI encoding & calldata layout. Storage slot layout — packing, mappings, dynamic arrays. Memory expansion costs. Yul / inline assembly basics. Gas-optimization techniques — packed structs, unchecked math (post-0.8), bytes32 over string, efficient hashing.
M3	**Solidity Language & Idioms** Solidity 0.8.x — types (uint, int, address, bytes, string, fixed-size vs dynamic arrays). Mappings, structs, enums. Function visibility (public, external, internal, private). Modifiers (onlyOwner pattern). Custom errors (cheaper than require strings). Events & indexed parameters for off-chain consumption. Interfaces vs abstract contracts. Libraries — embedded vs deployed. Inheritance & the diamond problem (linearization). Solidity-style guide — OpenZeppelin Contracts conventions. Vyper as a safer alternative — when to consider it.
M4	**Tooling — Hardhat, Foundry, Anvil** Development environment — Foundry (forge, cast, anvil) deep dive. forge test for unit testing in Solidity (preferred over Hardhat's JS testing for new projects). cast for command-line interaction with deployed contracts. anvil for local fork-mainnet testing. Hardhat as the alternative — when teams already use it. Hardhat-Network for snapshots, time travel. Fuzz testing (forge fuzz). Invariant testing (forge invariant). Coverage with forge coverage. Gas snapshots. Deployment scripts — Foundry's forge script, Hardhat's deploy scripts. Etherscan verification.
M5	**ERC Standards & DeFi Primitives** ERC-20 fungible tokens — implementation walkthrough using OpenZeppelin. Common pitfalls (USDT non-standard returns, double-spend on approve). ERC-721 NFTs — metadata, base URIs, royalties (ERC-2981). ERC-1155 multi-token. ERC-4626 vaulted assets — the new yield-bearing standard. DeFi primitives: AMMs (Uniswap V2 vs V3), lending (Aave/Compound mechanics), staking, yield farming. Oracle integration — Chainlink price feeds, custom oracles. Cross-chain — bridges, LayerZero, CCIP basics.
M6	**Security & Auditing** SWC (Smart Contract Weakness Classification) registry top patterns — reentrancy (and its checks-effects-interactions defense), integer overflow (less of an issue post-0.8 but still in unchecked blocks), front-running & MEV, oracle manipulation, flash-loan attacks, signature replay (EIP-712 structured signing), unchecked external calls, access-control flaws. Tools — Slither (static analysis), Mythril

Dr. K. BHAGAVAN 31/01/26
HoD-CSA
HOD-Computer Science & Applications
Koneru Lakshmaiah Education Foundation
(A - Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	(symbolic execution), Echidna (fuzzing), Halmos (formal). Manual audit methodology — Cyfrin Updraft / Spearbit / Code4rena standards. Bug-bounty platforms — Immunefi. Real-world exploit case studies — DAO hack, Parity multi-sig, Beanstalk, Nomad, Wormhole, Euler.

Reference Books:

1. Mastering Ethereum — Building Smart Contracts and Dapps Andreas M. Antonopoulos, Gavin Wood · O'Reilly · 2018 ISBN: 978-1-491-97194-9
2. Mastering Bitcoin (3rd ed) Andreas M. Antonopoulos, David A. Harding · O'Reilly · 2023 ISBN: 978-1-098-15009-9
3. Programming Bitcoin Jimmy Song · O'Reilly · 2019 ISBN: 978-1-492-03149-9 ·
4. Solidity Programming Essentials (2nd ed) Ritesh Modi · Packt · 2022 ISBN: 978-1-80-323321-2
5. Hands-On Smart Contract Development with Solidity and Ethereum Kevin Solorio, Randall Kanna, David H. Hoover · O'Reilly · 2019 ISBN: 978-1-492-04526-7
6. Foundry Book Foundry community · ongoing book.getfoundry.sh · Free
7. Smart Contract Security Field Guide Cyfrin / OpenZeppelin / Trail of Bits collected best practices
   scsfg.io + consensys/smart-contract-best-practices · Free

Defensive Security

COURSE CODE: 26CA6217E
COURSE NAME : Defensive Security
L-T-P-S :3-0-3-0
PRE-REQ : OFSC

CO. No	Course Outcome Description	PO	BTL
CO1	Apply SOC operations frameworks (NIST CSF, MITRE ATT&CK, Cyber Kill Chain) to map attacker tactics to defensive layers and design tier-1/2/3 SOC workflows.	PO1,PO4	3
CO2	Apply log collection & aggregation pipelines (rsyslog, Filebeat, Fluentd, Vector) and SIEM platforms (Wazuh, Elastic Security, Splunk) to ingest, parse, and normalize security telemetry.	PO1,PO4	3
CO3	Analyze detection-engineering pipelines — Sigma rules, Splunk SPL, KQL (Kusto), Yara, OSQuery — to author high-fidelity, low-false-positive alerts mapped to MITRE ATT&CK techniques.	PO2,PO4	4
CO4	Analyze network and endpoint telemetry — Zeek, Suricata, Sysmon, OSQuery, EDR — to perform threat hunting using pyramid-of-pain indicators.	PO1,PO2	4
CO5	Evaluate incident response playbooks (SANS PICERL, NIST SP 800-61) and conduct hands-on digital forensics — disk, memory	PO2,PO4	5

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	(Volatility 3), network captures — to triage, contain, and learn from incidents.		
CO6	Evaluate threat-intelligence consumption (CTI feeds, MISP, OpenCTI, ATT&CK Navigator) and threat-modeling techniques (STRIDE, PASTA) to inform proactive defense priorities.	PO2,PO3,PO4	5

Syllabus:

Module No.	Syllabus
M1	**SOC Frameworks & Architecture** SOC mission & metrics — MTTD, MTTR, alert triage rate. NIST Cybersecurity Framework (Identify / Protect / Detect / Respond / Recover). MITRE ATT&CK matrix — tactics, techniques, sub-techniques; ATT&CK Navigator for coverage mapping. Lockheed Martin Cyber Kill Chain. SOC tiering — T1 alert triage, T2 investigation, T3 hunting / engineering, SOAR (security orchestration). Defense-in-depth layers. Modern SOC architecture — XDR vs SIEM+EDR+NDR. Purple-team operations.
M2	**Logging & SIEM Foundations** Log sources — Linux auditd, Windows Event Logs (Sysmon ATT&CK config), syslog, application logs, cloud audit (CloudTrail, Cloud Audit Logs). Collection — rsyslog forwarding, Filebeat, Fluentd, Vector. Parsing & enrichment — Logstash filters, Vector transforms, ECS schema. SIEM platforms — Wazuh (open-source, Elastic-backed), Elastic Security stack, Splunk (commercial). Index management, retention, hot/warm/cold tiers. Cost optimization — sampling, log levels, data tiering. Time-series analytics, dashboards.
M3	**Detection Engineering** Detection-engineering lifecycle — research → develop → validate → deploy → tune. Sigma — vendor-neutral detection rule format; sigma-cli for cross-SIEM compilation. Splunk SPL deep-dive — search, stats, transaction, eventstats, lookup. KQL (Kusto) for Microsoft Sentinel + Defender. Yara rules for file content; YaraGen for sample-derived rules. OSQuery as SQL-over-host-state. False-positive reduction — base-rate fallacy, exception lists, baselines. ATT&CK technique coverage gap analysis. ML/UEBA basics for anomaly-based detection.
M4	**Network & Endpoint Defense** Network monitoring — Zeek (formerly Bro) for protocol-level metadata, Suricata for IDS/IPS, NetFlow/IPFIX for flow analytics. PCAP analysis with Wireshark + tshark. TLS inspection trade-offs. Endpoint — Sysmon configuration (SwiftOnSecurity / olafhartong baselines), OSQuery scheduled packs, EDR fundamentals (CrowdStrike, SentinelOne, Defender for Endpoint). Threat hunting — pyramid of pain (hash → IP → domain → artifact → tool → TTP). Hunting hypotheses; ATT&CK-driven hunts. Living-off-the-land binaries (LOLBAS / GTFOBins).
M5	**Incident Response & Forensics** SANS PICERL — Preparation, Identification, Containment, Eradication, Recovery, Lessons-Learned. NIST SP 800-61 incident-response lifecycle. Playbook authoring — runbook structure, decision trees, escalation paths. Containment trade-offs —

Dr. K. BHASKAN 31/7/26
HoD-CSA
HOD-Computer Science & Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	disconnect vs monitor, isolation vs business continuity. Digital forensics — chain of custody, image acquisition (dd, ddrescue, FTK Imager). Memory forensics with Volatility 3 — common plugins (pslist, malfind, netscan). Disk forensics — Autopsy, registry analysis. Network forensics — full-PCAP review. Post-incident — root-cause analysis, after-action reports.
M6	**Threat Intelligence & Threat Modeling** CTI lifecycle — collection, processing, analysis, dissemination, feedback. Sources — open feeds (AlienVault OTX, Abuse.ch), ISACs, paid feeds (CrowdStrike Falcon Intelligence, Recorded Future). MISP & OpenCTI for organization-wide CTI sharing. STIX 2.1 / TAXII protocols. Threat actor profiles — APTs, FIN groups, ransomware syndicates. Threat modeling — STRIDE (Microsoft), PASTA (Process for Attack Simulation & Threat Analysis), DREAD scoring. Attack-tree analysis. Translating threat intel into detection rules. Reporting to leadership — executive briefs, exposure metrics.

Reference Books:

1. The Practice of Network Security Monitoring Richard Bejtlich · No Starch Press · 2013 ISBN: 978-1-59327-509-9
2. Blue Team Handbook — Incident Response Edition Don Murdoch · CreateSpace · 2018 ISBN: 978-1-5005-7449-3
3. The Art of Memory Forensics Michael Hale Ligh, Andrew Case et al. · Wiley · 2014 ISBN: 978-1-118-82509-9
4. Threat Modeling — Designing for Security Adam Shostack · Wiley · 2014 ISBN: 978-1-118-80999-0
5. Practical Threat Intelligence and Data-Driven Threat Hunting Valentina Costa-Gazcón · Packt · 2021 ISBN: 978-1-83855-636-9
6. MITRE ATT&CK Design and Philosophy MITRE Corporation attack.mitre.org · Free

Web3 & dApp Development

COURSE CODE: 26CA6318E
COURSE NAME : Web3 & dApp Development
L-T-P-S :3-0-3-0
PRE-REQ : BCSC

3/1/26
HoD-CSA
Dr. K. GAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO. No	Course Outcome Description	PO	BTL
CO1	Apply Web3 frontend stack — ethers.js v6 / viem, wagmi hooks, RainbowKit / ConnectKit / Privy / WalletConnect — to build wallet-connected React dApps.	PO1,PO2	3
CO2	Apply on-chain reads & writes — provider, signer, contract instances, transaction tracking, gas estimation, ENS resolution — with proper UX feedback.	PO1,PO2	3
CO3	Analyze indexing patterns — The Graph subgraphs, GraphQL queries, alternatives (Goldsky, Alchemy Subgraphs, ponder.sh) — to design queryable on-chain data layers.	PO2,PO4	4
CO4	Analyze decentralized storage (IPFS, Arweave, Filecoin) and metadata standards (ERC-721 metadata, OpenSea standards) for NFT & content dApps.	PO2,PO4	4
CO5	Apply Layer-2 deployment patterns (Base, Arbitrum, Optimism, Polygon zkEVM) and cross-chain primitives (LayerZero, Wormhole, CCIP) for multi-chain dApps.	PO1,PO4	3
CO6	Evaluate account abstraction (ERC-4337) and gasless / sponsored transactions to design dApps with mainstream UX.	PO2,PO3,PO4	5

Syllabus:

Module No.	Syllabus
M1	**Web3 Frontend Stack** Modern Ethereum frontend libraries — ethers.js v6 vs viem (the new standard). wagmi hooks (useAccount, useConnect, useReadContract, useWriteContract). Wallet connection — Injected (MetaMask), WalletConnect v2, Coinbase Wallet, embedded wallets (Privy, Magic.link). Wallet-connection UIs — RainbowKit, ConnectKit, ThirdWeb. Multi-chain support. Network switching UX. Provider patterns — public RPCs (free tier) vs Alchemy / Infura / QuickNode. Type-safe contract hooks via wagmi-cli & ABIType.
M2	**On-Chain Interactions** Provider vs signer. Contract instances; ABI consumption; type-safe contract calls. Reading contract state — single calls, batched multicall, view functions. Writing transactions — gas estimation (estimateGas, max-fee/priority-fee modeling), transaction tracking (hash → receipt → confirmations), error handling (revert reasons, custom errors). ENS resolution — name → address & reverse. Event subscription — getLogs, contract event filters, real-time updates via WebSocket. EIP-712 typed signatures for off-chain auth (sign-in-with-Ethereum / SIWE).
M3	**Indexing & Subgraphs** Why subgraphs — RPC limitations, scalability of historical data. The Graph architecture — graph node, IPFS, indexers, queriers. Subgraph anatomy — manifest, schema (GraphQL), mappings (AssemblyScript). Authoring a subgraph for an ERC-721 marketplace. Hosted Service vs Decentralized Network. Querying via Apollo Client / urql / GraphQL Codegen. Alternatives — Goldsky (managed),

31/1/26
HoD-CSA

Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur Dist., Andhra Pradesh-522502.

	ponder.sh (TypeScript-native), Alchemy Subgraphs, custom indexers with Foundry's cast logs. Choosing among them.
M4	**Decentralized Storage & NFTs** IPFS — content addressing (CIDv0/v1), pinning services (Pinata, web3.storage, NFT.Storage). Arweave — permanent storage, bundlr/ardrive. Filecoin overview. NFT metadata standards — ERC-721 metadata JSON, OpenSea metadata extensions, ERC-2981 royalties. Metadata patterns — fully on-chain (SVG, base64), IPFS-pinned, centralized (anti-pattern). Reveal mechanics. Marketplace integration — OpenSea, Blur, Magic Eden APIs. Soulbound tokens (ERC-5114). On-chain art generative patterns.
M5	**L2s & Cross-Chain** Layer-2 landscape — Optimistic Rollups (Optimism, Arbitrum, Base) vs zk-Rollups (Polygon zkEVM, zkSync Era, Linea, Scroll, Starknet). Bridges — native L1↔L2 vs third-party (Across, Hop). Deployment to multiple L2s. Gas-token differences. RPC tooling for multi-chain (viem chains, wagmi config). Cross-chain messaging — LayerZero (omnichain apps), Wormhole, Chainlink CCIP. Cross-chain UX patterns — chain selector, source-chain awareness, atomic vs eventual cross-chain ops.
M6	**Account Abstraction & Gasless UX** Why account abstraction matters — onboarding cliff, gas-token requirement, key management. ERC-4337 architecture — UserOperations, Bundler, EntryPoint, Paymaster, Smart Account. Smart-account implementations — Safe (Gnosis), Kernel (ZeroDev), Biconomy, Coinbase Smart Wallet. Paymasters for sponsored transactions. Session keys for one-time-permission flows. Social recovery. Embedded wallets (Privy / Magic / Web3Auth) — pros/cons. Combining EOAs & smart accounts. ERC-7579 modular accounts.

Reference Books:

1. Mastering Ethereum (Antonopoulos & Wood) O'Reilly · 2018 ISBN: 978-1-491-97194-9
2. Building Ethereum Dapps Roberto Infante · Manning · 2019 ISBN: 978-1-617-29516-4
3. Hands-On Smart Contract Development with Solidity and Ethereum Solorio, Kanna, Hoover · O'Reilly · 2019 ISBN: 978-1-492-04526-7
4. The Graph Documentation thegraph.com/docs Free
5. viem & wagmi Documentation viem.sh · wagmi.sh Free
6. ERC-4337 Account Abstraction Documentation eip.tools/eip/4337 · ZeroDev / Biconomy / Stackup docs Free

DevSecOps

HoD-CSA 3/7/26
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

COURSE CODE: 26CA6319E
COURSE NAME : DevSecOps
L-T-P-S :3-0-3-0
PRE-REQ : DFSC

CO. No	Course Outcome Description	PO	BTL
CO1	Apply DevSecOps principles — shift-left, security-as-code, threat-modeling at design — to embed security gates throughout the SDLC.	PO1,PO3	3
CO2	Analyze application-security tooling — SAST (Semgrep, Snyk Code, GitHub CodeQL), DAST (OWASP ZAP, Burp Pro), SCA (Snyk, Dependabot), secrets scanning (Gitleaks, TruffleHog) — and integrate them into CI/CD pipelines with proper severity gating.	PO2,PO4	4
CO3	Analyze container & supply-chain security — image scanning (Trivy, Grype), SBOM generation (Syft, CycloneDX, SPDX), provenance attestations (SLSA, in-toto, Sigstore cosign).	PO2,PO4	4
CO4	Apply Infrastructure-as-Code security — Terraform Cloud / OpenTofu policies (OPA / Rego, Sentinel), Checkov, tfsec, Terrascan — to prevent misconfigurations.	PO1,PO4	3
CO5	Evaluate Kubernetes & runtime security — Pod Security Standards, network policies, OPA Gatekeeper / Kyverno, Falco runtime detection, eBPF-based observability (Cilium, Tetragon).	PO2,PO6	5
CO6	Evaluate cloud security posture management (CSPM) — AWS Security Hub, GuardDuty, Inspector, Prowler, ScoutSuite — and design cloud-native defense-in-depth aligned with CIS Benchmarks & Well-Architected pillars.	PO2,PO3,PO6	5

Syllabus:

Module No.	Syllabus
M1	**DevSecOps Foundations** DevSecOps philosophy — shift-left vs shift-right; security as a constraint vs an enabler. SDLC stages (plan, code, build, test, release, deploy, operate, monitor) — security activities at each. Threat modeling at design time — STRIDE, PASTA, attack-tree decomposition. Risk-rated security controls — what to enforce vs flag vs report. Security champions program. The DevSecOps maturity model. Anti-patterns — security theater, alert fatigue, blocking-everything cultures.
M2	**Application Security in CI/CD** SAST — Semgrep (open-source rules), Snyk Code, GitHub CodeQL. Custom Semgrep rules for project-specific patterns. DAST — OWASP ZAP (baseline vs full active), Burp Suite Pro CI integration. SCA & dependency scanning — Snyk, Dependabot, OSV-Scanner; vulnerable-dependency alert workflows. Secrets scanning — Gitleaks (pre-commit + CI), TruffleHog, GitHub Advanced Security secret scanning. License compliance scanning. CI gating strategy — fail-on-critical, warn-on-high, ignore-on-medium with SLA-driven remediation.

Dr. K. BHARGAVI
HOD-Computer Science & Engineering
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.
HoD-CSA
3/4/26

M3	**Container & Supply-Chain Security** Container image scanning — Trivy, Grype, Snyk Container — for OS packages + application dependencies. Distroless / minimal base images for shrunken attack surface. Multi-stage builds for safer images. SBOM generation — Syft, CycloneDX, SPDX standards; SBOM consumption for vulnerability tracking. Software supply-chain attacks — SolarWinds, Codecov, log4shell as case studies. SLSA framework levels 1-4. Provenance attestations with in-toto + Sigstore cosign. Signing & verifying images. Reproducible builds.
M4	**Infrastructure-as-Code Security** IaC misconfigurations as the leading cloud-breach root cause. Static scanners — Checkov, tfsec, Terrascan, KICS. Policy-as-code — Open Policy Agent (Rego language), HashiCorp Sentinel, AWS Service Control Policies. Pre-merge IaC validation in CI. Drift detection. Common IaC anti-patterns — public S3 buckets, security groups 0.0.0.0/0, unencrypted EBS, IAM wildcards. CIS Benchmark compliance via tooling. AWS Well-Architected Tool integration.
M5	**Kubernetes & Runtime Security** Pod Security Standards (Privileged / Baseline / Restricted) — replacing PodSecurityPolicy. Network Policies — default deny + allow-listing. Service mesh security (Istio mTLS). Admission controllers — OPA Gatekeeper, Kyverno. Runtime detection — Falco rules, eBPF-based tooling (Cilium Tetragon, Pixie). RBAC hardening. Secrets management — External Secrets Operator, Sealed Secrets, HashiCorp Vault on K8s. CIS Kubernetes Benchmark via kube-bench.
M6	**Cloud Security Posture Management** CSPM landscape — AWS Security Hub + GuardDuty + Inspector; Azure Defender; GCP Security Command Center. Open-source — Prowler, ScoutSuite, CloudSploit. CIS Benchmark mapping. Compliance frameworks — PCI-DSS, HIPAA, ISO 27001, SOC 2, India's DPDP. Continuous compliance reporting. Incident response in cloud — log centralization, response playbooks. Cloud-Native Application Protection Platforms (CNAPP) — Wiz, Orca, Palo Alto Prisma — what they unify and why orgs adopt them.

