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 Bachelor of Computer Applications

Academic Year :2026-2027

S.No	Course Code	Course Title	CO. No	Course Outcome
1	26CA1101E	JAVA PROGRAMMING	CO1	Apply object-oriented programming principles — encapsulation, inheritance, polymorphism, abstraction — to design Java classes and class hierarchies for application-domain entities.
			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.
			CO3	Analyze concurrent execution scenarios using threads, ExecutorService, and synchronization primitives to identify race conditions and design correct multi-threaded application components.
			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.
			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.
			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).
2	26CA1102E	MATHEMATICS FOR COMPUTER APPLICATIONS	CO1	Apply discrete mathematics — sets, logic, relations, functions — to model computational problems involving structured reasoning.
			CO2	Apply linear algebra — vectors, matrices, transformations, eigenvalues — using NumPy to solve systems and analyze data geometrically.
			CO3	Analyze derivative-based optimization (gradient descent) and integration to understand the math underlying machine learning training.
			CO4	Analyze probability distributions, expectation, variance, and Bayes' theorem to reason about uncertainty in data and predictions.
			CO5	Evaluate combinatorial counting and graph-theoretic

K. Kiran Kumar
Academic Professor – I/C

7/8/26
HOD-BCA

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				constructions to model dependencies, networks, and computational complexity.
			CO6	Evaluate numerical methods for solving equations, integration, and optimization to choose appropriate techniques for engineering problems.
3	26CA1103E	LINUX SYSTEMS PROGRAMMING	CO1	Apply Linux command-line tools — file manipulation, text processing, pipes, redirection — to perform common system administration and developer tasks fluently.
			CO2	Apply Bash scripting (variables, conditionals, loops, functions, parameter handling) to automate routine workflows and build reusable command-line tools.
			CO3	Analyze process management — fork, exec, signals, job control — to design programs that interact correctly with the operating system.
			CO4	Analyze file descriptors, system calls, and the C-level interface to OS services to write low-level system programs.
			CO5	Evaluate inter-process communication mechanisms (pipes, FIFOs, sockets) and choose appropriate IPC for specific application architectures.
			CO6	Evaluate Linux-based DevOps fundamentals — package management, services, networking, security — to provision and maintain a working Linux server.
4	26UC1108E	ENGLISH COMMUNICATION	CO1	Apply listening comprehension strategies to professional contexts including lectures, technical talks, podcasts, and meetings; produce structured notes and summaries.
			CO2	Apply spoken English fluency in professional and academic contexts including team discussions, brief presentations, and technical conversations using clear pronunciation and pacing.
			CO3	Apply reading strategies (skimming, scanning, intensive, critical) to technical articles, news pieces, research abstracts, and literary excerpts.
			CO4	Analyze grammar, syntax, and vocabulary errors in own and peers' writing; produce error-free professional prose using active voice and clear sentence structure.
			CO5	Apply academic and professional writing conventions to compose emails, reports, summaries, and short essays appropriate to the audience and purpose.
			CO6	Examine cross-cultural communication patterns and apply situational awareness to navigate workplace dialogues, interview interactions, and team feedback.
5	26CA1204E	DATA STRUCTURES(Java)	CO1	Apply linear data structures (arrays, linked lists) and asymptotic analysis (Big-O) to evaluate the cost of operations on application data.
			CO2	Apply abstract data types (stacks, queues, deques) and recursive algorithms to solve problems with natural LIFO/FIFO/recursive structures.
			CO3	Analyze tree-based data structures (BST, AVL, heap) to choose appropriate structures for ordered or priority-based access patterns.

Academic Professor – I/C

HOD-BCA 7/01/26

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
			CO4	Analyze hash-based data structures and collision handling to design efficient lookup-heavy data layers.
			CO5	Evaluate graph data structures and traversal algorithms (BFS, DFS, Dijkstra) to solve real-world routing and connectivity problems.
			CO6	Evaluate the performance characteristics of competing data-structure choices and refactor application code to use the optimal structure for each access pattern.
6	26CA1205E	PYTHON PROGRAMMING	CO1	Apply Python's idiomatic features — comprehensions, generators, decorators, dynamic typing — to write clean, expressive Python programs that contrast with Java's static style.
			CO2	Apply Python data structures (lists, dicts, sets, tuples) and File/JSON/CSV I/O to ingest, transform, and persist real-world structured data.
			CO3	Analyze data with NumPy and pandas — vectorized operations, DataFrames, groupby, joins — to extract insights from realistic datasets.
			CO4	Analyze visualization choices using matplotlib, seaborn, and Plotly to design effective charts that communicate data narratives.
			CO5	Evaluate strategies for consuming external data via HTTP APIs, web scraping, and authentication patterns to build data-integrated Python applications.
			CO6	Evaluate the deployment readiness of a Python data application — testing with pytest, packaging, virtual environments, and Streamlit for shareable dashboards.
7	26CA1206E	FRONTEND DEVELOPMENT	CO1	Apply semantic HTML5, modern CSS (Flexbox, Grid, custom properties), and responsive design (mobile-first, media queries) to build accessible, well-structured web pages.
			CO2	Apply modern JavaScript (ES2020+: let/const, arrow functions, modules, async/await, destructuring, spread/rest) and the DOM API to build interactive web pages.
			CO3	Analyze React fundamentals (components, JSX, props, state, hooks — useState, useEffect, useContext) to architect single-page applications.
			CO4	Analyze React Router for client-side routing, plus state management patterns (lifting state, Context API, intro to Zustand/Redux) for non-trivial apps.
			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.
			CO6	Evaluate frontend tooling (Vite, ESLint, Prettier), testing (Jest + React Testing Library), accessibility (WCAG basics), and deployment (Vercel / Netlify / Cloudflare Pages).
8	26UC1209E	TECHNICAL & PROFESSIONAL COMMUNICATION	CO1	Apply technical writing conventions to produce documentation including README files, API docs, runbooks, and design documents using clear, concise prose.

