Capstone Project Topics and Engineering Expectations

School of Computing, B.Tech CSE

Vedaj Padman

Purpose of This Document

This document provides the official Capstone Project topic list for the Software Engineering course, grouped into complexity levels to help teams select projects that align with their readiness and ambition. The purpose of the Capstone is to demonstrate deep engineering ability—not surface-level application development. Projects must reflect thoughtful design decisions, architectural reasoning, documentation discipline, testing, deployment, observability, performance analysis, and responsible implementation. Students are expected to deliver technically rigorous systems that address real-world problems and demonstrate innovation, ownership, and originality.

Engineering Intent and Expectations

The Capstone represents a complete end-to-end software development journey, beginning with structured problem framing and concluding with a deployed, evaluated, and maintainable working system. The goal is to demonstrate serious engineering craft rather than UI assembly or template customization. Work that resembles cloning existing applications without original depth will not meet academic expectations. Every team must be prepared to justify architectural choices, assumptions, performance trade-offs, ML model usage where applicable, and security considerations.

Team Composition and Deployment Requirements

- Teams must consist of 4–5 members (six only with justification and approval).
- All projects must be deployed to the cloud (AWS, Azure, GCP, Render, Railway, DigitalOcean or equivalent).
- Continuous Integration / Continuous Deployment (CI/CD) pipelines are mandatory (Jenkins, GitHub Actions, GitLab CI, etc.).
- Version control history must reflect structured contribution: branching, merging, and meaningful commit narratives.
- On-premise deployment is permitted only with strong technical justification.

Sprint Review Structure

- **Sprint 0 — Inception and Foundation:** Problem framing, target stakeholders, personas, epics and user stories, initial architecture diagrams, technology stack justification, deployment and workflow decisions.
- **Sprint 1 — Core Implementation and Visibility:** Demonstration of approximately 40% working build deployed on cloud, initial testing evidence (unit/integration), monitoring mechanisms, and sprint backlog visibility.
- **Sprint 2 — Completion and Validation:** Near-complete functionality, testing coverage, performance evaluation, documentation, readiness for deployment, user documentation, and live demonstration.

Common Engineering Requirements

Every project must include:

- UML and architecture diagrams, ERDs, API specifications, and design justifications.
- Testing evidence: unit tests, integration tests, usability, accessibility, regression, and performance.
- Observability: system monitoring, logging, exception handling, uptime visibility, resource usage.
- Documentation: installation and deployment guide, release notes, usage manual, environment details.
- Performance considerations, scalability reasoning, trade-off justification and reflection.

Complexity Classification

- **Low Complexity (L1–L5)** — Institutional workflow systems, planning/optimization platforms, resource coordination, limited ML/security.
- **Medium Complexity (M1–M7)** — Systems with AI/ML, real-time collaboration, optimization algorithms, decision-support.
- **High Complexity (H1–H3)** — Compliance, security, cryptography-aligned reasoning, federated identity ecosystems.
- **Ultra High Complexity (UH1–UH5)** — Advanced distributed systems, zero-knowledge security, codebase intelligence, real-time architecture reconstruction, API observability platforms.

Low Complexity Topics (L1–L5)

L1 — Intelligent Academic Planner and Learning Activity Optimization Platform

This project focuses on developing an intelligent academic planning and productivity support platform designed to help students manage coursework, deadlines, examinations, and extracurricular commitments more effectively. Unlike simple scheduling or reminder applications, this system should integrate AI-based workload analysis, recommendation models that propose optimal study sessions, and behaviour analytics that help students recognize patterns such as procrastination or burnout. The platform may generate personalized learning plans based on academic intensity, time availability, and historical performance while preserving student autonomy and privacy. Ethical considerations such as transparency of recommendations, avoidance of manipulative nudging, and fair treatment of different learning styles should be central to the design. The system should demonstrate thoughtful UX design and accessibility to support a diverse student population.