Reference Books:

1. DevOpsSec — Securing Software Through Continuous Delivery Jim Bird · O'Reilly · 2016 ISBN: 978-1-491-97155-0
2. Securing DevOps Julien Vehent · Manning · 2018 ISBN: 978-1-617-29429-7
3. Software Supply Chain Security Cassie Crossley · O'Reilly · 2024 ISBN: 978-1-098-13371-3
4. Container Security Liz Rice · O'Reilly · 2020 ISBN: 978-1-492-05670-6
5. Kubernetes Security and Observability Brendan Creane, Amit Gupta · O'Reilly · 2021 ISBN: 978-1-098-10708-6
6. Hands-On Security in DevOps Tony Hsu · Packt · 2018 ISBN: 978-1-78899-550-4
7. Cloud Security Handbook Eyal Estrin · Packt · 2022 ISBN: 978-1-80056-919-9

3/7/24
HoD-CSA
Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Data Science for Scale

COURSE CODE: 26CA6320E
COURSE NAME : Data Science for Scale
L-T-P-S :3-0-3-0
PRE-REQ : DSAV

CO. No	Course Outcome Description	PO	BTL
CO1	Apply distributed data-processing fundamentals — partitioning, shuffling, MapReduce semantics, joins at scale, fault-tolerance — to reason about large-data workloads.	PO1,PO2	3
CO2	Apply Apache Spark (DataFrame API + Spark SQL + PySpark) for batch ETL on multi-GB / multi-TB datasets, with proper partitioning, caching, and join strategies.	PO1,PO4	3
CO3	Analyze streaming patterns — Kafka topics, partitions, consumer groups, exactly-once semantics, watermarks, windowing — to design real-time data pipelines.	PO2,PO3	4
CO4	Analyze stream-processing engines — Flink (event-time, stateful), Spark Structured Streaming, Kafka Streams — and choose the right tool per workload.	PO2,PO4	4
CO5	Apply modern data-stack engineering — orchestration (Airflow, Dagster, Prefect), transformation (dbt), lakehouse formats (Delta, Iceberg, Hudi), data warehouse loading patterns.	PO1,PO4	3
CO6	Evaluate data-quality, observability, lineage, and cost optimization — Great Expectations, Monte Carlo / DataHub / OpenLineage, query-cost analysis — for production-grade pipelines.	PO2,PO5,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Distributed Data Foundations** From single-node to distributed — why scaling matters. CAP / PACELC revisited for data systems. MapReduce semantics. Data partitioning strategies — hash, range, round-robin. Shuffle as the most expensive operation. Co-located joins vs broadcast joins. Fault-tolerance — checkpointing, lineage, write-ahead logs. Storage formats — row-oriented (CSV, JSON) vs columnar (Parquet, ORC, Avro). Data layout impact on query cost. Object stores (S3, GCS, ADLS) as the modern data fabric.

Dr. K. BHAGAVAN 3/3/26
HoD-CSA
HOD-Computer Science & Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522

M2	**Apache Spark for Batch** Spark architecture — driver, executors, DAG, Catalyst optimizer, Tungsten engine. RDD vs DataFrame vs Dataset. PySpark DataFrame API — transformations vs actions, lazy evaluation. Spark SQL — registering tables, query optimization. Partitioning & bucketing for read/write performance. Cache vs persist. Broadcast joins for small tables. Skewed joins — salting. Reading/writing Parquet, JSON, CSV at scale. Spark on Databricks vs EMR vs self-hosted. Cluster sizing & cost optimization.
M3	**Streaming & Kafka** Stream-processing semantics — at-most-once, at-least-once, exactly-once. Event time vs processing time. Watermarks for late data. Windowing — tumbling, sliding, session windows. Kafka architecture — brokers, topics, partitions, replication, consumer groups, offsets. Kafka Connect for source/sink integration. Schema Registry & Avro. Producer & consumer tuning. Idempotent producers. Transactions across topics. Real-world Kafka topology — multi-region, mirror-maker, MSK / Confluent Cloud.
M4	**Stream-Processing Engines** Apache Flink architecture — JobManager, TaskManagers, state backends (RocksDB), checkpointing, savepoints. Flink DataStream API — keyed streams, ProcessFunctions, CEP for complex events. Flink SQL — table API. Spark Structured Streaming as alternative — micro-batch vs continuous mode, Trigger.AvailableNow. Kafka Streams for embedded stream processing. Choosing the right engine — Flink for true streaming & CEP, Spark when team already runs Spark, Kafka Streams for in-Kafka topology.
M5	**Modern Data Stack & Lakehouse** Lakehouse formats — Delta Lake (Databricks), Apache Iceberg (Snowflake/AWS), Apache Hudi (Uber/AWS). ACID transactions on object storage. Time travel & schema evolution. Modern data stack components — extract (Airbyte, Fivetran, Meltano), load (Snowflake, BigQuery, Redshift, Databricks SQL Warehouse), transform (dbt). Orchestration — Airflow vs Dagster vs Prefect. Reverse ETL — Hightouch, Census. Data warehousing concepts — star vs snowflake schema, slowly changing dimensions, fact vs dimension tables. Cost optimization for warehouses (clustering, partitioning, query optimization).
M6	**Data Quality, Observability & Lineage** Data quality dimensions — accuracy, completeness, consistency, timeliness, uniqueness, validity. Great Expectations / Soda for assertions. dbt tests — not_null, unique, accepted_values, custom. Data observability platforms — Monte Carlo, Bigeye, DataDog Data Streams. Lineage tracking — OpenLineage, DataHub, Marquez, Atlan. Anomaly detection on metrics. SLA / SLO for data pipelines. Cost observability — query cost dashboards, identifying expensive queries. Production runbooks for data engineers.

Reference Books:

1. Designing Data-Intensive Applications Martin Kleppmann · O'Reilly · 2017 ISBN: 978-1-449-37332-0

3/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

2. Learning Spark (2nd ed) Jules S. Damji et al. · O'Reilly · 2020 ISBN: 978-1-492-05004-9
3. Spark: The Definitive Guide Bill Chambers, Matei Zaharia · O'Reilly · 2018 ISBN: 978-1-491-91221-8
4. Kafka — The Definitive Guide (2nd ed) Gwen Shapira, Todd Palino, Rajini Sivaram, Krit Petty · O'Reilly · 2021 ISBN: 978-1-492-04308-7
5. Streaming Systems Tyler Akidau, Slava Chernyak, Reuven Lax · O'Reilly · 2018 ISBN: 978-1-491-98364-4
6. Fundamentals of Data Engineering Joe Reis, Matt Housley · O'Reilly · 2022 ISBN: 978-1-098-10830-4
7. Delta Lake — The Definitive Guide Vini Jaiswal, Denny Lee, Tristen Wentling, Prashanth Babu V. · O'Reilly · 2024 ISBN: 978-1-098-15191-1

Conversational AI & Speech Systems

COURSE CODE: 26CA61A1E
COURSE NAME : Conversational AI & Speech Systems
L-T-P-S :3-0-3-0
PRE-REQ : AIML

CO. No	Course Outcome Description	PO	BTL
CO1	Apply automatic speech recognition pipelines (Whisper, Deepgram, Google STT) including streaming ASR, language identification, code-switching, and on-device alternatives (whisper.cpp, faster-whisper) for low-latency voice apps.	PO1,PO4	3
CO2	Apply text-to-speech (TTS) systems — commercial (ElevenLabs, OpenAI TTS, Cartesia) and open-source (Coqui XTTS, Piper) — with voice-cloning, SSML markup, and real-time streaming for natural voice output.	PO1,PO4	3
CO3	Analyze dialog architectures — intent-based (Rasa, Dialogflow), retrieval-based, and LLM-based (Claude, GPT-4o) — to design state machines, slot filling, multi-turn flows, and graceful fallbacks.	PO2,PO4	4
CO4	Analyze voice-stack engineering — turn-taking, interruption handling, voice-activity detection (VAD), endpoint detection, latency budgets, jitter buffers — for telephony & real-time voice UX.	PO2,PO3	4
CO5	Evaluate conversational quality — ASR WER, TTS MOS, dialog success rate, completion rate, sentiment, escalation rate — using golden conversation sets, simulator-driven tests, and LLM-as-judge.	PO2,PO6	5
CO6	Evaluate Indian-context challenges — multilingual + code-switched speech (Hindi-English, Telugu-English, Tamil-English), low-resource voice cloning, accents, and IndicNLP / AI4Bharat models.	PO2,PO3,PO4	5

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Syllabus:

Module No.	Syllabus
M1	**ASR & Speech Foundations** Speech-signal basics — sampling, FFT, MFCCs, mel-spectrograms. Classical ASR briefly (HMM-GMM, then DNN-HMM). Modern ASR architectures — CTC, attention-based encoders (Listen-Attend-Spell), Whisper (Transformer encoder-decoder). Streaming vs offline ASR. Beam-search decoding, language-model rescoring. Word Error Rate (WER) computation. Evaluation datasets — LibriSpeech, CommonVoice, IndicSUPERB. Practical providers — Whisper API, faster-whisper, Deepgram, Google STT. Code-switching handling for Indian languages. AI4Bharat IndicConformer.
M2	**TTS & Voice Synthesis** Concatenative vs parametric vs neural TTS. Tacotron 2, FastSpeech, VITS. Neural codecs (Encodec, SoundStream). Mel-vocoders (HiFi-GAN, WaveGlow). Modern systems — ElevenLabs, OpenAI TTS, Cartesia Sonic, Coqui XTTS, Piper. SSML markup — prosody, pauses, emphasis. Voice cloning ethics & consent. Real-time streaming TTS — chunk boundaries, latency. Mean Opinion Score (MOS) for quality eval. Indian-language TTS with IndicTTS / Indri.
M3	**Dialog Architectures** Three architectures: rule-based (Rasa, Dialogflow CX), retrieval-based, LLM-orchestrated. Intent classification + entity extraction. Slot filling, slot validation, slot reprompting. Dialog state tracking. Multi-turn flows. Form-filling vs free-form. LLM-driven dialog — system prompt design for personas, tool-calling for actions, function schemas. Hybrid stacks (LLM + structured slot filler). Fallback strategies — clarify, escalate, hand-off to human. Conversation memory.
M4	**Voice-Stack Engineering** End-to-end voice latency budget (ASR + dialog + TTS < 2 seconds end-to-end target). Turn-taking and interruption handling. Voice Activity Detection (VAD) — Silero, WebRTC VAD. Endpointing — when to stop listening. Echo cancellation, noise suppression (RNNoise, Krisp). Telephony stack — Twilio, Vonage, Plivo. SIP / WebRTC fundamentals. LiveKit, Daily.co for browser-based voice. Pipecat / livekit-agents frameworks. Jitter buffers, packet loss, codec choice (Opus). Real-time conversation orchestration with low-latency LLM streams.
M5	**Conversational Quality & Eval** Quality metrics — WER for ASR, MOS for TTS, task-completion rate, dialog success rate, average turns to resolution, sentiment trajectory, escalation rate. Conversation simulators — user-bot vs bot-bot. Golden conversation set construction. LLM-as-judge for subjective eval. Latency profiling per turn. A/B testing voice prompts. Production observability — per-conversation traces, voice-recording sampling for review (with consent). Failure-mode analysis.
M6	**Indian-Context Multilingual Voice** Indian linguistic diversity — 22 scheduled languages, code-switching as the norm. Hindi-English, Tamil-English, Telugu-English mixed-language ASR. AI4Bharat models — IndicConformer ASR, IndicTrans2 NMT, Indri TTS. Bhashini ecosystem. Low-resource voice cloning for regional languages. Accent robustness. Cultural pragmatics — honorifics, formal/informal modes. Domain adaptation for

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	Indian use cases (banking IVR, telemedicine, agri-advisory). Compliance — consent for voice recording under DPDP Act 2023.

Reference Books:

1. Speech and Language Processing (3rd ed draft) Daniel Jurafsky, James H. Martin · web.stanford.edu/~jurafsky/slp3 · 2024 Free
2. Voice Applications for Alexa and Google Assistant Dustin A. Coates · Manning · 2019 ISBN: 978-1-617-29581-2
3. Conversational AI with Rasa Xiaoquan Kong, Guan Wang · Packt · 2021 ISBN: 978-1-80107-823-2
4. Hands-On Music Generation with Magenta + Audio Signal Processing for Music Applications Magenta team / Bello, Pizzi · O'Reilly + online
5. Designing Voice User Interfaces Cathy Pearl · O'Reilly · 2017 ISBN: 978-1-491-95541-3
6. AI4Bharat & Bhashini Documentation ai4bharat.iitm.ac.in · bhashini.gov.in Free online

Microservices, Service Mesh & Distributed Systems

COURSE CODE: 26CA61B1E
COURSE NAME : Microservices, Service Mesh & Distributed Systems
L-T-P-S :3-0-3-0
PRE-REQ : CDVO

CO. No	Course Outcome Description	PO	BTL
CO1	Analyze microservices design patterns — service decomposition (DDD), API contracts, backwards compatibility, anti-corruption layers, strangler-fig migrations — and apply them to real system designs.	PO2,PO3	4
CO2	Apply service mesh technology (Istio, Linkerd, Cilium) — sidecar proxies, mTLS everywhere, traffic policy, retry budgets, circuit breakers, canary releases — on Kubernetes.	PO1,PO4	3
CO3	Evaluate distributed-systems primitives — consensus (Raft, Paxos), leader election, distributed locks, vector clocks, gossip protocols — for correctness and performance trade-offs in real designs.	PO2,PO4	5
CO4	Analyze distributed transaction patterns — 2PC, Saga (choreography vs orchestration), outbox pattern, compensating transactions, eventual consistency — for cross-service data integrity.	PO2,PO4	4

HoD-CSA 31/7/26
Dr. K. BHARGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,

CO5	Apply API design with gRPC + Protocol Buffers, GraphQL Federation (Apollo), and event-driven async with NATS/Kafka topics for inter-service communication.	PO1,PO4	3
CO6	Evaluate failure-mode resilience — chaos engineering (Chaos Mesh, Litmus), fault injection, timeout cascades, retry storms, thundering herd — using game-day exercises and post-mortems.	PO2,PO3,PO4	5

Syllabus:

Module No.	Syllabus
M1	**Microservices Design Patterns** Monolith vs microservices — when each wins. Domain-Driven Design (DDD) — bounded contexts, ubiquitous language, aggregates. Service decomposition strategies. API contract-first design (OpenAPI, Buf, Spectral linting). Backwards compatibility — semantic versioning, deprecation policies. Anti-corruption layer pattern. Strangler-fig migration pattern. Database-per-service vs shared. Service-discovery mechanisms (DNS-based, Consul, Eureka). Observability fundamentals (logs, metrics, traces). The 12-Factor App principles for microservices.
M2	**Service Mesh on Kubernetes** Service mesh value proposition. Sidecar pattern (Envoy proxy). Istio architecture — istiod, ingress gateway, sidecar. Linkerd as lightweight alternative. Cilium / eBPF-based mesh. mTLS everywhere — certificate rotation, identity. Traffic management — VirtualService, DestinationRule, traffic shifting (10/90, canary). Resilience features — retry budgets, timeouts, circuit breakers, outlier detection. Authorization policies (deny-by-default network). Observability stack — Kiali, Jaeger, Grafana on top of mesh.
M3	**Distributed Systems Primitives** Why distributed systems are hard — partial failure, asynchronous networks, clock skew, FLP impossibility. CAP theorem revisited — PACELC. Consensus algorithms — Paxos basics, Raft (the practical choice). Leader election with etcd/ZooKeeper. Distributed locks — Redis Redlock vs etcd-based, the Redlock controversy. Vector clocks and Lamport clocks for causal ordering. Gossip protocols — Cassandra, Consul. Failure detection — phi accrual, SWIM. Read-your-writes, monotonic reads, session consistency.
M4	**Distributed Transactions** Why ACID is hard across services. Two-phase commit (2PC) — how it works, why it's avoided. Saga pattern — choreography (event-driven) vs orchestration (Temporal/Camunda). Compensating transactions. Idempotency keys for safe retries. Outbox pattern (transactional outbox). CDC (Change Data Capture) with Debezium. Event sourcing as a foundation. CQRS pairing with sagas. Eventual consistency reasoning — what consistency you can promise users vs not.
M5	**Inter-Service Communication** Synchronous vs asynchronous trade-offs. gRPC + Protocol Buffers — service definitions, streaming, deadlines, interceptors. GraphQL Federation (Apollo) — one client-facing schema spanning multiple services. REST best practices for backend-for-frontend (BFF). Async messaging — NATS (lightweight), Kafka (durable,

3/1/26
HoD-CSA
Dr.
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	large-scale), RabbitMQ (classical AMQP). Topic vs queue semantics. Schema registry (Confluent, Apicurio). Backpressure handling. Connection pooling.
M6	**Failure-Mode Resilience & Chaos** Failure modes in distributed systems — cascading failures, retry storms, thundering herd, backpressure failure, slow neighbor. Bulkheads, circuit breakers (Hystrix-style, modern resilience4j patterns). Timeout discipline — deadline propagation. Backoff and jitter on retries. Token bucket / leaky bucket rate limiting. Chaos engineering principles (Netflix's Principles of Chaos Engineering). Tools — Chaos Mesh, Litmus, Chaos Monkey. Game days — structured failure injection exercises. Post-mortem culture — blameless format.