Academic Professor – I/C

HOD-BCA 7/8/26

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
			CO2	Apply scientific and technical presentation norms to deliver short technical talks (Lightning, conference style, demo) with structured slides and confident Q&A handling.
			CO3	Analyze technical documentation patterns (Diátaxis framework: tutorial, how-to, reference, explanation) and select appropriate structure for given communication tasks.
			CO4	Examine cross-functional communication scenarios (developer-to-PM, developer-to-designer, developer-to-business stakeholder) and adapt language to audience expertise level.
			CO5	Apply business communication conventions including meeting facilitation, status reporting, escalation, and cross-team negotiation in professional simulation contexts.
			CO6	Examine ethical communication including disclosure, attribution, dissent, whistleblowing protocols, and Indian workplace communication norms (hierarchy, indirect speech, face-saving).
9	26CA1307E	APPLIED AI & MACHINE LEARNING	CO1	Apply the standard ML workflow — problem framing, data acquisition, EDA, feature engineering, model training, evaluation — to a real-world structured dataset.
			CO2	Apply supervised learning algorithms — linear/logistic regression, k-NN, SVM, naive Bayes — using scikit-learn to solve regression and classification problems.
			CO3	Analyze model performance using cross-validation, confusion matrices, ROC curves, precision/recall trade-offs to diagnose underfitting, overfitting, and bias-variance issues.
			CO4	Analyze unsupervised learning techniques — k-means, hierarchical clustering, PCA, t-SNE — to discover structure in unlabeled data.
			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.
			CO6	Evaluate ML production concerns — model serialization, REST API serving, monitoring, drift, ethical considerations — to deploy a trained model as a usable service.
10	26CA1308E	DBMS & BACKEND DEVELOPMENT	CO1	Apply relational database design principles (normalization, keys, constraints) and SQL (DDL, DML, joins, subqueries, CTEs) to design and query non-trivial schemas.
			CO2	Apply ORM patterns (JPA / Hibernate) and migration tools (Flyway / Liquibase) to manage schema evolution and persist Java objects.
			CO3	Analyze RESTful API design — resource modeling, HTTP semantics, status codes, versioning, OpenAPI documentation — to build well-structured Spring Boot services.
			CO4	Analyze authentication, authorization, input validation, and error-handling patterns in backend systems to harden services against common vulnerabilities.
			CO5	Evaluate trade-offs between SQL and NoSQL stores

Academic Professor – I/C

HOD-BCA 7/8/26

Dr. K.BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				(PostgreSQL vs MongoDB vs Redis) and apply caching strategies to optimize backend latency and throughput.
			CO6	Evaluate the production readiness of a backend — testing strategy, observability (logging, metrics), graceful shutdown, deployment via Docker — to ship a public REST service.
11	26CA1309E	OPERATING SYSTEMS & CONCURRENT PROGRAMMING	CO1	Apply process and thread concepts (PCB, TCB, fork/exec, pthreads, Java threads) to write correct multi-process and multi-threaded programs.
			CO2	Apply synchronization primitives (mutexes, semaphores, condition variables, monitors) and detect race conditions in concurrent code.
			CO3	Analyze CPU scheduling algorithms (FCFS, SJF, priority, RR, MLFQ, CFS) by simulating their behavior on workloads and computing turnaround / waiting / response metrics.
			CO4	Analyze memory management techniques (paging, segmentation, virtual memory, page replacement: FIFO, LRU, Optimal) and reason about TLBs, page faults, working sets.
			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.
			CO6	Evaluate concurrent programming patterns (thread pools, producer-consumer, reader-writer, futures/promises, async/await) for correctness, throughput, and scalability.
12	26UC0039E	INNOVATION & ENTREPRENEURSHIP	CO1	Apply design-thinking and customer-discovery techniques (problem interviews, empathy mapping, JTBD framework) to identify validated user problems worth solving.
			CO2	Analyzestartup business models using contemporary frameworks (Lean Canvas, Business Model Canvas, value proposition canvas) and assess product-market fit signals.
			CO3	Apply lean startup methodology (build-measure-learn) to validate hypotheses through MVPs, A/B tests, and iterative customer development.
			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.
			CO5	Analyze financial fundamentals for technology ventures: unit economics (CAC, LTV, payback), runway, burn rate, capital efficiency, valuation methods (DCF, comps, VC method).
			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.
13	26IE2040E	Social Internship — Community Engagement	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.
			CO2	Apply needs-assessment frameworks (asset mapping, root-

Academic Professor – I/C

HOD-BCA

Dr. K.BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				cause analysis, stakeholder mapping) to identify community priorities and existing resources or gaps.
			CO3	Examine the structural drivers of community challenges including socioeconomic, policy, infrastructure, and cultural dimensions specific to the host context.
			CO4	Apply participatory engagement methods (community meetings, focus groups, surveys, intergenerational dialogue) to co-design intervention concepts with community members..
			CO5	Examine the role of NGOs, government schemes (PMGKAY, MGNREGA, NRLM, ASHA, anganwadi), Panchayati Raj institutions, and CSR programs operating in the host community.
			CO6	Architect a sustainability-aware community impact report with verifiable observations, evidence-based recommendations, and reflective self-assessment.
14	26UC0009E	Ecology& Environment	CO1	Apply ecological concepts (biotic-abiotic interactions, energy flow, biogeochemical cycles, food webs, succession) to analyze local and global ecosystems.
			CO2	Analyze biodiversity at gene, species, and ecosystem levels and examine causes of biodiversity loss including habitat fragmentation, invasive species, climate change, and pollution.
			CO3	Examine major environmental challenges (climate change, air and water pollution, soil degradation, plastic waste, e-waste) and the policy/technology responses to each.
			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.
			CO5	Apply sustainability frameworks (UN SDG-13, ESG, circular economy, carbon footprint) to evaluate and reduce environmental impact of computing infrastructure and consumer choices.
			CO6	Build a green audit / sustainability report for the campus or a chosen organization including baseline measurement, impact analysis, and intervention recommendations.
15	26CA2110E	ALGORITHM DESIGN AND ANALYSIS	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
			CO2	Apply divide-and-conquer (merge sort, quicksort, binary search variants, Strassen) to design recursive algorithms and analyze them via recurrences.
			CO3	Analyze greedy algorithms (activity selection, fractional knapsack, Huffman, MST) to determine when local optima yield global optima — and when they don't.
			CO4	Analyze dynamic programming patterns (memoization, tabulation, optimal substructure, overlapping subproblems) on classic problems (LCS, edit distance, 0/1 knapsack, matrix chain).