L2 — Campus Resource Reservation and Utilization Optimization System

This project involves creating a campus resource reservation platform that supports structured booking of institutional facilities such as seminar halls, meeting rooms, laboratories, sports facilities, or specialized equipment. Rather than functioning as a basic reservation interface, the platform should provide analytics-driven insights into resource utilization and incorporate AI-based demand forecasting to optimize allocation, prevent resource monopolization, and ensure equitable access. The system could support approval workflows, conflict detection, and visibility into real-time resource occupancy. Sustainable and ethical design principles such as avoiding waste of underutilized spaces, promoting fair usage practices, and enabling transparent allocation policies should guide architectural choices. This project should address genuine institutional operational challenges and provide mechanisms to improve efficiency and fairness.

L3 — Indoor Navigation and Accessibility Assistance System for Large Institutional Environments

This project focuses on designing a navigation and accessibility assistance system to help students, staff, visitors, and differently-abled individuals locate classrooms, laboratories, departments, administrative offices, and event spaces across multi-building institutional campuses. Unlike mapping tools designed for outdoor navigation, this platform must consider indoor constraints such as multiple floors, interconnected corridors, limited GPS signal availability, and the need for accessibility-aware routing. AI-based shortest-path optimization, accessibility preference modelling, and optional AR-based visual guidance may be included. The system should incorporate inclusive design principles, supporting voice-based directions, icon-driven interfaces, and high-contrast visual layouts for low-vision users. Ethical considerations such as privacy-respecting geolocation practices and non-invasive tracking models should be reflected in design decisions.

L4 — Verified Lost-and-Found Tracking and Item-Matching System for University Campuses

This project involves developing a secure, structured lost-and-found management system for university campuses, replacing informal or paper-based processes. Users should be able to submit reports of lost or found items, including photographs, location, and time, while administrators oversee verification and claim approval workflows. The system may incorporate AI-based image similarity detection to suggest likely matches between reported and found items, reducing manual effort and increasing recovery success. Fairness and ethical considerations are essential, including transparent ownership verification mechanisms, anti-fraud handling policies, and privacy-conscious data usage practices. The platform should enhance trust and operational efficiency within the campus community rather than functioning as a basic listing system.

L5 — Smart Cafeteria Demand Forecasting and Token-Based Queue Optimization System

This project aims to develop an intelligent cafeteria management platform that uses data-driven insights to optimize food service operations, minimize waste, and improve student experience. Instead of implementing a simple food-ordering interface, the platform should allow users to prebook meal slots, forecast expected crowd levels, and receive predicted waiting times based on historical demand patterns, weather conditions, or academic schedules. AI/ML forecasting may support preparation planning to reduce waste, balance resource utilization, and improve sustainability. Token-based queue scheduling, real-time occupancy dashboards, and supply–demand analytics should be incorporated. Ethical and sustainability considerations—such as transparent allocation fairness, reduction of unnecessary food discard, and support for equitable serving—should drive system design and evaluation.

Medium Complexity Topics (M1–M7)

M1 — Accessible Language Learning Platform for Learners with Learning Disabilities

This project involves designing and developing a cloud-based language learning platform specifically tailored to support learners with cognitive, linguistic, or sensory learning disabilities such as dyslexia, ADHD, autism spectrum–related processing challenges, or auditory comprehension difficulties. The aim is not to produce a conventional lesson–quiz application, but to engineer a multi-modal instructional environment offering structured reinforcement through text, audio, visual cues, guided pronunciation, and collaborative learning spaces. The platform should incorporate meaningful accessibility and assistive features—such as dyslexia-friendly fonts and contrast themes, text-to-speech and speech-to-text support, adjustable pacing, distraction-free navigation, or layered scaffolding—demonstrating a thoughtful understanding of how software can address learning barriers rather than acting as a superficial checklist implementation.