Reference Books:

1. Designing Data-Intensive Applications Martin Kleppmann · O'Reilly · 2017 ISBN: 978-1-449-37332-0
2. Microservices Patterns Chris Richardson · Manning · 2018 ISBN: 978-1-617-29454-9
3. Building Microservices (2nd ed) Sam Newman · O'Reilly · 2021 ISBN: 978-1-492-03402-5
4. Istio in Action Christian E. Posta, Rinor Maloku · Manning · 2022 ISBN: 978-1-617-29582-9
5. Domain-Driven Design Eric Evans · Addison-Wesley · 2003 ISBN: 978-0-321-12521-7
6. Database Internals Alex Petrov · O'Reilly · 2019 ISBN: 978-1-492-04034-7
7. Chaos Engineering — System Resiliency in Practice Casey Rosenthal, Nora Jones · O'Reilly · 2020 ISBN: 978-1-492-04386-7

Advanced Threat Hunting & Adversary Emulation

COURSE CODE: 26CA61C1E
COURSE NAME : Advanced Threat Hunting & Adversary Emulation
L-T-P-S :3-0-3-0
PRE-REQ : OFSC

CO. No	Course Outcome Description	PO	BTL

Dr. K. [illegible]AVAN
HoD-CSA
3/1/26
Koneru Lakshmaiah Education Foun
(K.L. Deemed to be Universit
VADDESWARA
Andhra Pradesh-5

CO1	Analyze the cyber kill chain (Lockheed Martin) and MITRE ATT&CK framework — tactics, techniques, sub-techniques, procedures — and map adversary behaviors to specific TTPs.	PO2,PO4	4
CO2	Apply offensive-security tooling in authorized lab environments — Kali Linux, Caldera, Atomic Red Team, Sliver, Mythic, Metasploit — for adversary emulation exercises.	PO1,PO4	3
CO3	Analyze threat-hunting hypotheses, KQL/SPL/Sigma queries, and SIEM log analysis (Splunk, Elastic, Microsoft Sentinel, Wazuh) for proactive detection.	PO2,PO4	4
CO4	Apply detection engineering — writing Sigma rules, EQL queries, KQL analytics, mapping detections to ATT&CK techniques, false-positive triage.	PO1,PO2	3
CO5	Analyze threat intelligence sources — STIX/TAXII feeds, IOCs, MISP, OpenCTI — and incorporate them into hunt + detection workflows.	PO1,PO2	4
CO6	Evaluate purple-team exercises — coordinated red-blue collaboration, ATT&CK Navigator coverage maps, hunt reports, after-action reviews — for defensive maturity improvement.	PO2,PO5,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Threat Models & ATT&CK** Threat-actor taxonomies (Mandiant APT groups, NCSC categories, India-CERT advisories). Lockheed kill chain (recon → weaponize → deliver → exploit → install → C2 → actions on objectives). MITRE ATT&CK Enterprise — 14 tactics, 200+ techniques, sub-techniques, procedures. ATT&CK for Cloud, Containers, ICS, Mobile. ATT&CK Navigator for layered visualization. Threat modeling methodologies — STRIDE, PASTA, DREAD. Threat profiles for Indian-context organizations (BFSI, PSU, telecom, healthcare).
M2	**Adversary Emulation** Adversary-emulation methodology (CALDERA, MITRE Engenuity). Atomic Red Team — running atomic tests mapped to ATT&CK. Caldera framework architecture. Sliver C2 (modern, open source). Mythic C2 framework. Metasploit responsibly. Cobalt Strike concepts (without licensing). Initial access vectors (phishing, exploitation, valid accounts) — legal boundaries. Lateral movement (Pass-the-Hash, Kerberos, RDP). Persistence techniques. Living-off-the-land binaries (LOLBins). Privilege escalation. Note: All exercises run in isolated lab/range only; no real-world operations.
M3	**SIEM & Threat Hunting** SIEM landscape — Splunk (SPL), Elastic Stack (KQL/EQL), Microsoft Sentinel (KQL), Wazuh (open-source), Chronicle. Log sources — Sysmon for Windows, auditd for Linux, EDR telemetry, network flow (Zeek), DNS logs, proxy logs, cloud logs (CloudTrail, GCP audit). Threat-hunting hypothesis generation. PEAK framework. The Pyramid of Pain (Bianco). Hunt cycles — trigger, scope, plan,

31/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	execute, communicate. Anomaly-based vs IOC-based vs TTP-based hunts. Search-language fluency (SPL, KQL, EQL, Yara basics).
M4	**Detection Engineering** Detection engineering as a discipline (Florian Roth, Anton Chuvakin). Sigma rule format — condition, detection, falsepositives. Sigma converters (sigmac, pySigma) to vendor-specific languages. Alyssa Miller's detection-as-code. CI/CD for detections — Detection-as-Code repos, DTS (Detection Testing Standard). False-positive engineering — the most-overlooked skill. Detection coverage maps via ATT&CK Navigator. ML-based detection (UEBA, supervised classification). Detection tuning loop. KQL analytics rules in Sentinel. Elastic Detection Engine.
M5	**Threat Intelligence** Threat intelligence categories — strategic, operational, tactical, technical. Diamond Model of intrusion analysis. STIX 2.x & TAXII for sharing. MISP open-source platform. OpenCTI as graph-based threat intel platform. Commercial sources (Mandiant Advantage, Recorded Future, Crowdstrike Falcon Intelligence). Free sources — AlienVault OTX, abuse.ch (URLhaus, ThreatFox, MalwareBazaar), CIRCL OSINT, India-CERT advisories. Indicators of Compromise (IOCs) hierarchy (hashes → IPs → domains → TTPs). Yara rule writing for malware-family hunting. Pivoting on adversary infrastructure.
M6	**Purple Team Operations** Red vs blue vs purple teaming. Purple-team exercise structure — pre-brief, execution, hot wash, after-action review. Detection-rule lifecycle (red opens gap, purple writes detection, blue tunes). VECTR for tracking purple-team exercises. ATT&CK Navigator coverage maps. Hunt reports format — finding, evidence, scope, recommendation. Reporting to executives vs technical teams. Tabletop exercises. Coordinated disclosure ethics (CERT-In responsible disclosure for India). Career paths — threat hunter, detection engineer, red team operator, purple team lead.

Reference Books:

1. The Threat Hunter's Handbook Jared Atkinson, Roberto Rodriguez · self-published / Specter Ops · 2020 Free chapters
2. Adversary Emulation Plans (MITRE Engenuity) MITRE Engenuity · attack.mitre.org · 2024 Free
3. Practical Threat Intelligence and Data-Driven Threat Hunting Valentina Costa-Gazcón · Packt · 2021 ISBN: 978-1-83855-637-0
4. Detection Engineering — Building Detection Capabilities Megan Roddie, Jason Deyalsingh, Gary Katz · No Starch Press · 2023 ISBN: 978-1-7185-0381-8
5. The Hacker Playbook 3 Peter Kim · self-published · 2018 ISBN: 978-1-980901-75-3

Dr. K. BHAGAVAN 31/7/26
HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Gu

6. Operator Handbook — Red Team + OSINT + Blue Team Reference Joshua Picolet · self-published · 2022 ISBN: 978-1-734709-91-3
7. Applied Network Security Monitoring Chris Sanders, Jason Smith · Syngress · 2014 ISBN: 978-0-12-417208-1

DeFi Protocols & Tokenomics Engineering

COURSE CODE: 26CA61D1E
COURSE NAME : DeFi Protocols & Tokenomics Engineering
L-T-P-S :3-0-3-0
PRE-REQ : PYPS

CO. No	Course Outcome Description	PO	BTL
CO1	Apply advanced Solidity 0.8.x patterns — assembly, transient storage (EIP-1153), custom errors, immutables, packed structs, gas optimization, ERC-2771 meta-tx, EIP-712 signatures — using Foundry / Hardhat with fuzz + invariant testing.	PO1,PO2	3
CO2	Apply DeFi primitives — Uniswap V2/V3/V4 AMMs, Curve StableSwap, Compound III/Aave V3 lending, Velodrome ve(3,3), GMX perpetuals, Yearn vaults — via direct integration and forking.	PO1,PO4	3
CO3	Analyze tokenomics — token supply schedules, vesting, emissions, ve-tokenomics, liquidity mining, protocol-owned liquidity, bonding curves, real-yield models — and design new protocol token-economic systems.	PO2,PO3	4
CO4	Analyze oracle systems — Chainlink Data Feeds, Pyth, Tellor, UMA, oracle TWAP — and MEV (front-running, sandwich, JIT liquidity), with mitigation patterns (commit-reveal, batch auctions, MEV-Boost).	PO2,PO4	4
CO5	Evaluate DeFi security — reentrancy, integer overflow, price manipulation, flash loan attacks, governance attacks, signature replay, upgradeable proxy pitfalls — using Slither, Mythril, Echidna, Foundry fuzzing, and formal verification with Certora.	PO2,PO6	5
CO6	Evaluate multi-chain deployment — Ethereum L1, L2 rollups (Arbitrum, Optimism, Base, zkSync), Polygon PoS / zkEVM — and the Indian context (RBI digital asset advisories, 2022-23 crypto tax framework, FEMA implications, sandbox eligibility).	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Advanced Solidity & Foundry Mastery** Solidity 0.8.x advanced — custom errors, immutables, packed structs, transient storage (EIP-1153), assembly basics for gas-critical paths. Library patterns.

31/7/26
HoD-CSA
Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-522502

	Diamond Standard (EIP-2535). Upgradeable proxies — Transparent, UUPS, Beacon. ERC-2771 meta-transactions, EIP-712 typed signatures. Foundry — forge, cast, anvil, chisel; fuzz tests; invariant tests; fork tests; gas reports; coverage. Hardhat as alternative. Best-practice repo structure inspired by Uniswap, Aave, Compound.
M2	**DeFi Primitives & Forking** AMMs — Uniswap V2 (constant product), V3 (concentrated liquidity, ticks), V4 (hooks, singleton). Curve StableSwap invariant. Lending — Compound III (Comet), Aave V3 (e-mode, isolation, eMode). Perpetuals — GMX V2 (GLP / GM pools), dYdX, Hyperliquid model. Yield aggregators — Yearn V3 vaults. Forking mainnet for integration tests. Live integration with Uniswap V3 router. Reading and understanding production code at Uniswap v3-core / v3-periphery.
M3	**Tokenomics & Mechanism Design** Token supply schedules — fixed, inflationary, deflationary (burn). Vesting — linear, cliff, custom. Emissions — Curve gauges, Velodrome ve(3,3). Liquidity mining models. Protocol-owned liquidity (OlympusDAO bonding). Bonding curves (Bancor, Uniswap-style). veToken (Curve / Aave / Frax). Real yield (GMX, Gains). Token classification — utility, governance, security (Howey test). Modeling token velocity, MV=PT, market-cap-to-TVL. Designing new tokens with Cadcad / Radcad simulation.
M4	**Oracles, MEV & Cross-Chain** Chainlink Data Feeds, CCIP, VRF, Functions. Pyth pull-based oracle. UMA optimistic oracle. Tellor. Oracle TWAP from Uniswap V3. Oracle manipulation attacks — flash loan + thin-pool. MEV — front-running, sandwich, back-running, time-bandit. JIT (just-in-time) liquidity. Mitigation — commit-reveal, CoW Protocol batch auctions, MEV-Boost, Flashbots Protect, private mempool. Cross-chain bridges — LayerZero, Wormhole, CCIP, Hop, Synapse. Bridge security incidents. Account abstraction (ERC-4337).
M5	**DeFi Security & Audit** Top-10 DeFi vulns — reentrancy (cross-function, read-only), oracle price manipulation, flash loan attacks, governance attacks, signature replay (chainId), front-running, upgradeable proxy pitfalls (storage clashes, init), access control, integer issues, default values. SWC registry. Slither static analyzer. Mythril symbolic execution. Echidna property-based fuzzing. Foundry invariant testing. Formal verification with Certora. Security checklists — Trail of Bits Building Secure Contracts, ConsenSys best practices. Audit reports — how to read & learn from Spearbit, Trail of Bits, ConsenSys Diligence.
M6	**Multi-Chain Deployment & Indian Context** Ethereum L1 vs L2s — Arbitrum One / Nova, Optimism, Base, zkSync Era, Linea, Scroll, Polygon zkEVM. Bridging assets. L2 sequencer risk. Optimistic vs ZK rollups. Polygon PoS as commodity chain. Deployment with Foundry script + verify on Etherscan / Arbiscan. Indian context — RBI 2018 ban (overturned), 2022 crypto tax (30% gains, 1% TDS), FEMA implications for cross-border. RBI Digital Rupee (CBDC). MEITY's Web3 strategy. Sandbox eligibility (RBI / IFSCA). Indian Web3 unicorns — Polygon, CoinDCX, CoinSwitch, WazirX. Compliance frameworks for legitimate DeFi.

Reference Books:

Dr. K. BHAGAVAN
HOD-Computer Science And Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-522

HoD-CSA

1. Mastering Ethereum Andreas M. Antonopoulos, Gavin Wood · O'Reilly · 2018 ISBN: 978-1-491-97194-9 · Free online
2. How to DeFi: Advanced CoinGecko · 2021 Free · landmark.coingecko.com
3. Hands-On Smart Contract Development with Solidity and Ethereum Kevin Solorio, Randall Kanna, David H. Hoover · O'Reilly · 2019 ISBN: 978-1-492-04526-7
4. Building Secure Contracts (Trail of Bits) Trail of Bits Free · github.com/crytic/building-secure-contracts
5. Token Engineering Fundamentals Trent McConaghy · BlockScience / Ocean Protocol · 2022 Free essays · oceanprotocol.com ·
6. Foundry Book book.getfoundry.sh · Paradigm Free · canonical Foundry reference
7. Cryptocurrency Laws & Regulations India Subhash Bhutoria · Universal LexisNexis · 2023 ISBN: 978-93-93702-32-1

Analytics Engineering & Modern Data Platforms

COURSE CODE: 26CA61E1E
COURSE NAME : Analytics Engineering & Modern Data Platforms
L-T-P-S :3-0-3-0
PRE-REQ : PYPS

CO. No	Course Outcome Description	PO	BTL
CO1	Apply modern data platform architecture — cloud DW (Snowflake, BigQuery, Databricks, Redshift), data lakehouse (Iceberg, Delta, Hudi), ELT pipelines, separation of storage and compute — choosing the right tool for each workload.	PO1,PO4	3
CO2	Apply ingestion patterns — Fivetran / Airbyte / Stitch (SaaS connectors), CDC with Debezium / dbt-source-freshness, batch vs streaming (Kafka / Kinesis / Confluent), Indian-source connectors (Razorpay, Zoho, Tally, GST APIs).	PO1,PO4	3
CO3	Analyze dbt-based transformations — layered architecture (staging → intermediate → marts), models, sources, seeds, snapshots, exposures, tests (generic + singular), macros, packages, dbt-utils, dbt_project_evaluator, CI / CD with dbt Cloud or GitHub Actions.	PO2,PO3	4
CO4	Analyze semantic layers and metric stores — dbt MetricFlow, Cube, AtScale, Lightdash — for governed self-service BI; dimensional modelling (Kimball star schema), slowly-changing dimensions (SCD-2), one-big-table (OBT), data vault basics.	PO2,PO4	4
CO5	Evaluate data quality & observability — Great Expectations, dbt tests, Soda Core, Monte Carlo / Lightup / Acceldata data observability platforms, freshness / volume / schema drift / distribution checks, data contracts (PACT-style).	PO4,PO6	5
CO6	Evaluate orchestration (Airflow, Dagster, Prefect), governance (Unity Catalog, Atlan, Collibra, OpenMetadata, DataHub), Indian data-residency requirements (RBI Sep-2018 payments-data	PO2,PO4,PO6	5

31/7/26
Dr. K [illegible] HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	localisation, DPDP Act, IT Rules 2011 SPDI), cost optimization (warehouse sizing, query tuning, BI tool selection).		

Syllabus:

Module No.	Syllabus
M1	**Modern Data Platform Architecture** Modern data stack thesis — storage / compute separation. Cloud DW comparison — Snowflake (multi-cluster shared data, virtual warehouses, time travel, data sharing), BigQuery (serverless, slots, BI Engine), Databricks Lakehouse (Delta + Photon + Unity), Redshift (RA3 + Spectrum). Lakehouse formats — Apache Iceberg (table format, snapshot isolation, time travel), Delta Lake (ACID, optimistic concurrency), Apache Hudi (CoW vs MoR). Data lake on object storage (S3, GCS, Azure Blob). When to choose lake / warehouse / lakehouse. Indian-context cloud regions for data residency.
M2	**Ingestion & CDC** Ingestion taxonomy — batch (file dump, JDBC), CDC (Debezium / Fivetran / Hevo / Airbyte CDC), streaming (Kafka / Kinesis / Confluent Cloud / Pub-Sub). SaaS connectors — Fivetran, Airbyte (open-source + cloud), Hevo Data (Indian-origin, used by Razorpay / Cred / Meesho), Stitch. Connector authoring with Airbyte CDK. Schema evolution. Source freshness checks. Streaming vs micro-batch trade-offs. Confluent Cloud / MSK in India regions. Indian-source connectors — Razorpay, Zoho Books, Tally Prime, GST GSTR-1 APIs, NSE / BSE bhavcopy.
M3	**dbt Mastery & Layered Architecture** dbt Core vs dbt Cloud. Project structure — staging / intermediate / marts, naming conventions (stg_ / int_ / fct_ / dim_), 1:1 stg with sources. Sources, seeds, models, snapshots, exposures, analyses. Materializations — view, table, incremental (append / delete+insert / merge), ephemeral. Tests — generic (unique, not_null, accepted_values, relationships) + singular (custom SQL) + dbt_expectations. Macros, jinja, dbt-utils, dbt_audit_helper, dbt_project_evaluator. Documentation & lineage with dbt docs. CI/CD — SlimCI in dbt Cloud, GitHub Actions with state:modified, dbt-coverage.
M4	**Semantic Layers & Dimensional Modelling** Why semantic layers — metric definition drift, BI tool sprawl. dbt MetricFlow — semantic models, dimensions, entities, measures, metrics (simple, ratio, derived, cumulative, conversion). Cube.dev — data model, pre-aggregations. AtScale, Lightdash. BI tools — Looker, Metabase, Hex, Mode, Superset, Power BI, Tableau. Dimensional modelling refresher — Kimball star schema, fact / dimension tables, conformed dimensions. Slowly-Changing Dimensions (SCD-1, SCD-2, SCD-6). One-Big-Table (OBT) for OLAP cubes. Data Vault 2.0 basics. Choosing OBT vs star vs DV.
M5	**Data Quality, Contracts & Observability** Data quality dimensions — accuracy, completeness, timeliness, uniqueness, validity, consistency. Great Expectations — expectation suites, validation, data docs. dbt tests deep-dive. Soda Core / Soda Cloud. Data observability platforms — Monte Carlo, Lightup, Acceldata (Indian-origin), Bigeye. Five pillars of data

Dr. K. BHAGAVAN 3/7/26 HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	observability — freshness, volume, schema, distribution, lineage. Data contracts — PACT-style API contracts adapted for data, schema-as-code, versioning, breaking-change detection. dbt model contracts (v1.5+). Pact, Buz, Datasette. Incident management for data — data SRE patterns.
M6	**Orchestration, Governance & Cost** Orchestration — Airflow (DAG patterns, sensors, KubernetesPodOperator), Dagster (asset-based, software-defined assets, IO managers, dbt-Dagster integration), Prefect (flows, deployments, agents). Governance — Unity Catalog, Atlan, Collibra, OpenMetadata (open-source, used in India), DataHub (LinkedIn open-source), data lineage, glossary, classification. Indian data residency — RBI Sep-2018 payments-data localisation, DPDP Act 2023 cross-border, IT Rules 2011 SPDI. Cost optimization — warehouse sizing (XS-L / Snowflake), query tuning, partitioning, clustering, materialized views, BI tool license cost. FinOps for data.