Academic Professor – I/C

HOD-BCA

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
			CO5	Evaluate advanced graph algorithms (Bellman-Ford, Floyd-Warshall, max flow, strongly connected components) to choose appropriate techniques for path, flow, and connectivity problems.
			CO6	Evaluate computational tractability via P, NP, NP-completeness, and reductions; apply approximation, heuristics, and randomization where exact solutions are infeasible.
16	26CA2111E	COMPUTER NETWORKS & APPLICATIONS	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.
			CO2	Apply transport-layer protocols (TCP, UDP, QUIC) — connection setup, flow control, congestion control, reliability — to choose appropriate transports for application requirements.
			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.
			CO4	Analyze socket programming patterns — blocking, non-blocking, event loops, select/poll/epoll — to implement client/server applications in Python and Java.
			CO5	Evaluate distributed system fundamentals — RPC, message queues, service discovery, load balancing, caching, CDNs — to design scalable network services.
			CO6	Evaluate network security fundamentals — TLS handshake, certificates, OAuth 2.0, OWASP API Security Top 10 — to harden network applications.
17	26CA2112E	APPLIED NETWORKS & VIRTUALIZATION	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.
			CO2	Apply DNS, DHCP, HTTP/HTTPS, and TLS protocols to set up working network services and capture/decode their wire traffic.
			CO3	Analyze socket programming (TCP and UDP) in Python or Java to build client-server applications and reason about reliability, ordering, and concurrency.
			CO4	Analyze virtualization fundamentals (Type-1 vs Type-2 hypervisors, KVM/QEMU, VirtualBox), and Linux network namespaces to understand container networking foundations.
			CO5	Evaluate network design choices (subnetting, NAT, routing, VLANs, basic SDN concepts) on a Mininet-emulated topology.
			CO6	Evaluate network performance (latency, throughput, packet loss) and basic network security (firewalls, port scanning awareness, encrypted traffic) on a working topology.
18	FLP	FOREIGN LANGUAGE PROFICIENCY	CO1	Apply A1 (CEFR) listening competence in the chosen language to comprehend short conversations, basic instructions, and slow speech on familiar topics.
			CO2	Apply A1 speaking competence to introduce oneself, ask

K. Kiran Kumar

Academic Professor – I/C

7/8/26

HOD-BCA

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				and answer simple questions, describe people/places/objects, and engage in basic transactional conversations.
			CO3	Apply A1 reading competence to comprehend short texts (signs, menus, simple emails, brief news headlines) using context and cognates.
			CO4	Apply A1 writing competence to compose short messages, fill forms, and write 50-80 word personal descriptions.
			CO5	Examine cultural conventions of the target language community including greetings, formality registers, body language, and key cultural references.
			CO6	Examine the global business and academic relevance of the chosen language including industries, multinational companies, and study/work pathways.
19	26CA2213E	ADVANCED DATA STRUCTUES & ALGORITHMS	CO1	Apply advanced data structures (segment trees, Fenwick / BIT, sparse tables, persistent DS, treaps, link-cut trees intro) to range query and update problems.
			CO2	Apply string algorithms (KMP, Z, Manacher, Aho-Corasick, suffix array, suffix automaton intro) to matching, palindrome, and substring problems.
			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.
			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).
			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.
			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.
20	26CA2214E	CLOUD COMPUTING & DEVOPS	CO1	Apply core cloud computing service models (IaaS, PaaS, SaaS) and AWS primitives (EC2, S3, IAM, VPC, RDS) to provision basic cloud infrastructure.
			CO2	Apply containerization with Docker — images, containers, multi-stage builds, networking, volumes — to package applications consistently across environments.
			CO3	Analyze Kubernetes architecture and primitives (pods, deployments, services, ingress, ConfigMap, secrets) to orchestrate containerized applications at scale.
			CO4	Analyze Infrastructure-as-Code patterns using Terraform to declaratively provision and manage cloud resources reproducibly.
			CO5	Evaluate CI/CD pipelines using GitHub Actions or Jenkins to automate the build, test, and deployment lifecycle of applications.
			CO6	Evaluate cloud observability and reliability practices —

Academic Professor – I/C

HOD-BCA 7/7/26

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				Prometheus metrics, Grafana dashboards, log aggregation, alerting, blue-green and canary deployments — for production-grade operations.
21	26CA2215E	DEEP LEARNING FOUNDATIONS	CO1	Apply foundational neural network concepts (perceptron, MLP, forward pass, backpropagation, gradient descent) by implementing them from scratch in NumPy and verifying against PyTorch.
			CO2	Apply PyTorch fundamentals (tensors, autograd, nn.Module, optimizers, DataLoader) to train a feedforward network on a real dataset.
			CO3	Analyze convolutional neural networks (convolution, pooling, feature maps, classic architectures: LeNet → AlexNet → ResNet) for image classification tasks.
			CO4	Analyze sequence models (RNN, LSTM, GRU) and their limitations to motivate the Transformer architecture.
			CO5	Evaluate the Transformer architecture (self-attention, multi-head attention, positional encoding) and apply transfer learning with HuggingFacepretrained models.
			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.
22	26CA2216E	CYBERSECURITY FOUNDATIONS	CO1	Apply the CIA triad, AAA model, and threat-modeling frameworks (STRIDE, attack trees) to identify security goals and risks for an application.
			CO2	Apply cryptographic primitives (symmetric: AES; asymmetric: RSA, ECC; hashes: SHA-2, SHA-3, bcrypt; HMAC; digital signatures) correctly using standard libraries.
			CO3	Analyze network security mechanisms (TLS handshake, firewalls, IDS/IPS basics, VPNs) and detect basic attacks via packet capture analysis.
			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.
			CO5	Evaluate authentication and identity management designs (password storage, MFA, OAuth 2.0, OIDC, session management) for production applications.
			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).
23	26CA2317E	GENERATIVE AI FOUNDATIONS	CO1	Apply foundational generative-AI concepts — language modeling, autoregressive generation, tokenization, embeddings — to use modern LLMs (OpenAI, Anthropic, open-source) via APIs and SDKs.
			CO2	Apply prompt engineering patterns (zero-shot, few-shot, chain-of-thought, ReAct, structured output) to elicit reliable, structured responses from LLMs.
			CO3	Analyze the Retrieval-Augmented Generation (RAG)