To ensure rigor and depth, the project should incorporate advanced computer science elements such as intelligent recommendation engines, NLP-based automated evaluation of spoken or written responses, distributed real-time collaborative frameworks, or scalable microservices-based architectures. Authenticity is expected through original content design, localization to at least one Indian language where feasible, and justified architectural decisions. The project must demonstrate careful handling of accessibility, inclusivity, user experience, and pedagogical purpose, differentiating itself from generic language learning applications and signalling creativity, technical sophistication, and social relevance.

M2 — Intelligent Ride-Sharing and Sustainable Mobility Platform for Urban Transportation Efficiency

This project focuses on designing an intelligent ride-sharing and sustainable mobility platform intended to optimize transportation resources, reduce environmental impact, and address urban mobility challenges such as traffic congestion, inefficient vehicle utilization, and carbon emissions. The platform should enable ride pooling, optimized routing, dynamic resource allocation, and transparent coordination among riders, drivers, mobility operators, or public transit partners. Instead of replicating commercial taxi-booking systems, the system must demonstrate meaningful innovation by incorporating real-time route planning, demand prediction, congestion-aware scheduling, sustainability scoring, and usage analytics that promote environmentally responsible travel practices.

To achieve technical depth and relevance, students are encouraged to apply advanced computing techniques such as machine learning for demand forecasting, clustering algorithms for optimized ride grouping, GPS-based distributed routing logic, or gamification strategies that incentivize shared rides. The platform may integrate carbon footprint calculators, energy usage insights for electric vehicles, or collaboration features for campus or community mobility use cases. Attention must be given to security, trust, and data privacy for location sharing, along with thoughtful support for inclusivity and rider safety. The system should clearly differentiate itself from existing applications by presenting a compelling sustainability-driven perspective supported by technically sound architectural decisions.

M3 — Automated Timetable Scheduling and Faculty Workload Optimization System

This project involves developing an intelligent scheduling and academic workload optimization platform capable of automatically generating conflict-free institutional timetables while respecting real-world constraints. Rather than functioning as a static timetable entry interface or spreadsheet substitute, the system must model the complexity inherent in higher education scheduling—including faculty availability, program-level slot allocation, theory versus laboratory requirements, section groupings, elective overlaps, room capacity limitations, and institutional workload rules. The system should autonomously propose viable timetabling configurations, identify conflicts and resolution strategies, and adapt to updates such as faculty leave requests or the addition of new sections.

To achieve academic and engineering rigor, students are encouraged to apply computational optimization or heuristic techniques such as constraint programming, graph coloring, simulated annealing, genetic algorithms, or integer linear programming to generate optimal or near-optimal schedules. The platform may support scenario simulation, workload fairness analysis, multi-criteria optimization (e.g., commute minimization or balanced daily distribution of teaching hours), and visual analytics to compare generated alternatives. Attention should be given to transparency, user trust, and explainability of scheduling decisions, differentiating the system from simplistic CRUD-based tools commonly found online.

M4 — Real-Time Collaborative Digital Canvas and Creative Workspace

This project involves building a real-time collaborative digital drawing and sketching workspace that allows multiple remote users to co-create and interact with visual content simultaneously on a shared canvas. The goal is not to produce a basic offline paint application but to design a distributed, low-latency synchronization environment that supports multi-user editing with strong guarantees of consistency, responsiveness, and usability. Users should be able to draw, annotate, erase, insert shapes and text, manipulate layers, and collaboratively iterate on artwork, diagrams, or design concepts while observing smooth live updates from other collaborators regardless of network variation or geographic distance.

To support complex concurrent editing, students are encouraged to explore advanced real-time synchronization technologies such as Operational Transformation (OT), Conflict-Free Replicated Data Types (CRDTs), WebSocket-based messaging, or distributed state reconciliation strategies. Technical extensions may include timeline replay of drawing history, gesture recognition, AI-assisted enhancements such as stroke smoothing or auto-segmentation, version control and rollback support, or extensible plugins for exporting artwork to external media. The platform may include moderation controls, permission-based access for shared sessions, distributed storage for persistent canvas states, and protection against vandalism or accidental overwriting. This project requires balancing distributed systems engineering, HCI considerations, and performance optimization, demonstrating thoughtful innovation beyond simple GUI-based drawing tools.