Reference Books:

1. The Data Warehouse Toolkit (3rd ed) Ralph Kimball, Margy Ross · Wiley · 2013 ISBN: 978-1-118-53080-9
2. Analytics Engineering with SQL and dbt Rui Pedro Machado, Helder Russa · O'Reilly · 2024 ISBN: 978-1-098-14999-8
3. Fundamentals of Data Engineering Joe Reis, Matt Housley · O'Reilly · 2022 ISBN: 978-1-098-10830-7
4. Data Mesh: Delivering Data-Driven Value at Scale Zhamak Dehghani · O'Reilly · 2022 ISBN: 978-1-492-09239-1
5. Snowflake: The Definitive Guide Joyce Kay Avila · O'Reilly · 2022 ISBN: 978-1-098-10381-4
6. Designing Data-Intensive Applications Martin Kleppmann · O'Reilly · 2017 ISBN: 978-1-449-37332-0
7. Building a Scalable Data Warehouse with Data Vault 2.0 Daniel Linstedt · Morgan Kaufmann · 2015 ISBN: 978-0-12-802510-9

Computer Vision & Multimodal AI

COURSE CODE: 26CA61A2E
COURSE NAME : Computer Vision & Multimodal AI
L-T-P-S :3-0-3-0
PRE-REQ : ADRG

CO. No	Course Outcome Description	PO	BTL
CO1	Apply CNN architectures (ResNet, EfficientNet, ConvNeXt) and Vision Transformers (ViT, Swin, DINOv2) for image classification, transfer learning, and fine-tuning.	PO1,PO4	3

HoD-CSA 3/1/26
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO2	Apply object detection (YOLO v8/v9/v10/v11, RT-DETR, DETR) and segmentation (Mask R-CNN, SAM, SAM 2) for production CV pipelines.	PO1,PO4	3
CO3	Analyze Vision-Language Models (VLMs) — CLIP, SigLIP, BLIP-2, LLaVA, Claude/GPT-4o vision — for image understanding, captioning, and visual question answering.	PO2,PO3	4
CO4	Apply OCR & document AI (Tesseract, PaddleOCR, Donut, Marker, Mistral-OCR, Claude vision) for document understanding tasks.	PO1,PO4	3
CO5	Analyze video understanding (action recognition, tracking, temporal segmentation) and multimodal fusion patterns for time-series visual data.	PO2,PO4	4
CO6	valuate CV deployment options — cloud APIs vs self-hosted, ONNX/TensorRT optimization, edge devices (Jetson, Coral), latency vs accuracy trade-offs — for real-time vision systems.	PO2,PO4,PO5	5

Syllabus:

Module No.	Syllabus
M1	**Modern Image Classification** CNN architecture recap (VGG, ResNet, EfficientNet, ConvNeXt). Transfer learning & fine-tuning. Vision Transformers — ViT, Swin Transformer V2, DeiT. Self-supervised vision — DINOv2, MAE. timm library. Data augmentation — AutoAugment, RandAugment, Mixup. Class imbalance handling. Test-time augmentation. Calibration. Confidence-aware inference.
M2	**Detection & Segmentation** Object detection lineage — R-CNN family, single-shot (SSD, YOLO). Modern YOLO — v8, v9, v10, v11. RT-DETR (real-time DETR). Anchor-free detectors. Instance segmentation — Mask R-CNN, YOLO-Seg. Semantic segmentation — U-Net, SegFormer. SAM and SAM 2 for promptable segmentation. Tracking — SORT, DeepSORT, ByteTrack. Annotation tools — CVAT, Roboflow, Label Studio.
M3	**Vision-Language Models (VLMs)** CLIP architecture & training. SigLIP improvements. Visual encoders + LLM bridges — BLIP-2, LLaVA, Idefics, Qwen-VL, InternVL. Frontier VLMs — GPT-4o vision, Claude 3.5 Sonnet vision, Gemini Pro Vision. Visual prompting — image grounding, set-of-mark prompting. Visual question answering. Multi-image reasoning. Open-vocabulary detection (OWL-ViT, Grounding DINO). Few-shot visual learning.
M4	**OCR & Document AI** Classical OCR — Tesseract, PaddleOCR. Modern document understanding — LayoutLMv3, Donut (OCR-free), Marker (PDF to Markdown). Mistral-OCR. Claude/GPT-4o for direct document Q&A on images. Table extraction — Camelot, Tabula, Unstructured. Form parsing — AWS Textract, Azure Document Intelligence. Handwriting recognition. Indic-script OCR (Devanagari, Tamil, Telugu) — AI4Bharat IndicOCR. Multi-page document RAG.

Dr. K. BHAGAVAN 5/2/26
HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K L Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M5	**Video Understanding** Video classification — I3D, SlowFast, VideoMAE. Action detection — AVA, AVA-Kinetics. Temporal action localization. Object tracking on video. Person re-identification basics. Video VLMs — Video-LLaVA, VideoChat, Gemini long-context video. Multimodal fusion — concatenation, attention, gated fusion, **cross-**attention. Real-time inference patterns. Edge video on Jetson Nano / Orin.
M6	**Production CV Engineering** Cloud-API CV (AWS Rekognition, GCP Vision, Claude/GPT-4o vision) vs self-hosted. ONNX export and runtime. TensorRT optimization. Quantization — INT8, FP16. Edge deployment — Jetson Orin, Google Coral, OAK-D. Triton Inference Server for CV. Frame-rate budgets. Cost models per inference. Data drift detection on visual features. Model registry for vision. CV-MLOps with Weights & Biases.

Reference Books:

1. Modern Computer Vision with PyTorch (2nd ed) V. Kishore Ayyadevara, Yeshwanth Reddy · Packt · 2024 ISBN: 978-1-80323-928-6
2. Deep Learning for Vision Systems Mohamed Elgendy · Manning · 2020 ISBN: 978-1-617-29619-2
3. Multimodal Deep Learning Various authors · 2024 (open book) slds-lmu.github.io/seminar_multimodal_dl · Free
4. Computer Vision: Algorithms and Applications (2nd ed) Richard Szeliski · Springer · 2022 ISBN: 978-3-030-34372-9 · Free
5. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed) Aurélien Géron · O'Reilly · 2022 ISBN: 978-1-098-12597-8
6. Digital Image Processing (4th ed) — Indian textbook Rafael Gonzalez, Richard Woods · Pearson India · 2018 ISBN: 978-93-528-66952

Event-Driven & Stream Processing Architectures

COURSE CODE: 26CA61B2E
COURSE NAME : Event-Driven & Stream Processing Architectures
L-T-P-S :3-0-3-0
PRE-REQ : CDVO

CO. No	Course Outcome Description	PO	BTL
CO1	Apply Apache Kafka deeply — topics, partitions, replicas, consumer groups, offsets, idempotent producers, transactional producers, log compaction — for production-grade event streaming.	PO1,PO5	3
CO2	Analyze stream-processing semantics — event time vs processing time, watermarks, windowing (tumbling, sliding, session), late arrivals, exactly-once vs at-least-once vs at-most-once.	PO2,PO4	4

31/1/26
HoD-CSA
Dr. K. BHARGAVI
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO3	Apply stream-processing engines — Apache Flink (DataStream API), Spark Structured Streaming, Kafka Streams, ksqlDB — for real-time pipelines.	PO1,PO4	3
CO4	Analyze event-driven architecture patterns — event sourcing, CQRS, event-driven microservices, choreography vs orchestration, schema evolution — for distributed systems.	PO2,PO3	4
CO5	Apply real-time use cases — CDC pipelines, real-time fraud scoring, real-time analytics, IoT telemetry processing, real-time recommendation feature stores.	PO1,PO2	3
CO6	Evaluate stream-pipeline observability, scaling, backpressure handling, dead-letter strategies, and end-to-end SLO management for production streaming workloads.	PO2,PO5,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Apache Kafka In-Depth** Kafka architecture — brokers, ZooKeeper / KRaft mode, topic, partition, replica. Producer internals — batching, compression, acks=0/1/all, idempotent producer (max.in.flight=5 for ordering). Consumer groups, partition assignment strategies (range, round-robin, sticky), consumer rebalance protocol. Offsets — committing strategies, exactly-once semantics with transactions. Log compaction. Retention. Replication factor & min.insync.replicas. Schema Registry (Confluent, Apicurio) — Avro, Protobuf, JSON Schema. Schema evolution rules — backward, forward, full compatibility.
M2	**Stream Semantics & Time** Event time vs processing time vs ingestion time — the three "times" in streaming. Watermarks — what they mean, how Flink computes them, the late-data trade-off. Windowing — tumbling, sliding, session, global. Window triggers and accumulation modes. Late arrivals — allowed lateness, side outputs. Exactly-once across pipeline boundaries (source-to-sink). Two-phase commit sinks. Checkpointing in Flink. Watermark propagation across operators. Tyler Akidau's "Streaming Systems" mental model.
M3	**Stream-Processing Engines** Apache Flink — DataStream API, Table API, SQL. State management — ValueState, ListState, MapState, RocksDB state backend. Flink savepoints vs checkpoints. Spark Structured Streaming — micro-batch model, continuous processing mode. Kafka Streams — lightweight library approach, KStream/KTable abstraction. ksqlDB — SQL on Kafka topics. Apache Beam — portable pipeline definition. Choosing the right engine — latency, throughput, exactly-once needs, operational complexity.
M4	**Event-Driven Architecture** Event sourcing — persisting events as source of truth, derived views. CQRS — command-query separation, read models. Event-driven microservices — choreography vs orchestration revisited from MSDS perspective. Outbox pattern + CDC integration. Schema evolution under live traffic — the producer-multi-version

Dr. K. BHAGAVAN 21/3/26
HoD-CSA
HOD Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur Dist. Andhra Pradesh-522502

	dance. Event versioning. Domain events vs integration events. Event modeling exercises.
M5	**Real-Time Use Cases** CDC pipelines — Debezium for PostgreSQL/MySQL/MongoDB to Kafka. Real-time fraud scoring — feature aggregation, online ML inference. Real-time analytics dashboards (Redpanda, Materialize, RisingWave). IoT telemetry — high-volume sensor data, time-series state. Real-time recommendation feature stores (Feast online + Kafka-backed). Real-time alerting (anomaly detection on streams). Geo-streaming for ride-sharing & delivery (Indian use cases at Swiggy, Zomato, Ola scale).
M6	**Production Operations** Stream observability — Kafka lag, consumer-group health, JMX metrics, throughput, end-to-end latency. Backpressure — what it is, how each engine handles it. Scaling Kafka — partition count, broker scaling. Scaling Flink — parallelism, slot management, savepoint-based rescaling. Dead-letter topics for poison messages. Debezium connector failures. SLO design for streaming — latency-based SLIs (P99 end-to-end), freshness SLIs. Cost optimization — tiered storage on MSK, retention policies, compression.

Reference Books:

1. Kafka — The Definitive Guide (2nd ed) Gwen Shapira, Todd Palino, Rajini Sivaram, Krit Petty · O'Reilly · 2021 ISBN: 978-1-492-04308-9
2. Streaming Systems Tyler Akidau, Slava Chernyak, Reuven Lax · O'Reilly · 2018 ISBN: 978-1-491-98387-4
3. Stream Processing with Apache Flink Fabian Hueske, Vasiliki Kalavri · O'Reilly · 2019 ISBN: 978-1-491-97429-2
4. Designing Event-Driven Systems Ben Stopford · O'Reilly · 2018 Free O'Reilly
5. Building Event-Driven Microservices Adam Bellemare · O'Reilly · 2020 ISBN: 978-1-492-05789-5
6. Designing Data-Intensive Applications Martin Kleppmann · O'Reilly · 2017 ISBN: 978-1-449-37332-0

Privacy Engineering & Data Protection (DPDP, GDPR)

COURSE CODE: 26CA61C2E
COURSE NAME : Privacy Engineering & Data Protection (DPDP, GDPR)
L-T-P-S :3-0-3-0
PRE-REQ : DBDA

31/1/26
HoD-CSA
Dr. K. [illegible] SWAMY
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO. No	Course Outcome Description	PO	BTL
CO1	Analyze India's Digital Personal Data Protection (DPDP) Act 2023, EU GDPR, California CCPA/CPRA, and Singapore PDPA — compare obligations, definitions, lawful bases, and penalties.	PO2,PO6	4
CO2	Apply consent management platforms and design consent UI flows (notice, granular consent, withdrawal) consistent with DPDP/GDPR requirements.	PO1,PO4	3
CO3	Apply data subject rights (DSR) workflows — access, rectification, erasure, portability, objection — including identity verification and timeline tracking.	PO1,PO4	3
CO4	Analyze Privacy-Enhancing Technologies (PETs) — k-anonymity, l-diversity, t-closeness, differential privacy, homomorphic encryption, secure multiparty computation, federated learning — for use cases.	PO2,PO6	4
CO5	Apply privacy-by-design patterns — data minimization, purpose limitation, encryption schemes (envelope, AEAD, KMS-managed keys), tokenization, secure deletion, retention policies.	PO1,PO4	3
CO6	Evaluate Data Protection Impact Assessments (DPIAs), Privacy Impact Assessments (PIAs), data-flow mapping, vendor risk, breach-notification workflows for regulatory readiness.	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Privacy Law Landscape** India's Digital Personal Data Protection (DPDP) Act 2023 — key definitions (data principal, data fiduciary, data processor, significant data fiduciary), grounds for processing, consent requirements, children's data, rights, penalties (up to ₹250 crore). EU General Data Protection Regulation (GDPR) — lawful bases (Art 6), special categories (Art 9), data subject rights (Arts 15-22), 72-hour breach notification (Art 33). California Consumer Privacy Act (CCPA) + California Privacy Rights Act (CPRA). Singapore PDPA, Brazil LGPD overview. Cross-border data transfer rules — SCCs, BCRs, adequacy decisions, India's whitelist approach.
M2	**Consent Management** Consent under DPDP — clear, specific, informed, unambiguous, free. The DPDP "consent manager" intermediary concept. GDPR consent tests (freely given, specific, informed, unambiguous; can be withdrawn). Consent UI patterns — layered notice, granular toggles, no dark patterns. CMP (Consent Management Platform) implementations — OneTrust, Cookiebot, TrustArc, Didomi, open-source alternatives. IAB TCF for advertising. Cookie banners that are actually compliant. Implicit consent through "continued use" — why it fails. Children's consent under DPDP (verifiable parental consent).

31/7/26
HOD-CSA
Dr. K. Bhavani
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh

M3	**Data Subject Rights** Data subject rights catalog — Right to Information, Access, Correction, Erasure, Portability, Object, Withdrawal of Consent (DPDP), plus GDPR's Right to Restriction and Automated-Decision-Making restrictions. DSR workflow design — intake (web form, email), identity verification (without collecting MORE data), timeline (typically 30 days), response format. Right-to-erasure challenges — backups, audit logs, third parties, derived data. Portability format expectations (machine-readable). DSR ticketing systems. Proof-of-completion for compliance audits.
M4	**Privacy-Enhancing Technologies** Anonymization vs pseudonymization — the DPDP/GDPR distinction. k-anonymity (Sweeney) and its weakness against linkage attacks. l-diversity, t-closeness as improvements. Differential Privacy — Laplace mechanism, Gaussian mechanism, epsilon-budget. DP libraries — Google's open-source DP, Tumult Analytics, OpenDP. Differential Privacy as deployed (US Census 2020). Homomorphic encryption (FHE basics) — CKKS, BFV schemes; libraries (Microsoft SEAL, OpenFHE). Secure Multiparty Computation (SMPC) — secret sharing, garbled circuits. Federated Learning (Google, Apple) for ML on private data.
M5	**Privacy-by-Design Engineering** Cavoukian's 7 Principles. Data minimization in API design (only collect what's needed). Purpose limitation enforced via policy-as-code. Encryption schemes for compliance — envelope encryption, AEAD (AES-GCM, ChaCha20-Poly1305), KMS-managed key hierarchies (AWS KMS, GCP KMS, HashiCorp Vault). Tokenization (preserving format vs not). Field-level encryption (MongoDB CSFLE, AWS QLDB). Secure deletion (cryptographic shredding by destroying KEK). Retention policies as code. PII detection libraries (Microsoft Presidio, AWS Macie). Audit-grade logging schema (who accessed what when why).
M6	**DPIA, Audits & Breach Response** Data Protection Impact Assessment (DPIA) under GDPR. PIA template adapted for DPDP. Threshold tests — when DPIA is required. Conducting DPIA — data flows, risks, mitigations. Data-flow mapping techniques (LINDDUN methodology). Vendor risk assessment — processor selection, sub-processor chains, SCC clauses. Breach response — DPDP's notification requirement to Data Protection Board, GDPR's 72-hour rule. Breach severity classification. Communications plan. Indian Data Protection Board concept. ISO 27701 (privacy extension to 27001) overview. Career paths — Privacy Engineer, DPO, Privacy Counsel.

Reference Books:

1. The Digital Personal Data Protection Act 2023 — A Practitioner's Guide Multiple Indian authors · NLSIU / NALSAR commentaries · 2024 Various ISBNs
2. Privacy Engineering — A Data Flow and Ontological Approach Ian Oliver · CreateSpace · 2014 ISBN: 978-1-4974-5341-3
3. The Algorithmic Foundations of Differential Privacy Cynthia Dwork, Aaron Roth · Foundations & Trends · 2014 Free ·

HoD-CSA

Dr. K. [illegible] 31/1/26
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

4. A Practical Guide to Networking, Privacy, and Security in iOS 17 (technical privacy) Glenn Fleishman · TidBITS · 2024 ISBN: 978-1-953819-52-7
5. Strategic Privacy by Design (2nd ed) R. Jason Cronk · IAPP · 2022 ISBN: 978-1-948771-69-3
6. GDPR for Engineers Ehsan Foroughi · self-published · 2018 ISBN: 978-1-721977-65-0
7. Privacy in the Age of Big Data — Recognizing Threats, Defending Your Rights Theresa Payton · Rowman & Littlefield · 2014 ISBN: 978-1-4422-2545-4 ·

Fintech Apps — Payments, Trading, Lending

COURSE CODE: 26CA61D2E
COURSE NAME : Fintech Apps — Payments, Trading, Lending
L-T-P-S :3-0-3-0
PRE-REQ : FSPD

CO. No	Course Outcome Description	PO	BTL
CO1	Apply UPI integration patterns — UPI 2.0, UPI Lite, UPI mandates, intent + collect flows, NPCI specs, PSP-bank-acquirer architecture — using sandbox APIs (Razorpay UPI, PhonePe PG, Cashfree).	PO1,PO4	3
CO2	Apply card payments — PCI-DSS 4.0 scoping, tokenization (RBI tokenization mandate Sep-2022), 3-D Secure 2.0, ISO 8583 fundamentals, MDR economics, BBPS & e-NACH.	PO1,PO4	3
CO3	Analyze trading systems — Order Management System (OMS), market-data ingestion via Zerodha Kite Connect / Upstox / ICICIdirect Breeze, FIX protocol primer, order types (limit, market, SL, OCO), risk-management gates, NSE/BSE microstructure.	PO2,PO4	4
CO4	Analyze lending architecture — Loan Origination System (LOS), Loan Management System (LMS), credit decisioning, Account	PO2,PO4	4

Dr. K. BHAGAVAN 31/3/26
HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh

	Aggregator framework, RBI Digital Lending Guidelines (Sep-2022), KYC tiers, TransUnion CIBIL / Experian / Equifax integration.		
CO5	Evaluate fintech security — PCI-DSS scope reduction, tokenization, vault patterns, fraud detection (rule + ML), velocity checks, device fingerprinting, AML/KYC, sanctions screening, RBI / FIU-IND reporting.	PO2,PO6	5
CO6	Evaluate Indian fintech regulatory landscape — RBI PA/PG licensing, PSS Act 2007, NPCI bylaws, SEBI broker rules, RBI Digital Lending Guidelines, Account Aggregator (RBI 2016/2022), DPDP Act for fintech, SEBI cyber for capital markets, IRDAI for insurtech.	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**UPI Architecture & Integration** UPI architecture — PSP, payer bank, payee bank, NPCI switch. UPI 2.0 — one-time mandate, recurring mandate, IPO blocking. UPI Lite, UPI Lite X, UPI Circle. Intent vs collect flows. UPI deep links. UPI on RuPay credit cards. Razorpay UPI sandbox, PhonePe PG, Cashfree integration patterns. Server-side webhook handling. Idempotency keys for payment retries. Reconciliation flows. Settlement T+0 / T+1. Payments aggregator vs payments gateway.
M2	**Card Payments, BBPS & e-NACH** PCI-DSS 4.0 deep-dive — 12 requirements, levels 1-4. Scope reduction via tokenization. RBI tokenization mandate (Sep-2022). Card-on-file (CoF) tokenization with networks (Visa, Mastercard, RuPay). 3-D Secure 2.0 — biometric, OTP-less. ISO 8583 message structure. EMV chip basics. MDR & interchange economics. BBPS (Bharat Bill Payment System) — BBPCU, BBPOU, agent flow. e-NACH for recurring debits. NETC FASTag.
M3	**Trading Systems & Market Data** OMS architecture — order entry, validation, risk gates, exchange routing, ack flow. Market data — tick / minute / depth feeds. Zerodha Kite Connect API deep-dive — auth, place / modify / cancel order, postbacks. Upstox API. ICICIdirect Breeze. Smallcase API. FIX protocol primer (4.4 / 5.0). NSE NEAT, BSE BOLT. Order types — LIMIT, MARKET, SL, SL-M, OCO, AMO, GTT. Bracket orders. Pre-trade risk gates — margin, exposure, F&O lot validation. Settlement T+1. Backtesting frameworks — Backtrader, VectorBT.
M4	**Lending — LOS, LMS & Account Aggregator** Loan Origination System (LOS) flow — lead, KYC, eligibility, decisioning, sanction, e-sign, disbursement. Loan Management System (LMS) — ledger, EMI, prepayment, restructure, NPA classification. Credit decisioning models. CIBIL / Experian / Equifax integration. RBI Digital Lending Guidelines (Sep-2022) — FLDG cap, KFS (Key Fact Statement), data localisation, APR disclosure, no auto-disbursement to merchant accounts. Account Aggregator (AA) — consent flow.