Academic Professor – I/C

HOD-BCA 7/4/26

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				architecture — chunking, embeddings, vector stores, retrieval, re-ranking, generation — to ground LLM responses in private documents.
			CO4	Analyze fine-tuning approaches (full fine-tuning, LoRA, QLoRA, instruction tuning) to specialize a base LLM for a domain task with limited compute.
			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.
			CO6	Evaluate generative AI systems on cost, latency, faithfulness, hallucination rates, safety, and bias to make informed deployment decisions.
24	26CA2318E	MACHINE LEARNING OPERATIONS (MLOps)	CO1	Apply ML lifecycle stages (data collection, training, evaluation, deployment, monitoring) and reproducibility tools (MLflow, DVC) to manage machine learning experiments and artifacts.
			CO2	Apply data versioning, feature stores, and data validation frameworks (Great Expectations / Pandera) to build trustworthy machine learning data pipelines.
			CO3	Analyze machine learning deployment patterns (REST API, batch, streaming, edge) and model serving frameworks (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.
			CO5	Evaluate machine learning monitoring practices—data drift, concept drift, prediction drift, and model staleness—and design alerting and automated retraining strategies.
			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.
25	26CA2319E	AI PLATFORM ENGINEERING	CO1	Apply LLM serving patterns — provider APIs, vLLM, Ollama, TGI, Triton — to deploy LLMs and embedding models for production inference.
			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.
			CO3	Analyze vector database operations — collection design, sharding, replication, hybrid search, indexing trade-offs (HNSW, IVF) — for production-scale RAG.
			CO4	Analyze AI observability requirements — token usage, latency, cost, quality drift, prompt/response logging — and design observability for LLM applications.
			CO5	Evaluate AI cost engineering practices — model cascading, caching, batching, prompt compression, fine-tuning vs prompting trade-offs — to manage operational costs.

Academic Professor – I/C

HOD-BCA 7/8/26

Dr. K.BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
			CO6	Evaluate AI platform security and governance — prompt injection defenses, PII redaction, content moderation, audit trails, data residency — for enterprise deployment.
26	26CA2320E	SOFTWARE ENGINEERING & AGILE	CO1	Apply Git collaboration workflows (feature branches, pull requests, rebasing, conflict resolution, tagged releases) and conduct constructive code reviews.
			CO2	Apply Scrum and Kanban practices — backlog grooming, sprint planning, daily stand-ups, retrospectives, story pointing — to deliver software in short, predictable cycles.
			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.
			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.
			CO5	Evaluate software quality through static analysis, code metrics (cyclomatic complexity, coverage, maintainability index), technical debt accounting, and architecture decision records (ADRs).
			CO6	Evaluate software project management practices — estimation, dependencies, blockers, on-call rotations, postmortems — and contribute to a healthy team engineering culture.
27	26IE3041E	TECHNICAL INTERNSHIP	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.
			CO2	Apply technical contribution discipline (writing PRs, addressing review feedback, attending sprint ceremonies, communicating blockers) to ship measurable work product within the internship period.
			CO3	Examine the host organization's product, business model, technical architecture, customer base, and team structure to contextualize own contribution within the broader system.
			CO4	Examine cross-functional collaboration patterns (developer-PM-design-QA-DevOps) and the cadence of professional engineering work (planning, daily standups, retrospectives, on-call).
			CO5	Apply professional communication and documentation standards in industry-grade artifacts including design docs, runbooks, README files, postmortems, and status updates.
			CO6	Architect a structured internship reflection that maps observations to learning outcomes and articulates intentional career direction informed by the experience.
28	26UC0032E	INSIGHTS TO SUSTAINABLE DEVELOPMENT GOALS	CO1	Examine the 17 UN Sustainable Development Goals, their 169 targets, and the 248 indicators framework adopted in 2015 with target year 2030.
			CO2	Analyze India's progress on each SDG using NITI Aayog's SDG India Index, the State and District Indices, and identify regional disparities.

Academic Professor – I/C

HOD-BCA 7/8/26

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
			CO3	Examine the role of technology (AI, IoT, blockchain, satellite imagery, mobile platforms) in accelerating progress toward specific SDGs.
			CO4	Examine corporate ESG reporting, BRSR (Business Responsibility & Sustainability Report) framework mandated for top 1000 listed Indian companies, and SDG-aligned investment.
			CO5	Apply SDG-impact assessment to evaluate the contribution of a project, product, or organization to specific goals using verifiable indicators.
			CO6	Architect an SDG-aligned intervention or technology proposal for a real Indian challenge with measurable impact pathway and stakeholder analysis.
29	26CA3121E	HIGH PERFORMANCE COMPUTING	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.
			CO2	Apply shared-memory parallelism in Python — multiprocessing, concurrent.futures, threading limitations (GIL) — and vectorization with NumPy&Numba to accelerate CPU-bound data tasks.
			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.
			CO4	Analyze distributed-computing patterns — MapReduce, dataflow, partitioning, shuffle costs — using Apache Spark / PySpark for data-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.
			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.
30	26CA3122E	OFFENSICE SECURITY	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.
			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.
			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).