M5 — Multilingual Digital Marketplace and Decision-Support Platform for Indian Farmers

This project focuses on creating a multilingual digital marketplace and decision-support ecosystem that enables Indian farmers to directly connect with institutional buyers, retailers, cooperatives, and community procurement groups without excessive dependence on intermediaries. The system must be carefully engineered to address real challenges faced by farmers, such as limited access to transparent pricing, lack of negotiation power, unpredictable demand cycles, and varying levels of digital literacy. Instead of functioning as a typical e-commerce listing portal, the platform should offer rich decision-enabling capabilities—price comparison dashboards, historical price visualizations, demand forecast models, quality-based price assessment, and structured negotiation workflows. The interface must support native Indian languages and multimodal interaction including voice-based navigation, icon-based user guidance, simple visual metaphors, and SMS/IVR support where needed, recognizing that many stakeholders operate in low-technology environments.

Technical depth may include AI-driven price prediction models, crop-based recommendation systems, location-aware logistics planning, or analytics on market trends. Features such as secure identity verification, auction or bidding models, dispute resolution processes, and trust-building mechanisms like structured ratings or verified seller profiles can enhance credibility and adoption. The platform may further incorporate sustainability and financial empowerment components such as subsidy awareness prompts, government scheme integration, crop advisory channels, or micro-financing partner interactions. The system should highlight thoughtful engineering values—including inclusivity, fairness, security, and usability—and demonstrate originality beyond superficial marketplace templates commonly published online.

M6 — AI-Enabled Crop Disease Diagnosis and Remediation Support System with Native Language Assistance

This project involves developing an intelligent agricultural diagnostic platform that assists farmers in identifying crop diseases early and recommending contextually appropriate remediation strategies based on crop type, environmental conditions, severity, and regional agronomic practices. Rather than producing a basic image classifier that labels leaf diseases, this system should operate as a holistic decision-support tool that integrates computer vision–based disease detection, real agronomy knowledge, and actionable treatment steps communicated through practical and accessible formats. Farmers should be able to upload photographs or short videos of affected crops, receive diagnosis insights with confidence indicators, and obtain clear, stepwise guidance that may include chemical or organic treatment options, prevention guidelines, and warnings on safety or environmental implications.

Recognizing significant variation in user literacy and accessibility, the platform must support native language interaction using voice narration, audio instructions, icon-based steps, and simplified visual workflows. Offline-capable operation and data caching may be considered for rural network constraints. The project may incorporate weather and soil condition data integration, geospatial outbreak mapping, risk forecasts, or community support channels connecting nearby farmers and agricultural experts. Advanced directions could involve federated learning models to protect farmer privacy while improving model quality, AI explainability features to increase trust, or ML pipelines retrained using regional datasets. The emphasis is on research-grade engineering thinking that meaningfully impacts agricultural resilience rather than superficial application construction.

M7 — Food Redistribution & Waste Reduction Platform for Surplus from Events, Restaurants, and Institutional Kitchens

This project entails creating a technology-enabled ecosystem to reduce food wastage by redistributing surplus but safe-to-consume food generated at restaurants, corporate and university canteens, hostels, banquet halls, and large-scale social events to NGOs, shelters, and community kitchens serving food-insecure populations. The work must go beyond a superficial listing or donation interface and instead address real logistical and public health challenges such as perishability constraints, hygiene validation, location-based pickup coordination, prioritization based on need and distance, and real-time status tracking. Donors should be able to upload quantity and type of surplus food within safety windows, while NGOs or volunteers are matched dynamically based on routing optimization, availability, and verified trust credentials.