HoD-CSA 3/11/26
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

	FIPs, FIUs. Setu, Finvu, OneMoney, CAMS Finserv. BNPL economics. Co-lending. Microloans.
M5	**Fintech Security & Fraud Detection** PCI-DSS 4.0 scope minimization — tokenization, vault patterns, P2PE. HSM. KMS envelope encryption. Fraud detection — rule engine (Drools-style), velocity checks (max txn per minute / hour / day), behavioural ML, device fingerprinting (FingerprintJS, ThreatMetrix). AML/KYC — eKYC via Aadhaar (UIDAI), DigiLocker, video KYC, CKYC. Sanctions screening — OFAC, UN, RBI lists. PEP screening. Suspicious Transaction Reports (STR) to FIU-IND. Cash Transaction Reports (CTR). RBI Cyber Security Framework. Cybersecurity incidents to CERT-In within 6 hrs.
M6	**Indian Regulatory Landscape** RBI Payment Aggregator (PA) & Payment Gateway (PG) licensing (March-2020 framework). PSS Act 2007. NPCI bylaws. PA-CB (Payment Aggregator Cross-Border). SEBI broker rules — LODR, MCA filings. SEBI margin pledge framework. NSE/BSE compliance. T+1 settlement. RBI Digital Lending Guidelines complete tour. Account Aggregator (RBI 2016/2022). DPDP Act for fintech — consent, retention, breach. Insurance — IRDAI Sandbox. UPI volume caps. RBI Sandbox eligibility for fintech innovation. IFSCA GIFT City sandbox.

Reference Books:

1. Designing Data-Intensive Applications Martin Kleppmann · O'Reilly · 2017 ISBN: 978-1-449-37332-0
2. Trading and Exchanges — Market Microstructure for Practitioners Larry Harris · Oxford · 2003 ISBN: 978-0-19-514470-6
3. Building Algorithmic Trading Systems Kevin Davey · Wiley · 2014 ISBN: 978-1-118-77891-1
4. PCI-DSS 4.0 Standard PCI Security Standards Council Free · pcisecuritystandards.org
5. Indian Banking and Financial Services (Revised) M.Y. Khan · Tata McGraw-Hill · 2020 ISBN: 978-93-87067-79-3
6. Financial Technology in India: Opportunities and Challenges Rajashekar S, Sandhya · Sage · 2022
ISBN: 978-93-94534-00-2
7. RBI Master Directions — PA/PG, Digital Lending, AA RBI · official circulars Free · rbi.org.in

Causal Analytics & Experimentation

3/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002, Ph: +91 - 866 - 3500122, 2576129

COURSE CODE: 26CA61E2E
COURSE NAME : Causal Analytics & Experimentation
L-T-P-S :3-0-3-0
PRE-REQ : AIML

CO. No	Course Outcome Description	PO	BTL
CO1	Apply A/B test design — randomization unit (user/session/visit/cluster), MDE / power / sample-size calculation, sequential testing (mSPRT, AGT), Bayesian A/B (Beta-Binomial, Normal-Normal), peeking corrections, novelty / primacy effects, SUTVA assumption.	PO1,PO4	3
CO2	Apply variance-reduction techniques — CUPED (Microsoft), CUPAC, stratification, post-stratification, regression adjustment, triggered analysis, ratio metrics with delta method, robust standard errors.	PO1,PO2	3
CO3	Analyze quasi-experimental designs when randomization is impossible — difference-in-differences (DiD), synthetic control (Abadie), regression discontinuity (RDD), instrumental variables (IV), interrupted time series, propensity-score matching.	PO2,PO4	4
CO4	Analyze observational causal inference using DoWhy + EconML — DAG-based identification (Pearl), backdoor / front-door criterion, double machine learning (DML), causal forests, T-learner / S-learner / X-learner / R-learner meta-learners.	PO2,PO4	4
CO5	Evaluate switchback / interleaved / cluster-randomized experiments for two-sided marketplaces (Swiggy / Uber / Zomato), interference / SUTVA violations, network effects, geo-experiments (GeoLift, Meta's GeoLift), Bayesian structural time series (CausalImpact).	PO2,PO4	5
CO6	Evaluate uplift modeling for personalization (Causal ML, Uber's Pylift, X-learner), multi-armed bandits (epsilon-greedy, Thompson sampling, LinUCB, contextual bandits), production experimentation platforms (Statsig, Eppo, Optimizely, GrowthBook, internal Swiggy / Flipkart platforms).	PO2,PO3,PO4	5

Syllabus:

Module No.	Syllabus
M1	**A/B Test Design Foundations** Hypothesis formulation — null / alternative; one-sided vs two-sided. Randomization unit selection — user / session / visit / cluster. MDE (minimum detectable effect), power (1-β) typically 0.80, alpha 0.05. Sample-size formulas for proportions and means. Practical significance vs statistical. SUTVA (stable unit treatment value assumption). Sequential testing — mSPRT (mixture sequential probability ratio test) used by Optimizely, AGT (always-valid p-values), group sequential designs. Bayesian A/B — Beta-Binomial conjugate, Normal-Normal, posterior probability of being best. Peeking corrections.

HoD-CSA 31/3/26
Dr. K
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M2	**Variance Reduction** CUPED (Controlled-experiment Using Pre-Existing Data) — Microsoft 2013 paper, ~50% variance reduction on metrics with high pre-experiment correlation. CUPAC (Doordash's extension using ML predictions). Stratified randomization. Post-stratification. Regression adjustment with covariates. Triggered analysis — restrict to users who triggered the experiment surface. Ratio metrics — delta method, jackknife, bootstrap. Heteroskedasticity-robust standard errors (HC0-HC3). Cluster-robust SEs for cluster randomization.
M3	**Quasi-Experiments** When randomization is impossible — pricing decisions, regulation, geographic rollouts. Difference-in-differences (DiD) — parallel trends assumption, two-way fixed effects, recent advances (Goodman-Bacon, Callaway-Sant'Anna for staggered treatment). Synthetic Control (Abadie 2003 California Prop 99) — donor pool, weights, placebo tests. Regression Discontinuity Design (RDD) — sharp / fuzzy, bandwidth selection, McCrary density test. Instrumental Variables (IV) — relevance + exclusion + monotonicity, 2SLS, weak-instrument tests. Interrupted Time Series (ITS). Propensity-Score Matching (PSM).
M4	**Observational Causal Inference (Pearl + Rubin)** Two frameworks — Rubin Potential Outcomes (Y(0), Y(1)), Pearl Structural Causal Models (SCMs + do-operator). DAG construction. Identification — backdoor / front-door / instrumental variable criteria. ATE (average treatment effect), ATT (on the treated), CATE (conditional). DoWhy library — identify → estimate → refute. Refutation tests — placebo treatment, random common cause, dummy outcome, data subset. EconML library. Double Machine Learning (DML) — Chernozhukov et al. Causal Forests (Athey-Wager). Meta-learners — T-learner (two models), S-learner (single), X-learner, R-learner.
M5	**Two-Sided Marketplace Experiments** Why naive A/B fails for marketplaces — interference / SUTVA violations. Switchback experiments — alternate treatment / control across time windows (used by Swiggy / Uber / DoorDash). Cluster randomization (geo-clusters / city-clusters). Two-sided experiments — randomize on supply (restaurants / drivers) vs demand (users) vs both. GeoLift (Meta open-source, used for marketing measurement in cookieless world). Bayesian structural time series — CausalImpact (Google) for marketing campaigns. Synth methods for marketing measurement. Experiments at scale — metric repository, segmentation, simultaneous experiments, A/A tests.
M6	**Uplift, Bandits & Production Platforms** Uplift modeling — predict CATE per user for personalization, treatment-effect heterogeneity. Causal ML (Uber's Pylift), CausalLift, EconML uplift recipes. Two-model approach vs class transformation. Qini curve & AUUC evaluation. Multi-armed bandits — explore-exploit tradeoff, epsilon-greedy, Thompson sampling, UCB1, LinUCB (contextual). When MAB beats A/B (high-velocity, simple metrics) and when it doesn't (long lookback, heterogeneity matters). Production platforms — Statsig (used by OpenAI), Eppo, Optimizely, GrowthBook (open-source), VWO (Indian-origin), internal platforms at Swiggy / Flipkart / Razorpay.

Reference Books:

1. Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing Ron Kohavi, Diane Tang, Ya Xu · Cambridge · 2020 ISBN: 978-1-108-72426-5

31/1/26
HoD-CSA

Dr. K. [illegible]
Prof & Head
Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

2. Causal Inference: The Mixtape Scott Cunningham · Yale · 2021 ISBN: 978-0-300-25168-5 · Free online
3. Causal Inference: What If Miguel A. Hernán, James M. Robins · Chapman & Hall · 2024 Free online · hsph.harvard.edu/miguel-hernan/causal-inference-book
4. The Book of Why Judea Pearl, Dana Mackenzie · Basic Books · 2018 ISBN: 978-0-465-09760-9
5. Causal Inference for the Brave and True Matheus Facure · 2022 Free online · matheusfacure.github.io/python-causality-handbook
6. Mostly Harmless Econometrics Joshua Angrist, Jörn-Steffen Pischke · Princeton · 2009 ISBN: 978-0-691-12035-5
7. Bandit Algorithms Tor Lattimore, Csaba Szepesvári · Cambridge · 2020 ISBN: 978-1-108-48682-9 · Free online · banditalgs.com

Recommender Systems & Personalization at Scale

COURSE CODE: 26CA62A3E
COURSE NAME : Recommender Systems & Personalization at Scale
L-T-P-S :3-0-3-0
PRE-REQ : AIML

CO. No	Course Outcome Description	PO	BTL
CO1	Apply collaborative filtering (user-user, item-item, matrix factorization — ALS, SVD++, BPR), content-based, and hybrid recommender designs to baseline systems on industrial datasets.	PO1,PO4	3
CO2	Analyze deep-learning recommender architectures — two-tower neural retrieval, NCF, DeepFM, DCN-V2, DIN, BST, SASRec, BERT4Rec — for retrieval and ranking layers.	PO2,PO3	4
CO3	Analyze candidate-generation, ranking, re-ranking pipelines (the Pinterest / YouTube / Meta industrial pattern) and embedding-based retrieval at scale (FAISS, ScaNN, Vespa, Qdrant).	PO2,PO4	4
CO4	Apply learning-to-rank techniques (pointwise, pairwise, listwise; LambdaMART, RankNet) and multi-objective ranking with business constraints.	PO1,PO4	3
CO5	Evaluate offline metrics (NDCG, MAP, Recall@K, MRR, Hit-Rate) and online metrics (CTR, watch-time, conversion, retention) including A/B-test design for recommender experiments.	PO2,PO4	5
CO6	Evaluate cold-start strategies, exploration-exploitation (epsilon-greedy, Thompson sampling, contextual bandits), fairness, filter	PO2,PO3,PO4	5

HoD-CSA 31/01/26

Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	bubbles, and bias-aware design for production recommender systems.		

Syllabus:

Module No.	Syllabus
M1	**Classical Recommender Foundations** Recommender problem framing — explicit vs implicit feedback, sparse user-item matrix, long-tail. User-based and item-based collaborative filtering. Matrix factorization — SVD, ALS (Alternating Least Squares for implicit), SVD++ (with bias terms), BPR (Bayesian Personalized Ranking). Content-based recommenders with TF-IDF and embeddings. Hybrid designs — weighted, switching, mixed, feature combination. Surprise library, implicit library. Datasets — MovieLens, Amazon Reviews, Yoochoose, Last.fm. Indian-context: Flipkart, Hotstar, Spotify regional charts.
M2	**Deep Learning for RecSys** Neural Collaborative Filtering (NCF). Wide & Deep (Google). DeepFM (factorization machines + DNN). Deep & Cross V2 (DCN-V2). Deep Interest Network (DIN, Alibaba). Behavior Sequence Transformer (BST). Sequential models — SASRec, BERT4Rec, GRU4Rec. Two-tower neural retrieval (Google & Meta industrial pattern). Sampled softmax for negatives. In-batch negatives. Hard-negative mining. TensorFlow Recommenders (TFRS). Recommenders library
M3	**Industrial Pipeline Architecture** The 3-stage industrial pattern: candidate generation → ranking → re-ranking. YouTube, Instagram, Pinterest, Netflix architecture case-studies. Embedding-based retrieval — FAISS (Facebook), ScaNN (Google), Vespa (Yahoo), Qdrant. Approximate Nearest Neighbor (HNSW, IVF, PQ). Real-time feature stores — Feast, Tecton. Online vs offline features. Feature parity guarantees. Model serving — TF Serving, TorchServe, NVIDIA Triton, BentoML. Latency budgets at retrieval (<30ms) and ranking (<50ms) tiers.
M4	**Learning-to-Rank & Multi-Objective** Pointwise, pairwise, listwise approaches. RankNet, LambdaRank, LambdaMART. XGBoost-Ranker for LTR. ListNet, ListMLE. Calibration of click probabilities. Multi-task learning for ranking — click prediction + watch-time prediction + completion prediction (YouTube paper). Position bias correction. Multi-objective ranking with Pareto optimization. Business constraints — promoted content, diversity, fairness across creator cohorts.
M5	**Evaluation & Online Experimentation** Offline metrics — NDCG@K, MAP, Recall@K, MRR, Hit-Rate@K, AUC. Time-based vs random splits (the temporal-leakage trap). Counterfactual evaluation — IPS (Inverse Propensity Scoring), self-normalized IPS, doubly-robust. Online metrics — CTR, watch-time, completion rate, retention, revenue. A/B testing for recommenders — SUTVA assumption challenges, network effects, novelty effects, primacy bias. Sample-size calculation. Sequential testing. Interleaving experiments.

3/3/26
HoD-CSA
Dr. K. BHAGA
HOD-Computer Science and Applic
Koneru Lakshmaiah Education Foun
(K.L. Deemed to be Universit
Green Fields, VADDESWARA
Guntur District, Andhra Pradesh-5

M6	**Cold-Start, Fairness & Bandits** Cold-start — new users, new items, new domains. Content-features as cold-start proxy. Onboarding flows. Multi-armed bandits — epsilon-greedy, UCB, Thompson sampling. Contextual bandits — LinUCB, neural bandits. Exploration vs exploitation in production. Fairness in recommenders — consumer fairness (across users), provider fairness (across creators). Filter bubbles & echo chambers. Diversity injection — MMR, DPP. LLM-augmented recommenders (sequential prompting, item description generation).

Reference Books:

1. Recommender Systems Handbook (3rd ed) Ricci, Rokach, Shapira (eds) · Springer · 2022 ISBN: 978-1-0716-2196-7
2. Practical Recommender Systems Kim Falk · Manning · 2019 ISBN: 978-1-617-29270-5
3. Deep Learning for Search Tommaso Teofili · Manning · 2019 ISBN: 978-1-617-29479-2
4. Designing Machine Learning Systems Chip Huyen · O'Reilly · 2022 ISBN: 978-1-098-10796-3
5. Bandit Algorithms Tor Lattimore, Csaba Szepesvári · Cambridge · 2020 ISBN: 978-1-108-48682-8
6. Trustworthy Online Controlled Experiments Kohavi, Tang, Xu · Cambridge · 2020 ISBN: 978-1-108-72426-5

SRE & Advanced Observability

COURSE CODE: 26CA62B3E
COURSE NAME : SRE & Advanced Observability
L-T-P-S :3-0-3-0
PRE-REQ : CDVO

CO. No	Course Outcome Description	PO	BTL
CO1	Evaluate Service Level Indicators (SLI), Service Level Objectives (SLO), and error budgets to design reliability targets that balance feature velocity with stability.	PO2,PO4	5
CO2	Apply OpenTelemetry instrumentation (logs, metrics, traces) at framework + library + auto-instrumentation levels for polyglot microservices.	PO1,PO4	3
CO3	Analyze observability backends — Prometheus + Grafana, Loki, Tempo, Jaeger, OpenSearch, Honeycomb, Datadog, New Relic — for storage, query, alerting trade-offs.	PO1,PO2	4
CO4	Apply continuous profiling (Pyroscope, Parca, Pixie/eBPF) and event-driven observability for advanced root-cause analysis.	PO1,PO4	3
CO5	Analyze incident management — alert design, on-call runbooks, escalation policies, blameless post-mortems, MTTD/MTTR metrics, incident command pattern.	PO2,PO3	4

31/1/26
Dr. K. Bhujanga Rao
HoD-CSA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh

CO6	Evaluate capacity planning, load shedding, graceful degradation, and chaos engineering for sustained reliability under growth and failure.	PO2,PO3,PO6	5

Syllabus:

Module No.	Syllabus
M1	**SLI/SLO/Error-Budget Discipline** SRE history (Google), distinction from DevOps. SLI selection — latency, availability, throughput, correctness, freshness, durability. The 4 Golden Signals (latency, traffic, errors, saturation). RED method (rate/errors/duration) for request-driven services. USE method (utilization/saturation/errors) for resources. SLO definition — targets, time windows (30-day rolling), target percentiles. Error budget calculus (1 - SLO target = budget). Burn-rate alerting (Google's multi-window/multi-burn-rate). Sloth as SLO-as-code generator. SLO documents, ownership, SLO reviews.
M2	**Telemetry & OpenTelemetry** Three pillars revisited (logs, metrics, traces) + emerging fourth (profiles) + fifth (events). OpenTelemetry architecture — API, SDK, Collector, exporters. Auto-instrumentation per language (Python, Java, Go, Node). Manual span creation, attributes, baggage. Trace sampling — head-based (probabilistic, rate-limited) vs tail-based (Refinery, Collector tail-sampling). Context propagation (W3C Trace Context). Log-trace correlation (logfmt + trace_id injection). Structured logging best practices. PII redaction in telemetry.
M3	**Observability Backends** Prometheus deep-dive — pull model, PromQL, recording rules, alerting rules, federation, long-term storage (Thanos, Cortex, Mimir). Grafana for visualization — templating, alerting, unified observability. Loki for logs — LogQL, label cardinality discipline. Tempo or Jaeger for traces. OpenSearch + Filebeat for log search. Cardinality explosion as the #1 cost driver. Datadog, New Relic, Honeycomb as managed alternatives — when SaaS makes sense vs self-hosted. eBPF-based observability (Pixie, Cilium Hubble) for zero-instrumentation.
M4	**Profiling & Advanced Observability** Continuous profiling rationale (closing the gap between traces and resource usage). Pyroscope (now Grafana Pyroscope), Parca, Polar Signals. Flame graphs interpretation (Brendan Gregg). pprof format. CPU profiles, memory profiles, mutex profiles, goroutine profiles. eBPF basics — what it can observe at kernel level. Pixie and Falco. Event-driven observability — Honeycomb-style high-cardinality events. BubbleUp / outlier analysis. Distributed tracing as a debugger.
M5	**Incident Management** Alert design philosophy — symptoms not causes, alert fatigue, paging vs ticketing thresholds. PagerDuty / Opsgenie / Squadcast (Indian SaaS) for on-call. Incident command pattern — IC, comms, scribe roles. Severity levels (SEV0–SEV5). Status pages. Customer comms during incidents. Blameless post-mortems — the 5 Whys, contributing factors, action items. MTTD, MTTR, MTBF metrics. Runbook

HoD-CSA 31/3/26
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Andhra Pradesh-522502

	discipline. Game-day exercises. Recovery patterns. Incident reviews and SRE practice maturity.
M6	**Capacity, Load Shedding & Chaos** Capacity planning — load testing (k6, Gatling, Locust), Little's Law, queueing theory basics. Demand forecasting. Headroom targets. Vertical vs horizontal scaling. Auto-scaling policies (HPA, VPA, KEDA event-driven autoscaling). Load shedding strategies — rejecting low-priority traffic. Graceful degradation — serving partial responses when dependencies fail. Adaptive concurrency control. Chaos engineering revisited from MSDS angle but for reliability practice. Toil reduction — automation, reliability engineering culture.