Academic Professor – I/C

HOD-BCA

Dr. R. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
			CO4	Analyze network and Active Directory attack surfaces — SMB, Kerberos, NTLM, common misconfigurations — on authorized lab targets (Hack The Box / TryHackMe AD-themed boxes).
			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).
			CO6	Evaluate findings, write professional penetration test reports (executive summary, technical detail, CVSS scoring, remediation guidance, retest plan), and propose defensive controls.
31	26CA3123E	AGENTIC AI &APPLICATIONS	CO1	Apply foundational agent patterns — ReAct, function calling, planning, reflection — to build single-agent systems that complete multi-step tasks.
			CO2	Apply shared-memory parallelism in Python — multiprocessing, concurrent.futures, threading limitations (GIL) — and vectorization with NumPy&Numba to accelerate CPU-bound data tasks.
			CO3	Analyze agent memory architectures (short-term context, episodic, semantic, procedural) and orchestration frameworks (LangGraph, CrewAI, AutoGen, OpenAI Swarm, Anthropic SDK).
			CO4	Analyze multi-agent collaboration patterns — supervisor-worker, debate, hierarchical teams, role specialization, handoff protocols — to compose teams of agents.
			CO5	Evaluate agent behavior using trajectory evals, rubric-based scoring, automated red teaming, and benchmark suites (AgentBench, SWE-bench, GAIA, τ-bench).
			CO6	Evaluate agent safety controls — sandboxing, human-in-the-loop, action confirmation, prompt injection defenses, audit trails — for production agent deployment.
32	26CA31A1E	CONVERSATIONAL AI & SPEECH SYSTEMS	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.
			CO2	Analyse intent classification, entity extraction, and dialogue state tracking architectures to design conversational systems that handle multi-turn interactions and context..
			CO3	Examine voice UI patterns, barge-in handling, latency budgets, and conversational repair strategies to evaluate user experience quality of voice-first applications.
			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..
			CO5	Evaluate speech model performance using WER, MOS, intent F1, task success rate, and conversation analytics to identify failure modes and optimisation opportunities.
			CO6	Architect a multilingual, telephony-integrated voice assistant deployed on cloud infrastructure with monitoring, fallback policies, and PII redaction for regulated production

Academic Professor – I/C

HOD-BCA

Dr. K.BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				use..
33	26CA31B1E	CLOOUD -NATIVE INFRASTRUCTURE & GITOPS	CO1	Analyse advanced Kubernetes constructs (operators, CRDs, admission controllers, RBAC) to design extensions that automate operational concerns beyond stock Deployment/Service primitives..
			CO2	Design declarative infrastructure using Terraform and Pulumi modules, workspaces, and state management strategies suitable for multi-environment, multi-team cloud platforms...
			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.
			CO4	Examine service mesh architectures (Istio, Linkerd, Cilium) to evaluate trade-offs in observability, security, and traffic management for microservice deployments.
			CO5	Architect multi-cluster Kubernetes topologies including federation, fleet management, and cross-cluster networking to support production-grade availability and disaster recovery requirements.
			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.
34	26CA31C1E	WEB & API SECURITY	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.
			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.
			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.
			CO4	Implement automated security testing pipelines combining SAST, DAST, IAST, SCA, and secrets scanning integrated with CI/CD for continuous web/API security verification.
			CO5	Evaluate API-specific protections including rate limiting, schema validation, mTLS, GraphQL query cost analysis, and Web Application and API Protection (WAAP) deployment patterns.
			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.
35	26CA31D1E	SMART CONTRACT DEVELOPMENT & SECURITY	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.
			CO2	Design Solidity contracts implementing major token

Academic Professor – I/C

HOD-BCA

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				standards (ERC-20, ERC-721, ERC-1155, ERC-4626) using OpenZeppelin libraries with appropriate access control and event emission patterns..
			CO3	Implement comprehensive smart contract test suites combining unit, fuzz, invariant, and fork tests using Foundry to achieve high confidence before deployment.
			CO4	Architect upgradeable contract systems using transparent proxy, UUPS, and Diamond patterns with explicit migration strategies and access controls for upgrade authority.
			CO5	Examine smart contract security vulnerability classes (reentrancy, access control, integer issues, oracle manipulation, MEV, signature replay) and apply defensive coding patterns.
			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.
36	26CA31E1E	DATA ENGINEERING AT SCALE	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.
			CO2	Design batch data pipelines using orchestrators (Airflow, Prefect, Dagster) including DAG decomposition, dependency management, retry semantics, backfill, and lineage tracking.
			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..
			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.
			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.
			CO6	Build a production-grade data platform integrating ingestion, transformation, modelling, quality checks, orchestration, and BI consumption with documented cost and SLA targets.
37	26CA3224E	DEFENSIVE SECURITY(SOC, SIEM, DFIR)	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.
			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.
			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.
			CO4	Analyze digital forensics & incident response (DFIR)

Academic Professor – I/C

HOD-BCA

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				workflows — evidence acquisition, chain of custody, memory analysis (Volatility), disk forensics (Autopsy), timeline analysis — for Linux and Windows systems.
			CO5	Evaluate threat intelligence integration — MITRE ATT&CK navigator, threat feeds (MISP), IoC management, threat hunting hypotheses — and propose threat-informed defense improvements.
			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.
38	26CA32A2E	COMPUTER VISION & VISUAL INTELLIGENCE	CO1	Apply pretrained object detection, segmentation, and OCR models to extract structured information from images and video streams using contemporary frameworks.
			CO2	Analyse data labeling strategies, annotation quality, and dataset bias in computer vision pipelines to identify failure modes before model deployment.
			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.
			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.
			CO5	Evaluate CV model performance using mAP, IoU, FPS, latency p95, and domain-specific business metrics to make deployment go/no-go decisions.
			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.
39	26CA33B4E	EVENT-DRIVEN & SERVERLESS PATTERNS	CO1	Analyseserverless 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.
			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.
			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.
			CO4	Examine serverless data layers (DynamoDB, D1/Turso/Neon, Cloudflare KV/R2, UpstashRedis) to evaluate fit-for-purpose across access patterns, consistency requirements, and cost models.
			CO5	Evaluate serverless and edge applications using observability tooling (CloudWatch, Cloudflare Analytics, DatadogServerless) on cold start latency, p95/p99, error rate, and cost per invocation.
			CO6	Architect a globally-distributed application combining edge

