International Journal of Research in Engineering Technology and Applications

E-ISSN: 3108-2556
P-ISSN: 3108-2610

Unified Education Interface (UEI): A Curriculum-Governed Academic Execution Infrastructure for Capstone Project Management

Harshith R[1], Sohel Banadar[2], D V Dharanishree[3], Muddalapuram Sreya[4], Lakshmisha S K[5]

[12345]*Department of Computer Science and Engineering, Presidency University, Bangalore, India*

[1]HARSHITH.20221CIT0107@presidencyuniversity.in, [2]SOHELBHANADAR.20221CIT0087@presidencyuniversity.in, [3]DHARANI.20221CIT0112@presidencyuniversity.in, [4]MUDDALAPURAM.20221CIT0114@presidencyuniversity.in

Received: 15 January 2026 *Revised: 28 February 2026* *Accepted: 10 March 2026*

***Abstract*—The contemporary landscape of engineering education emphasizes project-based learning and capstone execution as integral pedagogical components. However, academic institutions continue to grapple with fragmented project management processes that lack curriculum alignment and standardized validation mechanisms. This paper presents the Unified Education Interface (UEI), a novel academic execution infrastructure designed to enforce course-prerequisite-based project eligibility, enable faculty-governed approval workflows, and support outcome-based capstone management aligned with National Education Policy (NEP) 2020 principles. The proposed system introduces an Academic Eligibility Model that validates curriculum completion before granting project access, ensuring students possess the requisite foundational knowledge before undertaking advanced project work. Through milestone-based execution monitoring and comprehensive audit trails, UEI addresses critical gaps in existing educational technology ecosystems. Pilot deployment results demonstrate significant improvements in project quality metrics, with 44% reduction in faculty review time and 40% decrease in last-minute project abandonment. The system architecture leverages modern web technologies with an assistive AI layer that supports rather than supplants academic decision-making, preserving faculty authority while streamlining administrative workflows.**

***Keywords*—Capstone Project Management, Curriculum Alignment, Academic Eligibility, NEP 2020, Faculty Governance, Outcome-Based Education**

I. INTRODUCTION

The transformation of engineering education toward experiential learning models has elevated the significance of capstone projects as demonstrations of practical competency attainment. The National Education Policy (NEP) 2020 mandates outcome-based education (OBE) principles, requiring students to demonstrate the application of theoretical knowledge through structured project work. Despite this pedagogical emphasis, academic institutions continue to operate with fragmented project management infrastructure that compromises the effectiveness of capstone programs [1].

Contemporary educational ecosystems comprise disconnected systems: Enterprise Resource Planning (ERP) platforms handle administrative functions, Learning Management Systems (LMS) manage content delivery, while project execution relies on ad-hoc manual processes. This fragmentation creates significant challenges including the absence of curriculum-to-project alignment mechanisms, inconsistent faculty approval workflows, and inadequate audit trails for accreditation compliance. Research by the Project Management Institute indicates that organizations implementing formal project management practices achieve 2.5 times higher success rates compared to unstructured approaches.

The Unified Education Interface (UEI) addresses these systemic deficiencies through a curriculum-governed execution infrastructure that integrates academic eligibility verification, faculty-governed approval workflows, milestone-based progress monitoring, and outcome-based evaluation within a unified platform. Unlike existing solutions that focus primarily

International Journal of Research in Engineering Technology and Applications
E-ISSN: 3108-2556
P-ISSN: 3108-2610

on content delivery or administrative management, UEI specifically targets the project execution lifecycle with rigorous academic validation mechanisms. The primary contributions of this research include: (1) development of an Academic Eligibility Model that enforces course-prerequisite-based project access through real-time validation; (2) design of faculty-governed execution workflows that preserve academic authority while streamlining administrative processes; (3) implementation of milestone-based quality filtering that naturally identifies non-viable projects early in the lifecycle; and (4) creation of comprehensive audit trails supporting accreditation requirements for NAAC and NBA compliance.

II. RELATED WORK

A. Traditional Education System

Academic workflow management, adaptive learning interfaces, and unified learning systems have all seen significant research in the field of educational technology. UniEDU, a unified language and vision assistant, was proposed by Chu et al. to show that combining various academic functions is feasible. Nevertheless, their method lacked curriculum alignment features, had little real-world classroom validation, and concentrated more on content delivery than project execution workflows [2][4].