Reference Books:

1. Site Reliability Engineering — How Google Runs Production Systems Beyer, Jones, Petoff, Murphy (eds) · O'Reilly · 2016 Free: sre.google/sre-book
2. The Site Reliability Workbook — Practical Ways to Implement SRE Beyer et al · O'Reilly · 2018 Free: sre.google/workbook
3. Observability Engineering Charity Majors, Liz Fong-Jones, George Miranda · O'Reilly · 2022 ISBN: 978-1-492-07621-6
4. Distributed Systems Observability Cindy Sridharan · O'Reilly · 2018 Free O'Reilly
5. Implementing Service Level Objectives Alex Hidalgo · O'Reilly · 2020 ISBN: 978-1-492-07685-8
6. Systems Performance — Enterprise and the Cloud (2nd ed) Brendan Gregg · Pearson · 2020 ISBN: 978-0-13-682015-4
7. Seeking SRE David Blank-Edelman (ed) · O'Reilly · 2018 ISBN: 978-1-491-97891-7

Cloud Native Security & Zero Trust Architecture

COURSE CODE: 26CA62C3E
COURSE NAME : Cloud Native Security & Zero Trust Architecture
L-T-P-S :3-0-3-0
PRE-REQ : CDVO

CO. No	Course Outcome Description	PO	BTL
CO1	Apply zero-trust principles — identity-as-perimeter, continuous verification, least-privilege, micro-segmentation, lateral-movement prevention — based on NIST SP 800-207 and Google BeyondCorp.	PO1,PO6	3
CO2	Analyze cloud IAM (AWS IAM + IAM Identity Center, Azure AD/Entra, GCP IAM) — policy structure, role assumption,	PO2,PO4	4

3/1/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	federated identity, OIDC for workloads, IAM Access Analyzer for permission boundaries.		
CO3	Analyze workload identity — SPIFFE/SPIRE, Kubernetes ServiceAccounts, IRSA, GKE Workload Identity, Mtls in service mesh (Istio, Linkerd) — for service-to-service authentication.	PO2,PO4	4
CO4	Analyze container & Kubernetes security — admission controllers (OPA Gatekeeper, Kyverno), Pod Security Standards, NetworkPolicies, image scanning (Trivy, Snyk), supply-chain (Cosign, SLSA), runtime security (Falco, Tetragon).	PO2,PO4	4
CO5	Evaluate secrets management (HashiCorp Vault, AWS Secrets Manager, sealed-secrets), encryption-at-rest & in-transit, KMS architecture, key rotation, BYOK/HYOK for regulated industries.	PO4,PO6	5
CO6	Evaluate cloud-security posture management (CSPM) — Wiz, Prowler, Checkov, Cloud Custodian, Sysdig — with Indian regulatory mapping (DPDP Act 2023, RBI cloud guidelines, MEITY Cyber Security Framework).	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Zero Trust Architecture Foundations** NIST SP 800-207 zero-trust tenets — identity-as-perimeter, continuous verification, least privilege, encrypt everything. Castle-and-moat vs zero-trust. BeyondCorp paper deep-dive (Google). The 7 ZTA pillars: identity, device, network, application, data, visibility, automation. Trust scores. Adaptive auth. ZTA reference architectures from CISA, DoD, NIST. Common myths and pitfalls. Roadmap from perimeter-based to ZTA.
M2	**Cloud IAM & Federated Identity** AWS IAM internals — principals, actions, resources, conditions; IAM policy evaluation logic; SCPs; IAM Identity Center (formerly SSO); Permission Sets; cross-account roles; AssumeRole flow; OIDC federation for GitHub Actions, EKS pods. Azure Entra ID — conditional access, PIM. GCP IAM — resource hierarchy, custom roles, Workload Identity Federation. SAML / OIDC / Oauth2 for human and machine identity. Permission boundary best practices. IAM Access Analyzer for unused permission detection.
M3	**Workload Identity & Mtls** SPIFFE specification — SVIDs, trust domains, SPIRE Server & Agent. SPIRE deployment on Kubernetes. Workload attestation. Kubernetes ServiceAccounts, projected tokens, audience-aware tokens. IRSA on AWS EKS — OIDC provider, role trust policies. GKE Workload Identity. Service mesh Mtls — Istio Citadel, Linkerd identity. Authentication policies (PeerAuthentication). Authorization policies in Istio. Certificate rotation. Secret-zero problem and resolution.

Dr. K. SRINIVAN
HoD-CSA
31/7/26
HOD-Computer Science & Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

M4	**Kubernetes & Container Security** Pod Security Standards (Restricted, Baseline, Privileged). Admission controllers — OPA Gatekeeper (Rego), Kyverno (YAML). NetworkPolicies, Calico, Cilium with Ebpf. Image scanning — Trivy, Snyk, Grype, Clair. Distroless / minimal images. Supply chain — SBOM (Syft, CycloneDX, SPDX), signing (Cosign), attestation (in-toto, SLSA levels). Runtime security — Falco rules, Tetragon Ebpf policies, Sysdig Secure. Secret detection in repos — gitleaks, trufflehog. Sealed Secrets, External Secrets Operator. CIS benchmarks for Kubernetes.
M5	**Secrets & Encryption Architecture** HashiCorp Vault — secret engines (KV, database dynamic creds, PKI, transit), auth methods (Kubernetes, JWT, AWS), policies, leases, response wrapping. AWS Secrets Manager + Parameter Store. KMS architecture — envelope encryption, CMK vs DEK, customer-managed vs AWS-managed. BYOK and HYOK for regulated industries. Key rotation policies. Encryption-at-rest — EBS, S3, RDS, Kubernetes Secrets encryption with KMS. Encryption-in-transit — TLS 1.3, Mtls, WireGuard, Tailscale. Hardware Security Modules (HSM). FIPS 140-2 / 140-3 considerations.
M6	**CSPM & Indian Regulatory Compliance** CSPM tools — Wiz, Prowler, Checkov (IaC), Cloud Custodian, Sysdig Secure, AWS Security Hub, GCP Security Command Center. CIS Benchmarks. DPDP Act 2023 — data fiduciary obligations, breach notification (72-hour), DPIA. RBI Cloud Guidelines for BFSI. MEITY Cyber Security Framework. SEBI Cybersecurity Framework for capital markets. CERT-In's 6-hour reporting requirement. Data localisation — RBI data localisation circular for payment systems. Cross-border data flow restrictions. ISO 27001:2022 mapping. SOC 2 Type II. Indian DPB-aligned controls.

Reference Books:

1. Zero Trust Networks (2nd ed) Razi Rais, Christina Morillo, Evan Gilman, Doug Barth · O'Reilly · 2024 ISBN: 978-1-098-13396-6
2. Solving Identity Management in Modern Applications (2nd ed) Yvonne Wilson, Abhishek Hingnikar · Apress · 2022 ISBN: 978-1-4842-8260-2
3. Kubernetes Security and Observability Brendan Creane, Amit Gupta · O'Reilly · 2021 ISBN: 978-1-098-10705-8
4. Container Security Liz Rice · O'Reilly · 2020 ISBN: 978-1-492-05670-6
5. Security as Code B.K. Sarthak Das, Virginia Chu · O'Reilly · 2024 ISBN: 978-1-098-12750-7
6. NIST SP 800-207: Zero Trust Architecture Scott Rose et al. · NIST · 2020 Free · csrc.nist.gov
7. Hashicorp Vault Up & Running Brian Chambers et al. · HashiCorp · 2024

RegTech & Compliance Automation

HoD-CSA 3/7/26
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

COURSE CODE: 26CA62D3E
COURSE NAME : RegTech & Compliance Automation
L-T-P-S :3-0-3-0
PRE-REQ : DBDA

CO. No	Course Outcome Description	PO	BTL
CO1	Apply digital KYC pipelines — Aadhaar eKYC (UIDAI), DigiLocker, Video KYC (RBI Jan-2020 framework), CKYC, Re-KYC triggers, periodic KYC, beneficial ownership identification (UBO).	PO1,PO4	3
CO2	Apply AML/CFT transaction monitoring — rule engines (Drools, Camunda DMN), velocity checks, structuring/layering detection, watchlist screening (OFAC, UN, RBI, SEBI debarred lists), STR/CTR generation in FIU-IND XML format.	PO1,PO4	3
CO3	Analyze regulatory reporting frameworks — RBI returns (XBRL filings, ADF, CIMS), SEBI BSE/NSE LODR filings, FIU-IND STR/CTR, GST returns (GSTR-1/3B/9), NPCI dispute reporting, Income Tax e-TDS / e-Invoicing.	PO2,PO4	4
CO4	Analyze ML-driven AML/fraud detection — supervised models (XGBoost, LightGBM), unsupervised anomaly detection (Isolation Forest, Autoencoders), graph analytics (community detection on transaction graphs with Neo4j/NetworkX), explainable AI (SHAP, LIME) for regulator-grade explanations.	PO2,PO4	4
CO5	Evaluate audit trail engineering — immutable logs (S3 object lock, AWS QLDB), evidence binders, regulator-replayable audit dashboards, board-grade compliance KPIs/KRIs aligned to RBI Cyber, SEBI Cyber, NPCI bylaws.	PO4,PO6	5
CO6	Evaluate emerging RegTech themes — CBDC (e₹) compliance flows, RBI Account Aggregator data fiduciary obligations, DPDP Act 2023 fintech impact, SupTech initiatives (RBI's CIMS / DAKSH), GIFT IFSC sandbox compliance, ESG / climate reporting (BRSR).	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Digital KYC & UBO** Aadhaar eKYC architecture — UIDAI Auth API, e-KYC API, OTP / biometric / face. Aadhaar Vault. Aadhaar masking (last 8 digits) per UIDAI compliance. DigiLocker integration. Video KYC framework (RBI Jan-2020) — geo-tag, liveness, time-stamp, randomized question. CKYC (Central KYC Registry of India). Re-KYC triggers. Periodic KYC for high-risk customers. Customer Due Diligence (CDD) and Enhanced Due Diligence (EDD). Beneficial ownership

HoD-CSA 31/1/26

Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-522502

	identification (UBO 25% rule per PMLA). PEP (politically exposed person) screening.
M2	**AML/CFT Transaction Monitoring** PMLA & FIU-IND obligations. Reporting entities. RBI Master Direction on KYC. Transaction monitoring rule engine architecture — Drools, Camunda DMN, OPA Rego. Velocity checks. Structuring & layering detection patterns. Watchlist screening — OFAC SDN, UN consolidated, RBI / FIU-IND lists, SEBI debarred entities. Fuzzy name matching (Levenshtein, Jaro-Winkler, Soundex). STR (Suspicious Transaction Report) format — FIU-IND XML schema. CTR (Cash Transaction Report > ₹10 lakh). NTR (Non-Profit Transaction Report). CCR (Counterfeit Currency Report). Suspicious Activity Report (SAR) for crypto.
M3	**Regulatory Reporting Frameworks** RBI returns — XBRL framework, ADF (Automated Data Flow), CIMS (Centralised Information Management System). RBI returns matrix — DSB returns, OSMOS, eXtensible Business Reporting Language (XBRL). SEBI LODR — BSE/NSE filings via NSEBI / BSE Listing Centre. SEBI SCORES complaints. FIU-IND STR/CTR XML submission flow. GST returns — GSTR-1 (outward), GSTR-3B (summary), GSTR-9 (annual), e-Invoice IRN, GSTR-2A reconciliation. e-TDS / TCS quarterly. Income Tax e-Invoicing (B2B threshold ₹5 cr). NPCI dispute / chargeback flows.
M4	**ML-Driven AML & Fraud Detection** Feature engineering for AML — aggregations, velocity, network features, peer benchmarks. Supervised models — XGBoost, LightGBM, CatBoost for fraud labels. Class imbalance handling — SMOTE, class weights, undersampling. Unsupervised anomaly detection — Isolation Forest, Autoencoders, One-Class SVM. Graph analytics on transaction networks — PageRank, community detection (Louvain, Leiden), suspicious-cluster identification with Neo4j or NetworkX. Explainable AI (XAI) — SHAP global / local explanations, LIME, counterfactual explanations. Model risk management (RBI MRM guidelines). Regulator-grade explanations.
M5	**Audit Trails & Board Reporting** Immutable audit-log architecture — S3 object lock + versioning + KMS encryption. AWS QLDB for cryptographic-verifiable journals. Hyperledger Fabric for shared audit ledgers. Audit-trail schema design — who/what/when/where/why/how. Evidence binders — per-customer, per-incident, per-control. Regulator-replayable audit dashboards (Streamlit, Grafana). Board-grade compliance dashboard with KPIs (TAT for STR, KYC completion %) and KRIs (false-positive rate, near-miss count). RBI Cyber Security Framework reporting. SEBI Cyber Resilience Framework. NPCI bylaws.
M6	**Emerging RegTech & SupTech** Central Bank Digital Currency (CBDC e₹) — RBI digital rupee architecture, retail vs wholesale, programmability, compliance flows. Account Aggregator (RBI 2016/2022) ecosystem — FIPs, FIUs, AAs (Setu, Finvu, OneMoney, CAMS Finserv), consent flows, data fiduciary obligations. DPDP Act 2023 fintech impact — consent, purpose limitation, retention, breach notification, DPB obligations. SupTech — RBI's CIMS, DAKSH (Daksha Autonomic Sup-Tech for Knowledge

31/7/24
HoD-CSA
Dr. K. [illegible] SAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	Hub) supervisory tech. SEBI's algorithmic trading surveillance. GIFT IFSC sandbox. ESG / BRSR (Business Responsibility & Sustainability Reporting) for top-1000 listed entities.

Reference Books:

1. RegTech, Compliance and Technology Judgement Rule Lev Bromberg, Andrew Godwin, Ian Ramsay (eds.) · Edward Elgar · 2020 ISBN: 978-1-789909-83-3
2. Anti-Money Laundering: International Law and Practice Wouter H. Muller et al. · Wiley · 2021 ISBN: 978-1-119-07598-2
3. Money Laundering: Indian Legislative and Regulatory Framework M.C. Mehanathan · Universal LexisNexis · 2022 ISBN: 978-93-93702-15-4
4. Practical Fraud Prevention: Fraud and AML Analytics for Fintech Gilit Saporta, Shoshana Maraney · O'Reilly · 2022 ISBN: 978-1-098-12046-1
5. Interpretable Machine Learning Christoph Molnar · Leanpub · 2024 Free online · christophm.github.io/interpretable-ml-book
6. Hands-On Graph Analytics with Neo4j Estelle Scifo · Packt · 2020 ISBN: 978-1-83921-261-0
7. RBI Master Directions & FIU-IND Guidelines RBI / FIU-IND · official circulars Free · rbi.org.in, fiuindia.gov.in

Business Forecasting & Demand Analytics

COURSE CODE: 26CA62E3E
COURSE NAME : Business Forecasting & Demand Analytics
L-T-P-S :3-0-3-0
PRE-REQ : AIML

CO. No	Course Outcome Description	PO	BTL
CO1	Apply classical time-series forecasting — ETS / Holt-Winters / ARIMA / SARIMA / Prophet / TBATS — using statsforecast / sktime / Prophet / pmdarima with Box-Jenkins methodology, ACF/PACF analysis, residual diagnostics, KPSS / ADF stationarity tests.	PO1,PO2	3
CO2	Apply machine-learning forecasting — LightGBM / XGBoost / CatBoost with lagged features + rolling stats + calendar features — using mlforecast, sklearn, and scikit-learn pipelines, with feature importance interpretation.	PO1,PO2	3

HoD-CSA

Dr. K. [illegible]
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO3	Analyze deep-learning forecasters — N-BEATS, NHiTS, Temporal Fusion Transformer (TFT), DeepAR (Amazon), DLinear/NLinear, PatchTST — using neuralforecast / GluonTS / PyTorch Forecasting / Darts; understand when DL beats classical (long horizons, many series, cross-learning).	PO2,PO4	4
CO4	Analyze time-series foundation models — TimeGPT (Nixtla), Chronos (Amazon), Lag-Llama, Moirai (Salesforce), TimesFM (Google) — for zero-shot / few-shot forecasting; benchmark against classical and DL baselines on M4 / M5 datasets.	PO2,PO4	4
CO5	Evaluate hierarchical reconciliation (top-down, bottom-up, MinT, OLS, WLS) for store/SKU/region hierarchies, intermittent demand (Croston, SBA, ADIDA, IMAPA, lumpy/sporadic patterns), probabilistic forecasting (quantile loss, MQRNN, conformal prediction), promotions/holiday modeling.	PO2,PO4	5
CO6	Evaluate production deployment — backtesting (rolling-origin / expanding-window / blocked CV), MLOps for forecasting (MLflow / Weights & Biases / Comet), drift detection (data drift, concept drift), model monitoring, business alignment (sMAPE / RMSSE / WAPE / Pinball loss vs business KPIs), Indian-context demand drivers (monsoon / Diwali / Onam / IPL / wedding seasons / GST changes).	PO2,PO4,PO5	5

Syllabus:

Module No.	Syllabus
M1	**Classical Time-Series Forecasting** Time-series components — trend / seasonality / cyclical / irregular. Box-Jenkins methodology — identify (ACF/PACF) / estimate / diagnose. Stationarity tests — ADF / KPSS / PP. Differencing (regular and seasonal). White-noise tests — Ljung-Box. ETS framework (Hyndman) — ANN, AAN, AAA, MAM. Holt-Winters additive/multiplicative. ARIMA / SARIMA / SARIMAX (with exogenous). Auto-ARIMA via pmdarima. Prophet (Facebook) — trend changepoints, seasonalities, holidays. TBATS for complex seasonality (multiple seasonal periods). statsforecast (Nixtla, > 100x faster than statsmodels).