Academic Professor – I/C

HOD-BCA

Dr. K.BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				functions, durable storage, queues, and traditional backends with explicit failure-mode handling and cost projections at scale.
40	26CA32C2E	PRIVACY ENGINEERING & DATA PROTECTION	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.
			CO2	Design privacy-by-design and privacy-by-default architectures applying data minimisation, purpose limitation, and storage limitation principles to real product specifications.
			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.
			CO4	Implement privacy-enhancing technologies (PETs) including secure aggregation, federated learning at small scale, homomorphic encryption primitives, and consent management platforms.
			CO5	Architect data subject rights (DSR) workflows including access, correction, erasure, portability, and consent withdrawal across distributed systems with proof of completion.
			CO6	Build a Data Protection Impact Assessment (DPIA) process and a breach notification runbook compliant with DPDP Act timelines and content requirements.
41	26CA32D2E	DECENTRALIZED FINANCE & TOKENOMICS	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.
			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.
			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.
			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.
			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.
			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.

Academic Professor – I/C

HOD-BCA 7/8/24

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
42	26CA32E2E	CUSTOMER & GROWTH ANALYTICS	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.
			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.
			CO3	Implement propensity score methods (matching, weighting IPTW, stratification) and double machine learning to estimate average treatment effects from observational data.
			CO4	Examine generative model families (GANs, VAEs, diffusion models, normalising flows, autoregressive models) for their suitability to tabular synthesis, image augmentation, and counterfactual generation.
			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.
			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.
43	26CA32A3E	DEEP LEARNING ARCHITECTURES	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.
			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.
			CO3	Analyse generative architectures (VAEs, GANs, diffusion models, normalising flows) to evaluate trade-offs in sample quality, training stability, and inference cost.
			CO4	Design self-supervised and contrastive learning pipelines (SimCLR, MoCo, MAE, DINO) that pretrain representations on unlabelled data for downstream tasks with limited labels.
			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.
			CO6	Critique recent deep learning research papers using a structured methodology and synthesise findings into architectural design decisions for a novel application.
44	26CA32B3E	SITE RELIABILITY ENGINEERING & OBSERVABILITY	CO1	Analyse Site Reliability Engineering principles (SLI, SLO, error budgets, toil, eliminating heroics) to translate vague reliability expectations into measurable engineering contracts.
			CO2	Design observability strategies combining metrics, structured logs, and distributed traces using OpenTelemetry,

Academic Professor – I/C

HOD-BCA

7/8/26

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				Prometheus, Loki, and Tempo to make production systems debuggable without re-deployment.
			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.
			CO4	Implement chaos engineering experiments using Litmus or Chaos Mesh to validate hypotheses about system resilience before incidents occur in production.
			CO5	Architect incident response processes including on-call rotations, runbooks, war-room facilitation, and blameless postmortems to learn from failures and improve systems.
			CO6	Build capacity planning and performance engineering practices including load testing, resource right-sizing, and predictive scaling to anticipate failure modes before they arrive.
45	26CA32C3E	CLOUD SECURITY & ZERO TRUST	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.
			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.
			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.
			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.
			CO5	Examine multi-account / multi-subscription landing zone architectures (AWS Organisations + SCPs, Azure Management Groups, GCP Organisation Policy) to enforce guardrails at scale.
			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.
46	26CA32D3E	FINANCIAL MARKETS & QUANTITATIVE METHODS	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.
			CO2	Design double-entry ledger systems in software with immutability, idempotency, and consistency guarantees suitable for production money movement.
			CO3	Implement payment gateway integrations (Razorpay, Stripe, PayU) including signed webhooks, idempotent operations, retries, refunds, and partial captures with proper failure handling.

Ic. Academic Professor – I/C

HOD-BCA 7/8/26

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
			CO4	Examine reconciliation systems comparing internal ledger to external statements (acquirer files, bank statements) including break investigation workflows and exception handling.
			CO5	Architect KYC, fraud-detection, and risk-scoring pipelines that meet regulatory requirements while maintaining acceptable customer onboarding and approval rates.
			CO6	Build a working payments slice including merchant onboarding, payment acceptance, ledger booking, settlement to merchant bank, reconciliation, and merchant-facing dashboards.
47	26CA32E3E	BIG DATA & DISTRIBUTED ANALYTICS	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.
			CO2	Design Apache Spark applications using DataFrame and Structured Streaming APIs with explicit awareness of partitioning, broadcast joins, skew, and Adaptive Query Execution (AQE).
			CO3	Examine columnar storage formats (Parquet, ORC, Arrow), file layout strategies, partitioning, bucketing, and Z-ordering to optimise query patterns on lakehouse storage.
			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.
			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.
			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.
48		FINANCIAL ACCOUNTING FOR TECH FOUNDERS	CO1	Apply double-entry bookkeeping principles to record day-to-day business transactions in journals, ledgers, and trial balances.
			CO2	Apply Indian Accounting Standards (Ind AS) and GST rules to prepare basic financial statements: P&L, balance sheet, and cash flow statement.
			CO3	Analyze financial statements using ratio analysis (liquidity, profitability, solvency, efficiency) to assess company health and performance.
			CO4	Examine startup-specific financial concepts: burn rate, runway, gross margin, unit economics, ARR/MRR, and how investors read these from financials.
			CO5	Apply tools like Tally Prime / Zoho Books / QuickBooks (or Excel) to maintain books for a small entity through one quarter of mock transactions.
			CO6	Examine compliance basics: GST returns (GSTR-1, GSTR-3B), TDS, advance tax, ROC filings, and penalties for non-