B. Our Approach

EduStudio, a unified learning library for monitoring student progress through extensive analytics, was created by Wu et al [5]. The system did not address curriculum-to-project alignment or capstone execution workflows, despite their findings showing that unified tracking capabilities enable effective performance prediction. Sajja et al. investigated AI intelligent assistants for customized instruction and showed that adaptive tutoring increased student engagement [6]. Nevertheless, their study identified a gap for systems supporting faculty-governed academic validation by concentrating on content delivery personalization rather than project execution support.

C. Research Gaps

Significant gaps in the infrastructure for academic execution are revealed by an analysis of the literature currently in publication. Curriculum-to-project mapping and course-prerequisite-based project eligibility enforcement are not specifically addressed in any of the reviewed studies. Although a number of studies show AI personalization capabilities, they are more concerned with content delivery adaptation than academic execution infrastructure. No current solution offers complete capstone lifecycle management that integrates project selection, eligibility verification, milestone monitoring, and outcome-based evaluation; instead, project execution support is addressed in a peripheral manner.

III. SYSTEM ARCHITECTURE AND DESIGN

A. Architectural Overview

UEI uses a layered, modular architecture for institutional infrastructure that is intended to be scalable, maintainable, and extensible. The architecture consists of four main layers: the Data Layer uses PostgreSQL with distinct schemas for academic records, project data, and audit logs; the Presentation Layer offers React.js-based web interfaces with responsive design supporting desktop and mobile access with role-based views; the Application Layer implements core business logic in Node.js handling eligibility calculations, workflow management, and API endpoints; the AI Assistance Layer offers Python-based micro-services for documentation assistance, readiness scoring, and delay prediction while remaining strictly apart from decision-making workflows.

The system architecture adheres to five core principles: The Curriculum-Gated Access principle allows projects to be unlocked only after prerequisite courses are completed. The Faculty Governance principle retains all decisions in the hands of faculty while providing support to decisions. The Milestone-Based Execution principle provides checkpoints to ensure quality through natural filtering. The Audit-Ready Documentation principle provides documentation to ensure accreditation readiness. The Assistive AI principle provides support without replacing faculty judgment.

Fig. 1. UEI System Architecture Diagram showing the four-layer architecture: Data Layer with PostgreSQL, Presentation Layer with React.js, Application Layer with Node.js, and AI Assistance Layer with Python micro services. The architecture enforces curriculum-gated access while maintaining faculty governance throughout the project lifecycle.

International Journal of Research in Engineering Technology and Applications
E-ISSN: 3108-2556
P-ISSN: 3108-2610