To establish meaningful engineering depth, teams may explore intelligent supply–demand matching, volunteer dispatch algorithms, load-balanced routing, and predictive models that anticipate surplus patterns based on season, geography, or event types. The platform should support multilingual and multimodal interaction, including voice-based flows and simplified UI elements, ensuring usability among diverse stakeholders with varying levels of digital accessibility. Incentive mechanisms such as sustainability credits, certification letters, or transparent impact analytics—measuring meals recovered and CO_2 saved—may encourage sustained adoption. Data privacy, food-safety compliance workflows, accountability, and resilience must be treated as first-class design concerns, reflecting the humanitarian intent underlying this platform.

High Complexity Topics (H1–H3)

H1 — Secure Healthcare Information & Patient Management System with Regulatory Compliance

This project focuses on building a secure healthcare information and patient management system that responsibly handles sensitive medical data while demonstrating compliance-aware engineering grounded in real-world regulatory frameworks such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). The intent is to go substantially beyond basic appointment booking or hospital inventory software and instead engineer a robust domain-specific environment addressing access control, confidentiality, traceability, consent-driven data sharing, and ethical information usage. The platform should realistically model interactions among stakeholders such as patients, physicians, nurses, laboratory technicians, pharmacists, and administrators, enabling controlled access to medical records, diagnostic history, prescriptions, laboratory results, and treatment workflows while maintaining auditability and privacy guarantees.

Since medical information governance is complex, students are expected to translate legal principles into engineering decisions—incorporating ideas such as patient consent workflows, data minimization, secure encryption at rest and in transit, breach notification logging, pseudonymization or anonymization strategies, and least-privilege access models. Advanced functionality such as clinical analytics dashboards, AI-assisted decision support, interoperability across distributed hospital systems, telemedicine modules, or secure data exchange mechanisms may be explored. Emphasis is placed on demonstrating trustworthy system behaviour, resilience under realistic threat models, architectural clarity, and correctness rather than replicating existing software without depth.

H2 — Secure and Transparent Electronic Voting System with Verifiable Integrity

This project involves designing a secure, transparent, and tamper-resistant electronic voting system that upholds democratic principles of privacy, fairness, accountability, and verifiability. The objective is not to construct a simple polling or survey interface but to engineer a voting platform capable of ensuring voter anonymity while guaranteeing accurate vote recording and trustworthy result computation. The system should incorporate robust identity verification without compromising ballot secrecy, enforce protections against double voting and unauthorized participation, and include auditable trails that allow independent verification of final results without exposing individual voter selections. Stakeholders such as voters, election administrators, observers, auditors, and system operators should be clearly modeled with appropriate permissions and monitoring capabilities.

To ensure significant engineering depth, students are encouraged to investigate advanced security and cryptography mechanisms such as homomorphic encryption for encrypted ballot counting, digital signatures for ballot authenticity, distributed ledger or blockchain for immutable vote logging, or zero-knowledge proofs to balance transparency with privacy. The system may additionally support real-time dashboards for aggregated result visualization, mechanisms for recount and auditing, simulation of threat scenarios, and resilience against network-level and application-level attacks. Technical and ethical trade-offs must be articulated clearly, emphasizing architectural justification and risk-aware decision-making rather than reproducing a basic vote–count web utility.

H3 — Federated Social Networking Platform with Decentralized Identity, Privacy, and Interoperability

This project focuses on constructing a federated social networking system grounded in decentralization, privacy protection, and user autonomy—an alternative to centralized social platforms where a single entity controls data and policy enforcement. In a federated setting, independently managed servers (instances) interoperate while maintaining authority over their data, moderation strategies, and community rules. Students are encouraged to analyze existing federated ecosystems such as Mastodon, Diaspora, Friendica, or ActivityPub-based implementations to understand architectural patterns and limitations rather than attempting to replicate them directly. The platform should enable distributed identity management, secure cross-instance communication, and synchronized content exchange while addressing challenges such as conflict resolution, propagation delays, privacy-preserving message visibility, and ethical moderation.