Academic Professor – I/C

HOD-BCA

Dr. K.BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				compliance — relevant for startup founders.
49		MARKETING &BRAND BUILDING	CO1	Apply marketing fundamentals — segmentation, targeting, positioning (STP) — to define a clear go-to-market strategy for a product or service.
			CO2	Examine the 4Ps (Product, Price, Place, Promotion) and 7Ps (adding People, Process, Physical Evidence) frameworks and apply them to product launch decisions.
			CO3	Analyze digital marketing channels (SEO, SEM, social media, content marketing, email, performance marketing) and select appropriate channel mix for given audiences and budgets.
			CO4	Apply brand-building frameworks (Aaker's Brand Identity, Kapferer's Prism, Jobs-to-be-Done) to develop coherent brand positioning and messaging.
			CO5	Examine consumer behavior fundamentals — the buying funnel (AIDA, AARRR), customer journey mapping, behavioral economics in marketing, B2B vs B2C dynamics.
			CO6	Architect a complete marketing plan for a real or simulated product, including STP, brand identity, content strategy, channel mix, and measurement plan.
50		BEHAVIORAL ECONOMICS & DECISION SCIENCES	CO1	Apply foundational behavioral economics concepts (heuristics, biases, bounded rationality, prospect theory) to analyze real-world decision-making patterns.
			CO2	Examine cognitive biases (confirmation, availability, anchoring, framing, sunk cost, hindsight, status quo) and identify their effects in product design, policy, and personal decisions.
			CO3	Apply nudge theory (Thaler&Sunstein) to design choice architecture that improves outcomes without restricting freedom of choice.
			CO4	Analyze applications of behavioral economics in product design (defaults, friction, loss aversion), public policy (savings programs, organ donation), and behavior change (health, sustainability).
			CO5	Examine ethical considerations in applying behavioral insights — the line between nudging and dark-pattern manipulation, transparency, autonomy, and informed consent.
			CO6	Architect a behavioral intervention proposal for a chosen problem (product, policy, or personal) with measurable hypotheses and an experimental design framework.
51		ORGANIZATIONAL BEHAVIOR & PEOPLE MANAGEMENT	CO1	Apply foundational concepts of organizational behavior — individual differences (personality, perception, motivation), group dynamics, and organizational culture — to analyze workplace situations.
			CO2	Examine major motivation theories (Maslow, Herzberg, McClelland, Self-Determination Theory) and apply them to design work environments that engage technology professionals.
			CO3	Analyze team dynamics through models like Tuckman's stages (forming-storming-norming-performing), Lencioni's

Academic Professor – I/C

HOD-BCA

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				Five Dysfunctions, and Google's Project Aristotle findings on psychological safety.
			CO4	Apply leadership frameworks — Situational Leadership (Hersey-Blanchard), Transformational vs Transactional leadership, Servant Leadership — to varied workplace contexts.
			CO5	Examine conflict management styles (Thomas-Kilmann), negotiation principles (Fisher &Ury), and difficult conversations frameworks for workplace effectiveness.
			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.
52		FILMMAKING & VISUAL STORYTELLING	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.
			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
			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.
			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.
			CO5	Examine sound design fundamentals — diegetic vs non-diegetic sound, music selection, dialogue clarity, ADR, foley basics — and apply them to enhance emotional impact.
			CO6	Architect a complete short film (3-5 minutes) from concept to delivery, demonstrating coherent integration of script, cinematography, editing, and sound design.
53		PHOTOGRAPHY FOR DIGITAL MEDIA	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.
			CO2	Apply composition principles (rule of thirds, leading lines, golden ratio, framing, negative space, symmetry, depth) to create visually compelling images.
			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.
			CO4	Apply post-processing fundamentals using Adobe Lightroom (or Darktable / Snapseed for free alternatives) — exposure correction, color grading, dodging/burning,

Academic Professor – I/C

HOD-BCA

Dr. K. BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				masking, batch processing.
			CO5	Examine photography genres and their conventions — portrait, street, documentary, landscape, food, product, event — and apply genre-appropriate techniques to specific assignments.
			CO6	Architect a coherent photo essay (10-15 images) on a chosen theme demonstrating technical proficiency, compositional rigor, and storytelling depth.
54		CREATIVE WRITING & NARRATIVE CRAFT	CO1	Apply fundamental craft elements of fiction — character, setting, plot, conflict, point-of-view, voice, dialogue, scene vs summary — to write coherent short narratives.
			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.
			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.
			CO4	Apply revision techniques — global revision (structure, voice), line editing (sentence-level), proofreading — across multiple drafts to substantially improve prose.
			CO5	Examine major Indian and global literary voices — across fiction, non-fiction, and essays — and identify craft moves that can be studied and applied.
			CO6	Architect a substantial original creative work (short story, personal essay collection, or hybrid piece) of 3000-5000 words demonstrating range, voice, and craft.
55	26IE3201E	TERM PAPER	CO1	Apply systematic literature search techniques (Google Scholar, ACM Digital Library, IEEE Xplore, arXiv, Semantic Scholar) and evaluate the quality of academic sources.
			CO2	Apply citation management (Zotero, Mendeley) and academic writing tools (LaTeX with Overleaf, BibTeX) to produce well-cited, properly 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.
			CO4	Analyze methodology choices for the contribution type — survey/review (PRISMA), empirical study (experimental design), comparative analysis (benchmark methodology), engineering case study.
			CO5	Evaluate research ethics — plagiarism prevention (Turnitin / iThenticate), authorship credit, AI-assistance disclosure, data privacy, IRB requirements (where applicable).
			CO6	Evaluate own and peer work through structured peer review, presentation, and revision cycles to produce a publication-standard paper.
56	26CA3325E	DEVSECOPS	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.
			CO2	Apply SAST (Semgrep, CodeQL, SonarQube), DAST