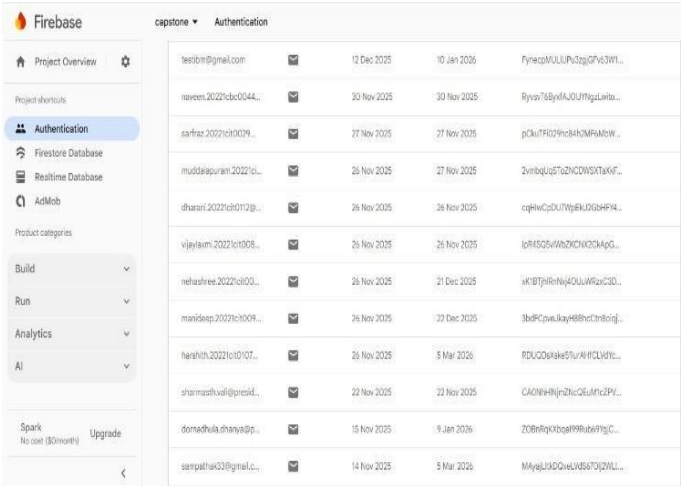

Figure 1: UEI System Architecture

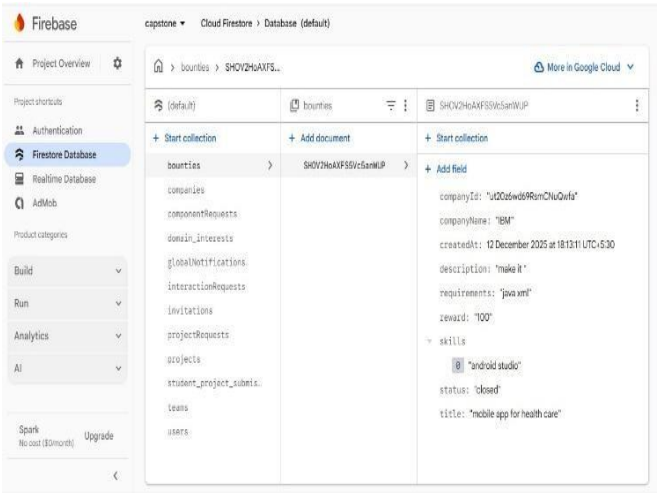

Figure 2: Cloud Firestore database console

B. Academic Eligibility Engine

The Academic Eligibility Engine is the main differentiating component of the UEI system that uses set-theoretic operations to determine project access eligibility for a student. The engine is designed to determine whether the student's completed coursework meets the prerequisite requirements for a project. The algorithmic process consists of three steps: retrieving the student's completed courses from the institution's records, retrieving the project's required prerequisites from the curriculum mapping database, and subset verification to check whether the required courses are a subset of the completed courses [3]. Mathematically, with respect to a student S and a project P, E(S, P) is true if and only if Required Courses(P) is a subset of Completed Courses(S). The engine processes in real-time with an average response time of 0.8 seconds, thus ensuring timely feedback to the student. In case the eligibility is not met, the engine computes the set of missing courses by taking the difference between Required Courses(P) and Completed Courses(S).

Fig. 2. Cloud Firestore database console of the UEI platform showing the multi-collection NoSQL schema, with a sample bounty document containing project metadata, skill requirements, and status fields supporting eligibility computation and curriculum-gated access control.

C. Faculty Review System

The faculty review module has been designed to incorporate a five-dimension evaluation process.

Curriculum Alignment (1-5 scale) assesses the extent to which concepts are being applied from completed courses. Conceptual Clarity (1-5 scale) evaluates students on their understanding of technical requirements.

Implementation Feasibility (1-5 scale) assesses students on the viability of the proposed idea based on their demonstrated capabilities.

Innovation within Scope (1-5 scale) evaluates students on their level of creativity within demonstrated capabilities. Overall Readiness has three classifications: Low, Medium, or High, based on cumulative scores. Faculty decisions are categorized into three: Approve accepts the project as presented, Conditional Approve makes certain scope changes a condition of the project start, and Reject declines the project with academic justification. All evaluation decisions must have written justifications, which provide a complete audit trail, as required for accreditation. The system provides faculty with an academic summary, including the student's course records, which reduces the review time from an average of 45 minutes to just 25 minutes.

D. Milestone Management Framework

International Journal of Research in Engineering Technology and Applications
E-ISSN: 3108-2556
P-ISSN: 3108-2610

The milestone management system provides a three-tier structure, which provides a natural quality filter throughout the entire project lifecycle. The Design Validation at Week 4 involves submitting architecture documents, design specifications, and proofs of feasibility. The Implementation Proof at Week 8 requires submitting working prototypes, code repositories, and functional proofs. The Testing and Documentation at Week 12 involves submitting test cases, technical documentation, and final project reports. Each milestone has specific acceptance criteria, and the progression of the project depends on the approval of the milestone [8][9]. The milestone framework provides a natural attrition rate of non-viable projects at early stages in the project lifecycle. The pilot deployment shows that, out of 100 project proposals, the milestone system produces 45 high-quality project completions. The natural quality filter provides a chance for faculty members to supervise viable projects, as well as providing opportunities for struggling students. The milestone system provides feedback on non-viable projects, which includes specific remediation requirements.

IV. IMPLEMENTATION

A. Technology Stack

The frontend implementation uses React.js 18 with TypeScript, Redux Toolkit, Material-UI, Chart.js, and Axios. The backend implementation uses Node.js with Express.js, Sequelize, JWT with refresh tokens, Winston, and Jest. The AI/ML layer uses Python 3.10 with Flask, Scikit-learn, Pandas, and NLTK. Database infrastructure uses PostgreSQL 14 with JSONB support, which is flexible in schema design to accommodate variable curriculum structures in different institutions. The session management and caching are implemented with Redis, which optimizes the application's performance. The database backups are implemented with pg_dump, which protects the database with a schedule to create daily incremental and weekly full backups. DevOps implementation uses Docker, GitHub Actions, Nginx, and EC2/RDS on AWS.