Technical complexity may involve designing custom federation protocols or extending open standards, enabling end-to-end encrypted messaging, constructing distributed activity timelines, or implementing scalable community governance models. Research attention should be given to threats and failure modes such as misinformation, harassment, moderation bias, network partitioning, or malicious server operators. Students must justify architectural choices and protocol design trade-offs, demonstrating depth beyond producing a surface-level clone of existing open-source systems. The project should highlight how federation supports digital sovereignty, user safety, and resilient communication at scale.

Ultra High Complexity Topics (UH1–UH5)

UH1 — Unified API Debugging, Distributed Tracing, and Scenario Automation Platform (Beyond Postman / Hoppscotch)

This project aims to design a next-generation API engineering platform intended to address limitations in existing tools such as Postman, Hoppscotch, and Insomnia, which primarily support manual request testing but provide limited visibility into real-world distributed system behaviour. Modern microservices architectures often involve complex call chains where a single user request propagates across multiple internal services, message queues, background workers, and downstream APIs, making debugging slow, difficult, and error-prone. This platform should offer a unified environment that captures and visualizes complete end-to-end distributed traces, correlating service-level logs, latency breakdowns, failure points, and dependency structures. The system should enable developers to replay captured requests, simulate downstream failures, and compose automated multi-step API workflows that mirror realistic application execution paths, turning historically reactive debugging into proactive engineering.

To achieve meaningful differentiation, students should integrate capabilities such as automatic scenario generation based on observed network traffic patterns, built-in service mocking, load simulation, and protocol extensibility across REST, GraphQL, gRPC, event streams, and WebSockets. The platform may include collaborative workspace features for development teams, CI/CD integration enabling automated regression runs, privacy-aware handling of environment configurations, and visual dashboards that accelerate performance and resiliency tuning. Rather than replicating existing API clients, this system is expected to unify features often scattered across multiple tools—tracing, mocking, debugging, replay, and automation—into a coherent engineering experience that supports reliability, observability, and scalability in modern distributed systems.

UH2 — Cross-Platform Zero-Knowledge Password Manager with Secure Synchronization

This project focuses on designing a secure, zero-knowledge password manager that allows users to store and access sensitive credentials across multiple devices—such as mobile clients, browser extensions, and desktop applications—while ensuring that servers remain blind to the actual stored secrets. Unlike basic encrypted web apps that rely on server-side trust, this system must employ strong client-side cryptography for key derivation, encryption, and authentication, ensuring that no plaintext secrets are ever transmitted or exposed. Additional capabilities such as secure autofill assistants, password strength analytics, breach monitoring, and offline mode with later synchronization should be considered as part of a comprehensive security and usability approach.

The system must explore rigorous security engineering, including hardened key derivation functions, threat modelling for device loss or compromise, and encrypted conflict resolution during multi-device sync. Secure multi-factor authentication, encrypted storage of metadata, and secure clipboard handling may be incorporated. The platform should demonstrate real multi-platform support—not merely a single responsive web interface—and be validated through security-centric testing such as penetration testing simulations, cryptographic correctness evaluation, and usability evaluation under realistic constraints. The goal is to foster trustworthy credential management backed by defensible architectural choices and zero-knowledge guarantees.

UH3 — Real-Time Codebase Understanding, Impact Analysis, and Architecture Visualization Platform

This project involves designing a real-time visualization and code intelligence platform that enables software teams to understand, inspect, and analyze large-scale or polyglot codebases efficiently. Rather than producing static dependency diagrams or manually curated architecture charts, this system should dynamically compute structural relationships, call flows, service interactions, and ownership patterns across repositories. The platform should support visual exploration of the architecture through representation of modules, classes, functions, communication edges, dependency hierarchies, and change impact projections, helping developers reason about unfamiliar or rapidly evolving systems. Key considerations include multi-language analysis, scalability across large monorepos and polyrepos, and responsiveness to ongoing code changes.