Dr. HoD-CSA GAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

M2	**ML Forecasting** Reformulating forecasting as supervised regression. Lagged features (lag_1, lag_7, lag_30 for daily data). Rolling statistics (rolling_mean, rolling_std, rolling_max). Calendar features (day_of_week, month, quarter, is_weekend, is_holiday). Indian-context features — festival_calendar (Diwali, Holi, Onam, Eid, Christmas), monsoon_phase, ipl_match_day, wedding_season. Encoding strategies — mean encoding, target encoding with KFold to avoid leakage. mlforecast library. LightGBM hyperparameter tuning for forecasting. Quantile regression forests. Direct vs recursive multi-step forecasting.
M3	**Deep-Learning Forecasting** When DL beats classical — long horizons, cross-series learning (10K+ series), nonlinear interactions, exogenous covariates. Recurrent baseline — LSTM / GRU. DeepAR (Amazon, autoregressive RNN with global parameters across many series). N-BEATS (Bengio et al., MLPs with stacks). NHiTS (extension with multi-rate signal sampling). Temporal Fusion Transformer (TFT, Google) — attention-based with variable-selection networks + LSTM encoder. DLinear / NLinear (simple linear baselines that beat transformers on many benchmarks — Zeng et al. 2022). PatchTST (patching + transformer). neuralforecast (Nixtla). GluonTS (Amazon). PyTorch Forecasting. Darts. Choosing architecture for use-case.
M4	**Time-Series Foundation Models** Foundation-model paradigm for time series — pretrain on massive corpus, zero-shot / few-shot inference. TimeGPT (Nixtla, paid API, zero-shot strong on M4). Chronos (Amazon, open-source, tokenizes time-series as language). Lag-Llama (open, Llama-style architecture). Moirai (Salesforce, 1B params, multi-frequency). TimesFM (Google, 200M params). When zero-shot beats fitting a model (small datasets, cold-start SKUs, exploratory analysis); when fitting still wins (long history with stable patterns). Benchmarking foundation models against classical baselines on M4 / M5 / M6 / Tourism / hospital datasets. Cost / latency considerations.
M5	**Hierarchical, Intermittent & Probabilistic Forecasting** Hierarchical forecasting — group structures (region / city / store / SKU). Reconciliation methods — bottom-up, top-down (proportional / Gross-Sohl), middle-out, MinT (minimum trace), OLS, WLS, ERM. hierarchicalforecast (Nixtla). Intermittent demand patterns — smooth / erratic / lumpy / intermittent (Syntetos-Boylan classification). Croston, SBA (Syntetos-Boylan approximation), ADIDA, IMAPA (intermittent multiple aggregation prediction algorithm). Probabilistic forecasting — pinball loss, quantile regression, MQRNN. Conformal prediction for any forecaster (split conformal, conformal prediction intervals). Promotions / holiday modeling with regressors / dummy variables / event embeddings.
M6	**Production Forecasting & Indian Context** Backtesting strategies — rolling-origin / expanding-window / blocked CV / time-series CV. Production metrics — sMAPE, MAPE, MASE, RMSSE (M5 winner metric), WAPE (used by Walmart), Pinball loss, CRPS. Aligning metrics to business — over-forecast cost vs under-forecast cost / asymmetric loss. MLOps stack — MLflow tracking, Weights & Biases, Comet, Neptune. Model registry. Drift detection — data drift (KS / PSI), concept drift (model error distribution shift). Indian context — monsoon (Jun-Sep), Diwali (Oct/Nov), Onam (Aug/Sep), wedding seasons (Nov-Feb), IPL (Mar-May), Republic Day / Independence Day sales, GST rate changes.

31/3/26
HoD-CSA

Dr. K. [illegible]
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Reference Books:

1. Forecasting: Principles and Practice (3rd ed) Rob J. Hyndman, George Athanasopoulos · OTexts · 2021 Free online · otexts.com/fpp3
2. Modern Time Series Forecasting with Python (2nd ed) Manu Joseph · Packt · 2024 ISBN: 978-1-83588-637-1
3. Time Series Analysis: Forecasting and Control (5th ed) George Box, Gwilym Jenkins, Gregory Reinsel, Greta Ljung · Wiley · 2015 ISBN: 978-1-118-67502-9
4. Practical Time Series Analysis Aileen Nielsen · O'Reilly · 2019 ISBN: 978-1-492-04165-8
5. Forecasting with Exponential Smoothing Hyndman, Koehler, Ord, Snyder · Springer · 2008 ISBN: 978-3-540-71916-8
6. Demand Forecasting and Inventory Control A. Eric Schweitzer · Routledge · 2020 ISBN: 978-0-367-37301-2
7. Supply Chain Analytics & Modelling: Quantitative Tools and Solutions (Indian context) Nicoleta Tipi · Kogan Page · 2021 ISBN: 978-1-789-66579-2

Intelligent Document Processing & RPA

COURSE CODE: 26CA63A4E
COURSE NAME : Intelligent Document Processing & RPA
L-T-P-S :3-0-3-0
PRE-REQ : ADRG

CO. No	Course Outcome Description	PO	BTL
CO1	Apply OCR & document-AI tools (Tesseract, PaddleOCR, AWS Textract, Azure Document Intelligence, Google Document AI, Donut, LayoutLMv3) for structured extraction from forms, invoices, contracts.	PO1,PO4	3
CO2	Apply VLMs (Claude vision, GPT-4o vision, Mistral-OCR) for direct document Q&A and entity extraction without intermediate OCR.	PO1,PO4	3

31/7/26
HoD-CSA
Dr. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K L Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO3	Analyze classical RPA platforms (UiPath, Automation Anywhere, Power Automate) and AI-native browser/computer-use automation (Anthropic Computer Use, browser-use, BrowserBase) for workflow automation.	PO2,PO4	4
CO4	Analyze workflow orchestration patterns — queues, retries, dead-letter handling, human-in-the-loop checkpoints, audit logging — using Temporal, Inngest, or Camunda.	PO2,PO6	4
CO5	Apply Indian-context document AI — Aadhaar, PAN, GST invoices, bank statements, healthcare reports — using IndicOCR + AI4Bharat models.	PO1,PO3	3
CO6	Evaluate IDP/RPA systems — field-level accuracy, end-to-end success rate, exception rate, cost per document, regulatory compliance — for production deployments.	PO2,PO5,PO6	5

Syllabus:

Module No.	Syllabus
M1	**OCR & Document AI Foundations** Document types — structured (forms with fixed fields), semi-structured (invoices with variable layout), unstructured (contracts, emails). OCR options — Tesseract, PaddleOCR, EasyOCR. Cloud OCR — AWS Textract (with Forms & Tables), Azure Document Intelligence (prebuilt + custom), Google Document AI. LayoutLMv3 architecture — multimodal pretraining on document images. Donut as OCR-free alternative. Marker for PDF→Markdown. Practical: invoice extraction with Donut + cloud comparison.
M2	**VLM-Based Document Understanding** VLMs as OCR replacement — Claude 3.5 Sonnet vision, GPT-4o vision, Gemini Pro Vision, Mistral-OCR. Cost-quality trade-offs vs traditional OCR. Direct document Q&A patterns. Schema-driven extraction with structured outputs (JSON mode, function calling). Multi-page document handling. Confidence scoring from VLMs. Self-hosted alternatives (LLaVA-Next, Qwen2-VL, InternVL). Hybrid pipelines (OCR for layout + VLM for understanding).
M3	**Classical RPA & Modern Automation** Classical RPA platforms — UiPath Studio + Orchestrator, Automation Anywhere A360, Microsoft Power Automate Desktop. Selectors, recorders, attended vs unattended bots. Queueing (UiPath Queues), credentials management (UiPath Asset Manager, Orchestrator vault). Modern AI-native automation — Anthropic Computer Use, browser-use library, BrowserBase, Stagehand. Playwright + LLM agent patterns. When to choose classical vs AI-native (predictable repetitive vs adaptive).
M4	**Workflow Orchestration** Why workflow engines — reliability, retries, idempotency, observability. Temporal — activities, workflows, signals, timers. Inngest as serverless alternative. Camunda for BPMN-modeled processes. Queue patterns — FIFO, priority, dead-letter, redrive. Human-in-the-loop checkpoints — pausing workflow for approval,

Dr. K. [illegible] 31/7/26
HoD-CSA
HOD-Computer Science & Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur Dist., Andhra Pradesh-522502.

	escalation. Audit logging schema — who did what when. SLA monitoring. Compensation patterns for failed workflows.
M5	**Indian-Context Documents** Indian document types — Aadhaar (with masking for compliance), PAN card, GSTIN-bearing invoices, bank statements (variability across banks), CIBIL reports, ITR forms, healthcare reports. Indic-script OCR — IndicOCR (AI4Bharat), Tesseract Indic, PaddleOCR multilingual. KYC pipelines — document verification + face match + liveness. UIDAI APIs (eKYC, offline Aadhaar XML). DigiLocker integration. RBI/SEBI document compliance. Account Aggregator framework basics.
M6	**IDP/RPA Production & Compliance** Field-level accuracy measurement — precision, recall, F1 per field. End-to-end success rate. Straight-through-processing (STP) percentage as the north-star metric. Exception handling routing. Cost per document model (OCR + VLM + RPA license). Bot lifecycle management — staging, production, version control. Regulatory considerations — DPDP Act 2023, RBI data localization, GDPR equivalents. Audit-trail design. Process mining basics (Celonis, UiPath Process Mining) to discover automation candidates.

Reference Books:

1. Robotic Process Automation Projects Nandan Mullakara, Arun Kumar Asokan · Packt · 2020 ISBN: 978-1-83921-973-2
2. Hyperautomation Matt Calkins · Wiley · 2020 ISBN: 978-1-119-81306-3
3. Modern Computer Vision with PyTorch (2nd ed) Ayyadevara, Reddy · Packt · 2024 ISBN: 978-1-80323-928-6
4. Designing Distributed Systems Brendan Burns · O'Reilly · 2018 ISBN: 978-1-491-98364-5
5. Building Event-Driven Microservices Adam Bellemare · O'Reilly · 2020 ISBN: 978-1-492-05789-5
6. Practical Process Automation Bernd Ruecker · O'Reilly · 2021 ISBN: 978-1-492-06145-8

Serverless, Edge & FaaS Computing

COURSE CODE: 26CA63B4E
COURSE NAME : Serverless, Edge & FaaS Computing
L-T-P-S :3-0-3-0
PRE-REQ : CDVO

CO. No	Course Outcome Description	PO	BTL

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

CO1	Apply Function-as-a-Service (FaaS) platforms — AWS Lambda, GCP Cloud Functions / Cloud Run, Azure Functions, Cloudflare Workers, Vercel Functions, Deno Deploy — for event-driven workloads.	PO1,PO4	3
CO2	Analyze FaaS internals — cold-start mechanics, container/microVM (Firecracker), V8 isolates (Cloudflare), provisioned concurrency, cold-start mitigation strategies.	PO2,PO4	4
CO3	Analyze edge-computing primitives — CDN compute (Workers, Lambda@Edge, Fastly Compute@Edge), edge KV stores (Cloudflare KV, D1, R2), edge inference, edge-database replication.	PO2,PO4	4
CO4	Apply serverless orchestration — AWS Step Functions, GCP Workflows, durable-functions patterns, fan-out/fan-in patterns, error-handling and retries.	PO1,PO2	3
CO5	Evaluate serverless data — DynamoDB, FaunaDB, PlanetScale, Turso (libSQL), Cloudflare D1, Upstash Redis — for globally-replicated, serverless-friendly storage.	PO2,PO4	5
CO6	Evaluate when to use serverless vs containers vs traditional compute — cost models, lock-in considerations, scale-to-zero benefits, observability gaps, performance trade-offs.	PO2,PO3,PO6	5

Syllabus:

Module No.	Syllabus
M1	**FaaS Foundations** Serverless principles — pay-per-execution, scale-to-zero, no-server-management. AWS Lambda — runtimes, layers, concurrency limits. Lambda triggers (S3, DynamoDB Streams, EventBridge, API Gateway, Function URL). Memory-CPU trade-off (Lambda Power Tuner). GCP Cloud Functions vs Cloud Run (request-based scaling). Azure Functions. Cloudflare Workers (V8 isolates not containers). Vercel Functions / Edge Functions. Deno Deploy. Bun runtime emergence. Comparing FaaS options — cold-start, runtime support, max execution time, pricing.
M2	**FaaS Internals & Cold Starts** Container-based FaaS (Lambda, Cloud Functions Gen 2) — how it boots. Firecracker microVMs (the AWS Lambda magic). V8 isolates as alternative model (Cloudflare Workers, Deno). Cold start anatomy — init phase, runtime phase,

HoD-CSA 31/1/26

Dr. K. [illegible]
HOD-Comp. Sc. & Applns.
Koneru Lakshmaiah Education Foundation
(Deemed to be University)
Green Fields, VADDESWARAM
Guntur District, Andhra Pradesh-522

	handler phase. Cold-start mitigation — provisioned concurrency, Lambda SnapStart (Java), keeping warm. Bundle size & tree-shaking impact on cold starts. Custom runtimes. Container image-based Lambda. Bursty traffic patterns. Cost & latency analysis — when serverless beats containers.
M3	**Edge Computing** CDN evolution — from cache to compute. Cloudflare Workers + KV (eventually consistent) + R2 (S3-compatible) + D1 (SQLite). Lambda@Edge / CloudFront Functions. Fastly Compute@Edge (WebAssembly-based). Edge inference — running ML models at edge (Workers AI, Vercel AI SDK with edge-compatible models). Edge databases — PlanetScale, Turso (libSQL), Cloudflare D1 globally distributed. Geo-routing. Edge auth (JWT verification at edge). Static asset optimization at edge. Indian-context: low-latency for tier-2/tier-3 cities via edge POPs.
M4	**Serverless Orchestration** Why orchestration — multi-step processes need durability. AWS Step Functions — state machine model, Standard vs Express. ASL (Amazon States Language). Step Functions patterns — sequential, parallel, choice, fan-out/fan-in, map. GCP Workflows. Azure Durable Functions (orchestration with code). Inngest as a serverless durable-execution platform. Temporal Cloud as a serverless workflow option. Cloudflare Durable Objects for stateful coordination. Saga patterns in serverless. Error handling — retries, catch, fallback states.
M5	**Serverless Data** DynamoDB — partition-key design, GSI, single-table design pattern, on-demand vs provisioned. FaunaDB — transactional NoSQL with global consistency. PlanetScale — serverless MySQL with branching. Turso / libSQL — SQLite at edge with embedded replicas. Cloudflare D1. Upstash Redis — serverless Redis (REST + Redis protocol). Neon — serverless Postgres. Cost models — per-request vs always-on. Connection pooling problem in serverless — RDS Proxy, Neon's pooling, Supabase pooler. Read-replica strategies for global reads.
M6	**Serverless Trade-offs & Production** When serverless wins — bursty traffic, low average load, fast time-to-market, ops minimization. When containers win — long-running, predictable load, complex stateful systems, fine-grained network control. Hybrid architectures (serverless front-doors with container backends). Vendor lock-in evaluation. SST (Serverless Stack), Pulumi, AWS CDK as IaC. Observability gaps in serverless — X-Ray, Lambda Insights, Datadog Serverless. Security — IAM roles per function, KMS for secrets. Cost optimization — right-sizing memory, reducing invocation count, batching. Multi-region serverless deployments.

Reference Books:

1. Serverless Architectures on AWS (2nd ed) Peter Sbarski, Yan Cui, Ajay Nair · Manning · 2022 ISBN: 978-1-617-29582-9
2. Programming AWS Lambda John Chapin, Mike Roberts · O'Reilly · 2020 ISBN: 978-1-492-04105-4
3. The DynamoDB Book Alex DeBrie · self-published · 2020 dynamodbbook.com

HoD-CSA 31/7/26
Dr. K. BHARGAVA
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

4. Building Serverless Applications with Python Jalem Raj Rohit · Packt · 2018 ISBN: 978-1-78712-503-3
5. Cloudflare Workers Documentation (book-length) Cloudflare · workers.cloudflare.com/docs
6. AWS Step Functions Developer Guide (book-length) AWS · docs.aws.amazon.com Free

Security Governance, Risk & Compliance (GRC)

COURSE CODE: 26CA63C4E
COURSE NAME : Security Governance, Risk & Compliance (GRC)
L-T-P-S :3-0-3-0
PRE-REQ : OFSC

CO. No	Course Outcome Description	PO	BTL
CO1	Apply security governance frameworks — ISO 27001:2022, NIST CSF 2.0, COBIT 2019, SOC 2 (Trust Services Criteria) — including policy authoring, control mapping, and ISMS scoping.	PO1,PO6	3
CO2	Analyze risk management methodologies — ISO 31000, NIST SP 800-30/37/39, FAIR — and build quantitative + qualitative risk registers with realistic threat events.	PO2,PO6	4
CO3	Analyze compliance regimes relevant to Indian industries — DPDP Act 2023, IT Rules 2011/2021, RBI Cyber Security Framework for banks/NBFCs, SEBI cybersecurity, IRDAI guidelines, MEITY framework, CERT-In directions.	PO2,PO6	4
CO4	Analyze global compliance regimes — GDPR, HIPAA, PCI-DSS v4.0, SOX — and the gaps / overlaps with Indian regulations relevant to Indian IT-services exporters.	PO2,PO6	4
CO5	Evaluate audit lifecycle — readiness assessment, gap analysis, remediation, evidence collection, internal audit, external audit, certification — using GRC platforms (Drata, Vanta, Sprinto, Trustero) and traditional spreadsheets.	PO4,PO6	5
CO6	Evaluate incident response, business continuity (BCP) & disaster recovery (DR), third-party risk management (TPRM), cyber insurance, board reporting, and KPI/KRI dashboards aligned to NIST CSF 2.0 outcomes.	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Governance Frameworks & Policy Authoring** ISO 27001:2022 structure — clauses 4-10 + Annex A 93 controls. NIST CSF 2.0 — Govern, Identify, Protect, Detect, Respond, Recover. COBIT 2019 governance vs management. SOC 2 Trust Services Criteria (security, availability, confidentiality, processing integrity, privacy). Policy hierarchy — charter, policy, standard, procedure, guideline. Authoring an ISMS scope statement, information security

31/7/25
HoD-CSA
Dr. K. Bhagavan
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Andhra Pradesh-522502.

	policy, acceptable use policy, access control policy. Document control. Statement of Applicability (SoA).
M2	**Risk Management & Risk Registers** SO 31000 risk management framework. NIST SP 800-30 (risk assessment), 800-37 (RMF), 800-39 (organization-wide). FAIR (Factor Analysis of Information Risk) for quantitative risk. Threat sources, threat events, vulnerabilities, likelihood, impact. Risk treatment options — mitigate, accept, transfer, avoid. Risk register structure. Heat maps. Inherent vs residual risk. Risk appetite vs risk tolerance. Linking risk to controls. Top-10 risks for an Indian SaaS startup.
M3	**Indian Regulatory Landscape** DPDP Act 2023 deep-dive — data principal rights, consent manager, data fiduciary obligations, breach reporting (72-hour to DPB, 'as soon as possible' to data principals), DPIA, significant data fiduciary, cross-border transfer, age-related provisions. IT Rules 2011 (sensitive personal data), IT Rules 2021 (intermediary, social media), IT Act 2000. CERT-In April-2022 directions — 6-hour incident reporting, 180-day log retention. RBI Cyber Security Framework for SCBs, UCBs, NBFCs, payment systems. SEBI cybersecurity for capital markets. IRDAI for insurance. MEITY Cyber Security Framework.
M4	**Global Frameworks for Indian Exporters** GDPR (EU) — lawful basis, data subject rights, DPIA, breach 72-hour, fines up to 4% global turnover. Comparison with DPDP Act 2023. HIPAA (US healthcare) — covered entities, business associates, safeguards (admin/physical/technical), breach notification. PCI-DSS v4.0 — 12 requirements, levels 1-4, ROC/AOC. SOX (US public companies) — ITGC controls. CCPA / CPRA. NIS2 Directive (EU). Cross-border data transfer mechanisms — SCCs, BCRs, adequacy decisions. Data localisation circulars (RBI 2018, 2020).
M5	**Audit Lifecycle & GRC Platforms** Audit lifecycle — readiness, gap analysis, remediation, evidence collection, internal audit, Type 1 vs Type 2, certification cycle (3-year ISMS, annual SOC 2). Evidence types — configuration, screenshots, logs, attestations, walkthroughs. ISMS internal audit process. Management review. CAR (corrective action requests). GRC platforms — Drata, Vanta, Sprinto (Indian), Trustero, Secureframe. Continuous compliance — automated evidence collection. Big-4 audit walkthrough patterns — KPMG, PwC, Deloitte, EY. ISO 27001 vs SOC 2 cost / time comparison.
M6	**Incident Response, BCP/DR, TPRM & Reporting** Incident response lifecycle — preparation, identification, containment, eradication, recovery, lessons learned. Tabletop exercises. CERT-In incident reporting workflow. Business Continuity (BCP) — BIA, RTO, RPO. Disaster Recovery (DR) — pilot light, warm standby, multi-site active. Third-Party Risk Management (TPRM) — vendor due diligence questionnaires (SIG Lite, CAIQ), continuous monitoring (BitSight, SecurityScorecard). Cyber insurance. CISO board reporting — KPIs, KRIs, NIST CSF outcome scoring. Quarterly dashboard.