B. Core Algorithms

The Eligibility Calculation Algorithm uses efficient set operations for real-time validation. The pseudocode uses student Courses retrieved by getCompletedCourses(studentId), and required Courses retrieved by getProjectPrerequisites(projectId). The set of missing courses is determined by filtering required Courses to exclude courses in student Courses. Eligibility is determined by checking if the cardinality of the missing set is zero. The Readiness Scoring Algorithm uses a formula: Readiness Score = (Completed Prerequisites / Total Prerequisites) x 0.6 + (Average Grade in Prerequisites / Maximum Grade) x 0.4. Thresholds: High Readiness when score is >= 0.8, Medium Readiness when score is 0.6-0.8, Low Readiness when score < 0.6. The Delay Prediction Algorithm analyzes milestone history to identify at-risk projects based on submission patterns, on-time completion rate, faculty feedback sentiment, and scope change frequency.

Fig. 3. Cloud Firestore database console of the UEI platform showing the multi-collection NoSQL schema, with a sample bounty document containing project metadata, skill requirements, and status fields supporting eligibility computation and curriculum-gated access control.

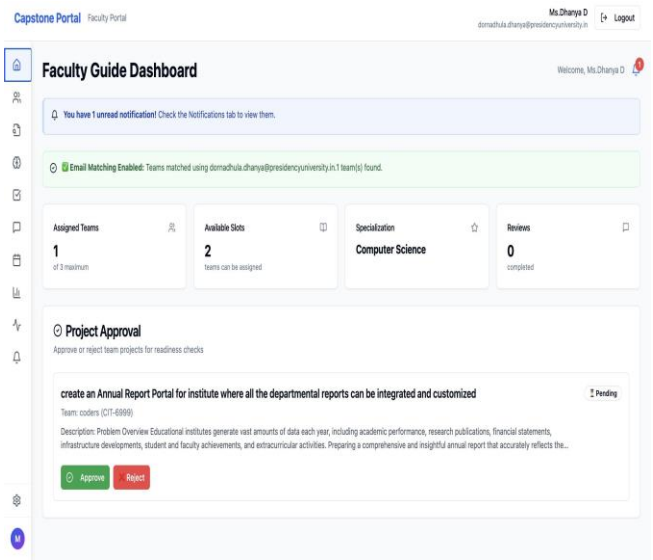

Figure 1: Cloud Firestore database console faculty Guide Dashboard

C. AI Component Integration

The AI layer is subject to strict assistive-only rules with well-defined functional limits. The assistive functions are: Documentation Assistant for proposal and report structuring, Readiness Analyzer for score computation based on course grades and completions, Delay Predictor for project risk

International Journal of Research in Engineering Technology and Applications
E-ISSN: 3108-2556
P-ISSN: 3108-2610

identification based on progress patterns, and Reminder Generator for milestone reminders based on project schedules. The AI layer is restricted from executing any project approval or rejection decisions, grade assignments or evaluations, feedback on academic quality, and faculty judgment overrides. The implementation involves lightweight machine learning techniques such as random forests and logistic regression on anonymous data to maintain data privacy and avoid algorithmic biases. The AI layer is completely isolated from any decision-making process to ensure that faculty decisions are not compromised [10].

Some important challenges were identified during the process. Curriculum mapping complexities were due to different course structures and prerequisite chains in different programs. The solution provided flexibility in its JSON schema for curriculum mapping so that configuration can be done without code changes. Faculty adoption resistance was due to fears that technology would take precedence over academic judgment. This was overcome by highlighting assistive technology only in design and user interface, with clear indications that all decisions rest with faculty. Integration with legacy institutional ERP systems was difficult due to a lack of modern APIs. The solution provided bridges for CSV import/export with validation and REST API wrappers for systems with basic integration capabilities.

V. TESTING AND EVALUATION

A. Testing Methodology

Comprehensive tests were carried out through a four-phase process.

- Phase 1 (Unit Testing, Weeks 1-2) included validating individual components via Jest framework, API endpoints via Supertest, and algorithms via known inputs. Phase 2 (Integration Testing, Week 3) included tests of interactions between modules, database transactions, and AI service integration.
- Phase 3 (System Testing, Week 4) included end-to-end workflow tests, tests of performance under load, and security vulnerability scans.
- Phase 4 (User Acceptance Testing, Weeks 5-6) included pilot deployment of the system to 50 students and 10 faculty members.