The platform should extend beyond existing tools such as Sourcegraph and CodeSee by addressing pain points such as performance degradation with huge repositories, limited semantic understanding, and minimal support for architectural reasoning or change forecasting. Opportunities for innovation include automatic anomaly detection such as cyclic dependencies, dead code clusters, or high-risk hotspots based on commit activity; Git history mining for contributor analysis; semantic search and refactoring impact visualization; and CI-integrated real-time updates of diagrams upon commits or pull requests. Students should justify architectural trade-offs, visualization strategies, and data processing pipelines, demonstrating a balance between engineering performance, usability, and cognitive clarity.

UH4 — Real-Time Codebase Intelligence, Architecture Reconstruction, and Impact Analysis Platform

This project focuses on developing a real-time codebase intelligence and architecture reasoning platform that supports developers in understanding, navigating, and analyzing large-scale or polyglot codebases. Instead of producing static dependency diagrams or manually maintained architecture charts, this system should dynamically reconstruct architecture from the underlying code—visualizing module boundaries, dependency relationships, call flows, communication paths between services, and change-impact forecasts. The platform must help engineers gain deep insight into complex codebases, enabling tasks such as onboarding new developers, identifying risky components and hotspots, validating architectural adherence, and reasoning about the consequences of proposed modifications. Key objectives include scalability across large monorepos and distributed polyrepos, multi-language code parsing, and real-time responsiveness to continuous changes in the repository.

This project goes beyond the capabilities of existing tools like Sourcegraph or CodeSee by offering advanced semantic analysis, interactive exploration graphs, anomaly detection (including cyclic dependencies, unstable modules, or dead code clusters), ownership and contribution heat maps based on commit history, and diff-aware architectural impact visualization. Features such as semantic search, lineage tracking, refactoring assistance, and CI-driven architecture integrity validation can contribute to the project's engineering depth. Students should justify visualization approaches, underlying data processing pipelines, clustering and graph algorithms used, and trade-offs in architectural reconstruction fidelity versus performance.

UH5 — CI-Integrated Living Documentation System for Engineering Teams

This project aims to build a continuous documentation automation platform that eliminates documentation drift and ensures that engineering artefacts—such as architectural diagrams, sequence and flow charts, API references, README and onboarding guides, changelogs, and ADRs (Architectural Decision Records)—remain tightly synchronized with the evolving state of a codebase. Instead of functioning as a static documentation repository or relying on manual curation, the system should integrate with CI/CD pipelines, triggering intelligent documentation updates whenever code changes occur, and performing automated drift detection, consistency validation, and change summarization. The platform may generate documentation deltas, visual and textual summaries, and automated pull requests proposing updates, thereby reducing maintenance burden and increasing reliability and onboarding efficiency.

Advanced capabilities may include automatic generation of sequence diagrams, dependency or ER diagrams from actual code relationships; detection of documentation drift; suggestion or enforcement of architectural decision logs (ADRs); and creation of pull requests proposing documentation updates when required. Opportunities for innovation include NLP-based summarization of diffs, automated screenshot capture from running environments, versioned documentation dashboards showing history-aware comparison, and ML-based detection of undocumented or ambiguous components. Rather than functioning as a static wiki or Markdown generator, this platform should operate as an intelligent, self-updating documentation pipeline that strengthens engineering discipline, improves onboarding efficiency, and reduces operational risk.

Conclusion

The topics presented in this document reflect a commitment to high-quality engineering education, innovation, and real-world relevance. You are encouraged to choose a topic that motivates you and to approach it with rigor and curiosity. Successful Capstone projects will demonstrate originality, engineering excellence, thoughtful design, meaningful testing and evaluation, clear documentation, and responsible deployment. The goal is to produce work that you would be proud to showcase in interviews, portfolios, and future professional settings.