Academic Professor – I/C

HOD-BCA

Dr. K. BHAGAVAN
HOD - BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				(OWASP ZAP), and IaC scanning (Checkov, tfsec) within CI/CD pipelines.
			CO3	Analyze software supply chain risks — SBOM generation (CycloneDX, SPDX), artifact signing (Sigstore, cosign), SLSA framework levels — and implement supply-chain controls.
			CO4	Analyze policy-as-code patterns with Open Policy Agent (OPA) and Conftest to enforce security guardrails on Kubernetes manifests, Terraform, and CI workflows
			CO5	Evaluate threat modeling methodologies (STRIDE, attack trees) and integrate threat models into design reviews and PR templates.
			CO6	valuate runtime security controls — Kubernetes admission webhooks, Falco, distroless / minimal images, RBAC hardening — and propose a balanced controls portfolio.
57	26CA33A4E	NATURAL LANGUAGE PROCESSING & LLM FOUNDATIONS	CO1	Analyse classical NLP foundations (tokenisation, embeddings, parsing, NER) to choose appropriate techniques for tasks where modern LLMs are overkill or unsuitable.
			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.
			CO3	Design pretraining and fine-tuning pipelines including masked language modelling, causal language modelling, and instruction tuning for a small language model.
			CO4	Implement alignment techniques (supervised fine-tuning, RLHF, DPO, constitutional AI) at small scale to shape model behaviour and reduce harmful outputs.
			CO5	Evaluate language models using perplexity, MMLU, HELM, BIG-bench, and task-specific benchmarks to make informed model selection and reporting decisions
			CO6	Architect a small-scale language model end-to-end including tokeniser training, pretraining on a domain corpus, instruction tuning, alignment, evaluation, and serving.
58	26CA33B4E	SERVERLESS & EDGE COMPUTING	CO1	Analyse service mesh data plane and control plane architectures (Istio, Linkerd, Cilium) to evaluate latency, resource overhead, and operational complexity trade-offs.
			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.
			CO3	Examine event-driven architectural patterns (event notification, event-carried state transfer, event sourcing, CQRS) to choose the right pattern for each business domain.
			CO4	Implement event streaming systems using Apache Kafka including partitioning, consumer groups, exactly-once semantics, schema evolution, and stream processing topologies
			CO5	Compare event-driven communication backbones (Kafka,

Academic Professor – I/C

HOD-BCA 7/4/26

Dr. K.BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				NATS, RabbitMQ, Redpanda, AWS EventBridge) to make justified choices based on throughput, ordering, durability, and operational requirements.
			CO6	Architect a microservices system combining service mesh for synchronous communication and event streaming for asynchronous workflows with explicit consistency boundaries and failure handling.
59	26CA33C4E	SECURITY GOVERNANCE, RISK &COMPLIANCE	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.
			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.
			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.
			CO4	Implement security policies, standards, and procedures (the policy hierarchy) that balance regulatory completeness with practical operability for a small-to-mid-size organisation.
			CO5	Architect compliance automation pipelines using continuous-control-monitoring platforms (Drata, Vanta, Sprinto, Tugboat Logic) to replace manual evidence collection with engineering-driven attestation
			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
60	26CA33D4E	REGTECH & COMPLIANCE AUTOMATION	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.
			CO2	Design risk-based KYC and Customer Due Diligence (CDD) workflows including simplified, standard, and enhanced due diligence tiers based on customer risk classification.
			CO3	Implement transaction monitoring systems combining rule-based scenarios (velocity, structuring, threshold) with ML-based anomaly detection, generating alerts with low false-positive rates.
			CO4	Examine sanctions screening, PEP screening, and adverse-media screening pipelines including list management, fuzzy matching, and false-positive workflows
			CO5	Architect Suspicious Transaction Report (STR) and Cash Transaction Report (CTR) generation systems compliant with FIU-IND XML schemas with proper audit trails.
			CO6	Build a comprehensive RegTech platform integrating KYC, transaction monitoring, sanctions screening, regulatory

Academic Professor – I/C

HOD-BCA

Dr. K.BHAGAVAN
HOD BCA
KLEF, VADDESWARAM

S.No	Course Code	Course Title	CO. No	Course Outcome
				reporting, and audit trail across the customer lifecycle.
61	26CA33E4E	EXPERIMENTATION & CAUSAL ANALYTICS	CO1	Analyse the difference between interpretability (intrinsically transparent models) and post-hoc explainability to choose the appropriate approach for regulatory, debugging, and trust scenarios.
			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.
			CO3	Examine algorithmic fairness frameworks including demographic parity, equal opportunity, equalised odds, and individual fairness to evaluate model decisions across protected groups.
			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.
			CO5	Architect causal analytics workflows distinguishing predictive vs causal questions, identifying confounders, and applying observational causal methods to make actionable business or policy recommendations.
			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).
62	26IE3030E	CAPSTONE PROJECT	CO1	Apply requirements engineering, system design, and architecture decision-making to scope a 12-week buildable product aligned with the chosen specialization stream.
			CO2	Apply full-stack engineering practices — version control hygiene, branching strategy, code review, automated testing, CI/CD — across the build phase.
			CO3	Analyze cloud deployment options, scaling characteristics, and cost trade-offs to deploy the product on a managed cloud platform with infrastructure-as-code.
			CO4	Analyze security posture — OWASP Top 10, dependency vulnerabilities, secrets management, authentication/authorization — and apply remediations before launch.
			CO5	Evaluate observability, performance, and reliability through structured logging, metrics, traces, SLOs, and load testing.
			CO6	Evaluate the product through user testing, demo presentation, technical defense, and a published artifact (live URL + open-source repo + technical write-up).

Academic Professor – I/C

HOD-BCA

Dr. K. BHAGAVAN
HOD BCA
RLEF, VADDESWARAM