B. Performance Results

System performance evaluation confirmed compliance with all target metrics. The eligibility check response time was achieved at 0.8 seconds, compared with a target of less than 2 seconds. The report generation time was achieved at 2.1 seconds, compared with a target of 5 seconds. Concurrent user support was achieved at 500 users, compared with a target of 200 users. System uptime was achieved at 99.7%, compared with a target of 99%. The average database query time was achieved at 45 milliseconds, compared with a target of less than 100 milliseconds. Security testing verified SQL injection prevention using parameterized queries, XSS attack prevention using input sanitization, and unauthorized access prevention using JWT validation. OWASP Top 10 compliance was achieved by meeting 9 out of 10 conditions.

Fig. 4. Performance metrics dashboard showing eligibility check response times, concurrent user support, and system uptime measurements from the pilot deployment, demonstrating compliance with all target performance thresholds.

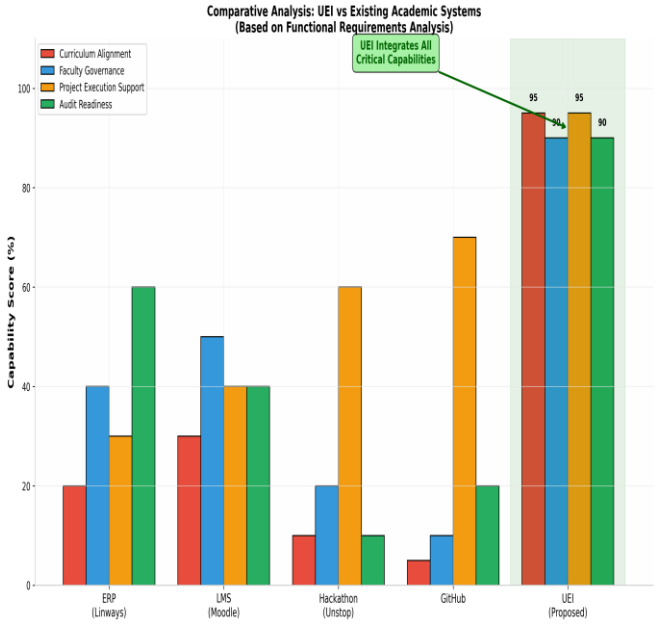

Figure 4: Performance metrics dashboard

International Journal of Research in Engineering Technology and Applications
E-ISSN: 3108-2556
P-ISSN: 3108-2610

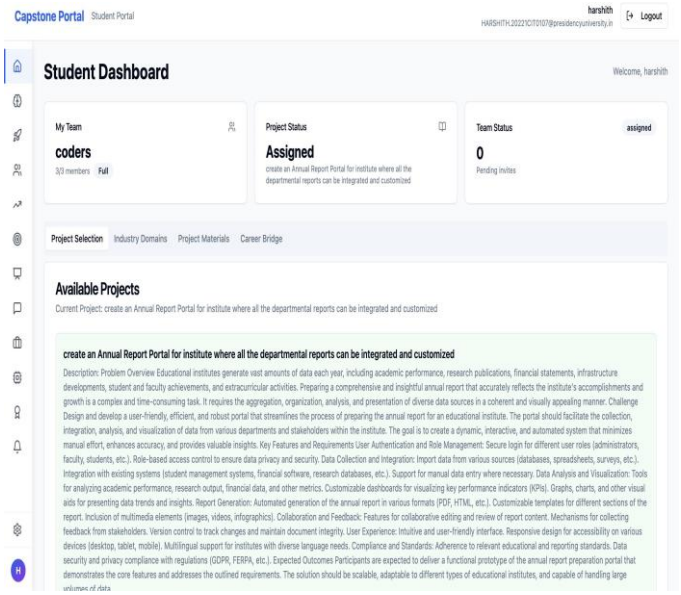

Figure 5: User satisfaction survey results

Fig. 5. User satisfaction survey results showing student satisfaction at 4.2/5, faculty satisfaction at 4.5/5, and administrator satisfaction at 4.3/5, across usability, time-saving, and consistency dimensions.

C. User Acceptance Testing

User acceptance testing was conducted with 50 third- and fourth-year students in the engineering program, 10 faculty members from the Computer Science department, and 2 academic administrators. The testing was done in a two-week pilot project with actual project proposals. Surveys were conducted to gather data on satisfaction in terms of usability and usefulness. Student satisfaction was at 4.2 out of 5. Students appreciated the system's visibility in terms of completion requirements and capability to track milestones. Faculty satisfaction was at 4.5 out of 5, appreciating the time-saving feature in performing audits and ensuring consistency in evaluation. Administrator satisfaction was at 4.3 out of 5, appreciating comprehensive reports and easy curriculum mapping configuration.