Reference Books:

HoD-CSA 31/3/26
Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

1. ISO/IEC 27001:2022 Standard + Annex A Controls ISO official Available via BIS India ($120)
2. Building an Effective Information Security Policy Architecture Sandy Bacik · CRC Press · 2008 ISBN: 978-1-4200-5905-9
3. Measuring and Managing Information Risk: A FAIR Approach Jack Freund, Jack Jones · Butterworth-Heinemann · 2014 ISBN: 978-0-12-420231-3
4. Privacy and Data Protection Law in India (DPDP Act 2023) Stephen Mathias, Naveen Hegde · LexisNexis · 2024 ISBN: 978-93-91211-39-5
5. CISA Review Manual (28th ed) ISACA · 2024 ISBN: 978-1-60420-994-9
6. CISSP Official (ISC)² Study Guide (10th ed) Mike Chapple, James M. Stewart, Darril Gibson · Wiley · 2024 ISBN: 978-1-394-25469-9
7. NIST CSF 2.0 + NIST SP 800-30/37/39 NIST · 2024 Free · csrc.nist.gov

Enterprise Distributed Ledger & Layer-2 Solutions

COURSE CODE: 26CA63D4E
COURSE NAME : Enterprise Distributed Ledger & Layer-2 Solutions
L-T-P-S :3-0-3-0
PRE-REQ : BCSC

CO. No	Course Outcome Description	PO	BTL
CO1	Apply Hyperledger Fabric 2.x — channels, ordering service (Raft), MSP, chaincode (Go / Node.js), endorsement policies, private data collections, identity mixer, gateway SDK — for permissioned enterprise networks.	PO1,PO4	3
CO2	Apply alternative enterprise DLTs — R3 Corda (notary, flows, contracts in Kotlin), Quorum / Hyperledger Besu (private Ethereum), Hyperledger Indy / Aries (decentralized identity), DAML smart contracts — with deployment patterns.	PO1,PO4	3
CO3	Analyze Ethereum Layer-2 architectures — optimistic rollups (Arbitrum Nitro, Optimism Bedrock, Base), ZK rollups (zkSync Era, StarkNet, Linea, Scroll, Polygon zkEVM), validiums, plasma, sidechains.	PO2,PO4	4
CO4	Analyze modular blockchain architecture — Celestia data availability, EigenLayer restaking, AltLayer rollup-as-a-service,	PO2,PO6	4

Dr. K BHAGAVAN
HOD-Computer Science and Applications
HoD-CSA
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	Polygon CDK, OP Stack, Arbitrum Orbit — for app-specific rollups.		
CO5	Evaluate cross-chain interoperability — LayerZero, Wormhole, Axelar, Chainlink CCIP, Hyperlane, IBC (Cosmos) — security trade-offs (lock-and-mint vs burn-and-mint vs liquidity pools), bridge security incidents (Ronin, Wormhole, Nomad), and mitigation patterns.	PO2,PO4	5
CO6	Evaluate enterprise blockchain use-cases — trade finance (Marco Polo, we.trade), supply chain (IBM Food Trust, TradeLens), CBDC pilots (RBI e₹, Project Dunbar), tokenized securities (HSBC Orion, JPMorgan Onyx), Indian enterprise blockchain (NPCI Vajra, IndiaChain).	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Hyperledger Fabric Production** Fabric 2.x architecture — peers, orderers (Raft), MSP, channels, system channel deprecation, gossip protocol. Chaincode in Go / Node.js / Java — lifecycle (init, invoke, query). Endorsement policies (AND, OR, OutOf). Private Data Collections (PDC). Identity Mixer (Idemix) for selective disclosure. Gateway SDK (v2.4+) — submitTransaction, evaluateTransaction. CouchDB rich queries. CA setup with Fabric CA. TLS & mutual TLS. Production deployment with Kubernetes (operator pattern, Hyperledger Fabric Operator).
M2	**Alternative Enterprise DLTs** R3 Corda — states, contracts (Kotlin), flows, notary, network map. Strict UTXO model and 'need-to-know' visibility. Quorum / Hyperledger Besu — private Ethereum, IBFT 2.0, QBFT consensus, private transactions (Tessera). DAML — smart contracts with rights/obligations model. Hyperledger Indy — decentralized identity (DIDs, verifiable credentials). Hyperledger Aries — agent framework. SSI (self-sovereign identity) patterns. ION (Microsoft DID on Bitcoin). When to use which DLT — permissioned vs permissionless decision matrix.
M3	**Ethereum L2 Architectures** L1 vs L2 mental model. Optimistic rollups — Arbitrum Nitro internals (geth fork, fraud proofs, challenge period 7 days), Optimism Bedrock (modular, op-stack, output proposals), Base (Coinbase's L2 on OP Stack). ZK rollups — zkSync Era (LLVM, custom VM), StarkNet (Cairo, STARK proofs), Linea (zkEVM with Truncated Type-2), Scroll (zkEVM Type-2), Polygon zkEVM. Validiums (data off-chain, ZK proof on-chain). Plasma (legacy). Sidechains — Polygon PoS, Gnosis Chain, Ronin. Sequencer decentralization debate. EIP-4844 proto-danksharding & blobs.
M4	**Modular Blockchains & App-Specific Rollups** Modular thesis — execution / settlement / consensus / DA layers. Celestia data availability layer with namespaced merkle trees. EigenLayer restaking primitive. AltLayer rollup-as-a-service. Polygon CDK (Chain Development Kit) — deploy zkEVM L2 in <1 hour. OP Stack — Optimism's open framework powering Base.

31/3/26
HoD-CSA

Dr. [illegible]
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225[illegible]

	Arbitrum Orbit — deploy custom Arbitrum chain. App-specific rollup design — choose DA layer, settlement layer, sequencer model. Sovereign vs settled rollup. Real examples — dYdX V4 (Cosmos app-chain), Aevo (OP Stack), zkLink (zkEVM hyperchain).
M5	**Cross-Chain Interoperability** Bridge architectures — lock-and-mint, burn-and-mint, liquidity-pool. LayerZero V2 — ULNs, decentralized verifier networks. Wormhole — guardians, message verification. Axelar — PoS validator network. Chainlink CCIP — risk management network. Hyperlane — permissionless interop. IBC (Cosmos) — canonical interop standard. Bridge security incidents post-mortems — Ronin ($625M), Wormhole ($325M), Nomad ($190M), Harmony ($100M). Mitigation patterns — multi-message verification, rate limits, optimistic verification with fraud window, ZK-bridge approaches (zkBridge, Polyhedra). Account abstraction (ERC-4337) for cross-chain UX.
M6	**Enterprise Use-Cases & Indian Context** Trade finance — Marco Polo, we.trade (post-mortems), Contour, Komgo. Supply chain — IBM Food Trust, TradeLens (shut down post-mortem), Maersk-IBM. CBDC pilots — RBI e₹ retail / wholesale, Project Dunbar (BIS / Singapore / Australia / Malaysia / South Africa). Tokenized securities — HSBC Orion (digital bond), JPMorgan Onyx, Goldman GS DAP, Singapore MAS Project Guardian. Indian enterprise blockchain — NPCI Vajra (cheque clearing), IndiaChain (NITI Aayog), KYC consortium (banks), GSTN blockchain pilot. Land records (Telangana / Andhra Pradesh blockchain pilots). RBI digital rupee architecture & programmability.

Reference Books:

1. Hands-On Smart Contract Development with Hyperledger Fabric V2 Matt Zand, Xun (Brian) Wu, Mark Anthony Morris · O'Reilly · 2021 ISBN: 978-1-492-08612-3
2. Mastering Blockchain (4th ed) Imran Bashir · Packt · 2023 ISBN: 978-1-80324-666-0
3. The R3 Corda Developer Guide R3 official documentation Free · docs.r3.com
4. Self-Sovereign Identity (SSI) Alex Preukschat, Drummond Reed · Manning · 2021 ISBN: 978-1-617-29659-8
5. L2BEAT Documentation & Research l2beat.com Free · canonical L2 reference
6. The Modular Blockchain Thesis Celestia Foundation · 2024 Free · celestia.org/learn
7. Blockchain in Indian Banking and Finance RBI Working Papers + Sahamati / IDRBT publications Free · rbi.org.in, idrbt.ac.in

Customer Growth & Behavioral Analytics

COURSE CODE: 26CA63E4E
COURSE NAME : Customer Growth & Behavioral Analytics

HoD-CSA 31/01/26
Dr. K. [illegible] GAVAN
HOD-Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-52250

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

L-T-P-S :3-0-3-0
PRE-REQ : DSAV

CO. No	Course Outcome Description	PO	BTL
CO1	Apply cohort analysis and retention curves — classic, stratified by acquisition channel / geography / device / persona; D7 / D14 / D30 retention; rolling retention vs classic retention; Sean Ellis growth equation.	PO1,PO2	3
CO2	Apply LTV (Lifetime Value) modeling — historical (cohort-based), predictive (BG/NBD + Gamma-Gamma via lifetimes library, BTYDplus), neural CLV, contractual vs non-contractual settings, payback period, LTV/CAC ratio.	PO1,PO4	3
CO3	Analyze churn prediction — logistic regression / XGBoost / survival analysis (Cox PH, Kaplan-Meier), feature engineering for behavioral signals, class imbalance, calibration (Platt scaling, isotonic), threshold optimization, business action mapping.	PO2,PO4	4
CO4	Analyze segmentation — RFM (Recency / Frequency / Monetary), K-means / Gaussian Mixture / hierarchical / DBSCAN, behavior-based segmentation with embeddings, latent-class models, persona development, ICP (Ideal Customer Profile) analytics.	PO2,PO4	4
CO5	Evaluate marketing attribution (last-touch / first-touch / linear / time-decay / position-based / data-driven Markov / Shapley-value), Marketing Mix Modeling (MMM with PyMC + Robyn-style geo + saturation curves), incrementality testing (CausalImpact, GeoLift) in cookieless world.	PO2,PO4	5
CO6	Evaluate product analytics — event taxonomy (Mixpanel / Amplitude / PostHog), funnel analysis, feature adoption, North Star metric design, AARRR pirate metrics, growth loops, viral coefficient (k-factor), Net Revenue Retention (NRR), Indian context (DPDP consent, server-side tracking).	PO2,PO4,PO6	5

Syllabus:

Module No.	Syllabus
M1	**Cohort Analysis & Retention** Cohort definition — acquisition cohorts (signup_date, first_purchase_date), behavioral cohorts (used_feature_X). Classic retention — D1 / D7 / D14 / D30. Rolling retention vs classic retention. N-day retention / unbounded retention. Stratification — by channel (organic, paid, referral), geo, device, demographic. Smile curve vs decay curve. Sean Ellis growth-equation framework. Reactivation / dormant / power-user definitions. Cohort size effects, week-over-week vs YoY comparisons. Pandas / DuckDB cohort-table generation patterns.
M2	**LTV & Customer Value Models** Historical LTV — cumulative ARPU per cohort. Predictive LTV — contractual setting (subscription) vs non-contractual (D2C). BG/NBD model — Beta-Geometric/Negative-Binomial-Distribution for purchase frequency. Gamma-

31/7/26
HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

	Gamma model for monetary value. lifetimes library + BTYDplus. Pareto/NBD alternative. Neural CLV (Wang & Jiang 2018). LTV/CAC ratio — SaaS rule of thumb 3:1, D2C 1.5-2:1. Payback period. Channel-level LTV. Indian D2C benchmarks — ~₹800-2000 average LTV for beauty / wellness, ~₹3000-8000 for fintech, ~₹500-1500 for edtech.
M3	**Churn Prediction & Survival Analysis** Churn definitions — voluntary vs involuntary, hard churn (cancel) vs soft churn (no activity in N days). Feature engineering for churn — recency / frequency / engagement / support tickets / NPS. Logistic regression / XGBoost / LightGBM baselines. Survival analysis — Kaplan-Meier curves, Cox Proportional Hazards model, Accelerated Failure Time (AFT) models, lifelines library. Class imbalance — SMOTE, undersampling, class weights. Calibration — Platt scaling, isotonic regression, sklearn CalibratedClassifierCV. Threshold optimization — expected-utility maximization. Mapping churn risk to retention actions (offer, win-back email, app push).
M4	**Segmentation & Personas** RFM segmentation — quintile / decile bucketing, RFM score, action mapping (Champions / Loyal / At Risk / Hibernating / Lost). K-means + elbow / silhouette. Gaussian Mixture Models (GMM). Hierarchical clustering for tree-shaped segments. DBSCAN for non-spherical clusters. Behavior-based segmentation with sentence-transformers embeddings on event sequences. Latent-class models. Persona development — combining quantitative segments with qualitative interviews. Ideal Customer Profile (ICP) — B2B SaaS-specific. Cohort × segment cross-tab analyses.
M5	**Attribution & Marketing Mix Modeling** Multi-touch attribution — last-touch / first-touch / linear / time-decay / position-based (U-shape, W-shape). Data-driven attribution — Markov chains (transition probabilities, removal effect), Shapley value (game-theoretic fair attribution). Marketing Mix Modeling (MMM) — PyMC-Marketing (Bayesian MMM), Robyn (Meta's Nevergrad-based), LightweightMMM (Google JAX-based). Adstock / saturation / carryover transformations. Hill / S-curve saturation. Geographic MMM. Incrementality — CausalImpact, GeoLift. Cookieless world / Apple ATT / GDPR / DPDP impact on attribution. Server-side tagging via GTM SS / Stape.
M6	**Product Analytics & Growth** Event taxonomy design — entity-action-property pattern (User Signed Up, Order Placed). Tools — Mixpanel, Amplitude, PostHog (open-source, deployable on-prem/India region). Funnel analysis — conversion / drop-off / time-to-convert. Feature adoption — engagement curves, time-to-first-value. North Star metric design — weekly active users (Slack), nights booked (Airbnb), gross merchandise volume (Amazon). AARRR pirate metrics framework (Acquisition / Activation / Retention / Referral / Revenue). Growth loops — viral, content, paid. Viral coefficient (k-factor). Net Revenue Retention (NRR), Gross Revenue Retention (GRR). Indian context — DPDP consent management, server-side tracking, SDK weight on Indian Android devices.

Reference Books:

1. Lean Analytics Alistair Croll, Benjamin Yoskovitz · O'Reilly · 2013 ISBN: 978-1-449-33567-1

31/7/26
HoD-CSA
Dr. K. SRAVAN
HOD Computer Science and Application
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

2. Hacking Growth Sean Ellis, Morgan Brown · Currency · 2017 ISBN: 978-0-451-49721-4
3. Customer Analytics For Dummies Jeff Sauro · Wiley · 2014 ISBN: 978-1-118-93724-0
4. Probability Models for Customer-Base Analysis Peter S. Fader, Bruce G.S. Hardie · Wharton · canonical papers Free · brucehardie.com · Coverage: M2 — BG/NBD + Gamma-Gamma canonical
5. Survival Analysis: A Self-Learning Text (3rd ed) David G. Kleinbaum, Mitchel Klein · Springer · 2012 ISBN: 978-1-4419-6645-2
6. Marketing Analytics: A Practical Guide to Improving Consumer Insights Using Data Techniques (3rd ed) Mike Grigsby · Kogan Page · 2024 ISBN: 978-1-398-61454-9
7. Marketing Analytics for Indian SMEs Mukul Pandya, Vinay R. Singh · Sage India · 2022 ISBN: 978-93-91138-15-9

3/1/26
HoD-CSA
Dr. K. [illegible]
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2578129

Department of Computer Science & Applications

Minutes of the 14th DAC meeting held on 15-06-2026 at 4.00 P.M
VENUE : S518

Agenda Items:

1. To review the existing structures of BCA and MCA.
2. To review the various courses under specific domains.
3. To discuss the merits and demerits of Trimester and Regular.
4. To analyze the current Specializations of BCA and MCA.
5. To review the CO and PO attainments of 2025-26.

This meeting is chaired by Dr.K.Bhagavan, Chair Person, HOD, Department of CSA

Members:

S.No	Name of the Member	Designation / Role in Committee
1	Dr. K. Bhagavan	Chairperson
2	Dr.K.Kiran Kumar	Professor
3	Dr. C.M.Sheela Rani	Professor
4	Dr.V.Valli Mayil	Professor
5	Dr.MNV Kiran Babu	Assoc. Professor
6	Dr. B.Mouleswararao	Professor
7	Dr. V.S Narayana Tinnaluri	Assoc. Professor
8	Dr. Y.Rama Krishna	Asst. Professor

31/7/26

HoD-CSA

Dr. K. BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002, Ph: +91 - 866 - 3500122, 2576129

9	Dr. Chunduri Kiran Kumar	Asst. Professor
10	Mr. A.Siddhartha Reddy	Asst. Professor
11	Dr. N.Venkata Ramana	Asst. Professor
12	Dr.Jeevan Jala	Asst. Professor
13	Mrs. T.Ratna Kumari	Asst. Professor
14	Dr. Ch.Madhava Rao	Asst. Professor
15	Dr.M.Ragini	Asst. Professor
16	Mrs.V.Swathi	Asst. Professor
17	Mrs.T.Hema Latha	Asst. Professor
18	Dr.B. Krishna Prasad	Asst. Professor
19	Dr. G.Venkateswarlu	Asst. Professor
20	P.VASU DEV(2400520064)	2nd Year-Student Representative -BCA
21	M.SIVANAGA GANESH(2500520024)	3rd Year-Student Representative -BCA
22	BANTROTH SUHASINI (2501600007)	2nd Year-Student Representative-MCA
23	UJJWAL RAJ(2501600045)	2nd Year-Student Representative-MCA

31/7/26

HoD-CSA
Dr. K BHAGAVAN
HOD-Computer Science and Applications
Koneru Lakshmaiah Education Foundation
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-522502.

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Approved by AICTE ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002, Ph: +91 - 866 - 3500122, 2576129

Minutes of Meeting:

1. DAC recommended to refine the structures of BCA and MCA.
2. The Department Academic Committee's suggestions are discussed, and key proposals such as integrating research-oriented learning are considered.
3. Faculties Smt T.Hemalatha & Dr.M.N.V.Kiran Babu suggested to include few AI related courses in the curriculum of BCA & MCA from Y26 regulation.
4. Faculty Dr.Valli Mayil & Dr. T.V.S.Narayana proposed to run MCA & BCA in Trimester Mode instead of Regular Mode.
5. Faculty Smt.Priya Ranjani suggested to include more hours for skilling to acquire the knowledge.
6. Dr. Ch. Madhava Rao and Dr. Jeevan Jala suggested to increase practical hours to get more hands-on experience.
7. Faculty Dr. Raginia & Dr.G.Venkateswarlu suggested to introduce Blockchain & Fintech specialization for the Y26 batch
8. Dr. Y.RamaKrishna & Dr.B.Krishnaprasad suggested to introduce Term Paper and Capstone Project from Y26 Regulation.

Dr.K.Bhagavan
BoS Chairperson

31/7/26

HoD-CSA
Dr. K. BHAGAVAN
HOD-Computer Science and Applicatio
Koneru Lakshmaiah Education Foundati
(K.L. Deemed to be University)
Green Fields, VADDESWARAM,
Guntur District, Andhra Pradesh-5225