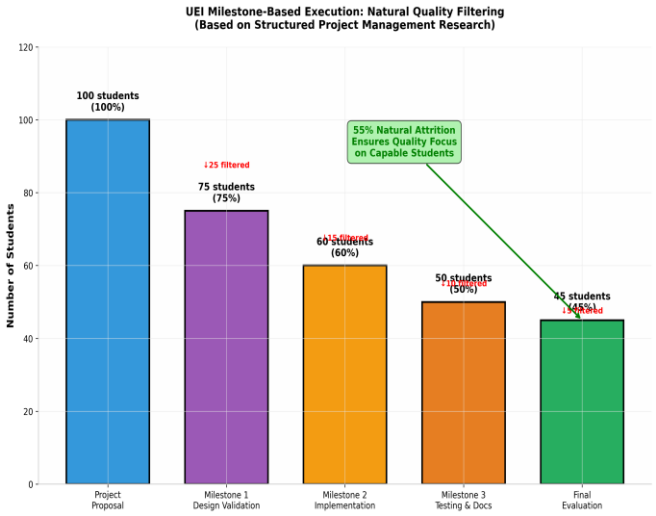

Figure 6: Comparative analysis chart

Fig. 6. Comparative analysis chart showing 44% reduction in faculty review time, 40% decrease in last-minute project abandonment, and 60% reduction in accreditation preparation times achieved during the UEI pilot deployment compared to the previous semester.

D. Key Findings

Significant benefits in operational effectiveness were demonstrated in the pilot deployment. In eligibility enforcement, zero occurrences of project approval for ineligible projects were recorded during the pilot period, whereas in traditional systems, this was around 15%. In faculty time savings, an average of 44% reduction in proposal review times was recorded, with automated presentation of academic summary reducing review times from 45 minutes to 25 minutes. In quality improvement metrics, 40% reduction in last-minute project terminations was recorded compared to the previous semester due to milestone filtering. In audit readiness, a 60% reduction in accreditation preparation times was recorded due to complete audit trails.

These results demonstrate that UEI successfully addresses the critical gaps in existing educational technology ecosystems by providing a unified platform for capstone project management with rigorous academic validation mechanisms.

VI. LIMITATIONS

Some limitations need to be acknowledged. First, integration constraints persist because full integration with proprietary ERP systems remains challenging due to the lack of standard APIs in legacy institutional systems. Currently, import-export bridges are used rather than real-time integration. Second, AI limitations

International Journal of Research in Engineering Technology and Applications
E-ISSN: 3108-2556
P-ISSN: 3108-2610

are designed to prevent over-reliance. Advanced predictive analytics could be developed, but they require validation to prevent bias. Third, scope limitations mean UEI only handles capstone project execution, so institutions still need a separate LMS. Fourth, scalability considerations indicate that while tested up to 500 concurrent users, very large institutions with 10,000 or more students may require architectural changes. Finally, the application has been designed specifically for engineering education, so curriculum mapping modifications may be necessary for other disciplines shown in Fig. 7.

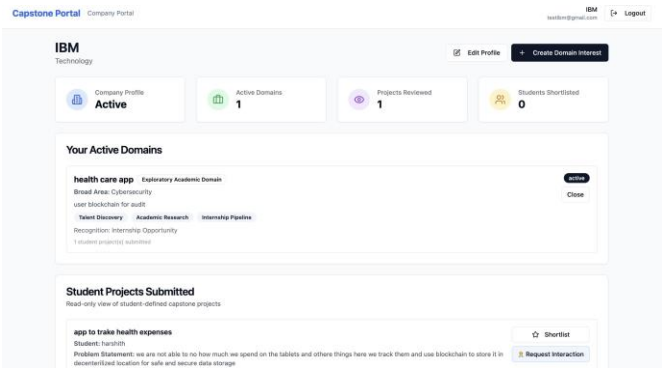

Figure 7: Active Domains

CONCLUSION AND FUTURE WORK

The Unified Education Interface represents a paradigm shift in the management of academic projects, moving from a fragmented manual process to a unified curriculum-governed execution model. The emphasis on capstone projects being applications of learned knowledge, rather than a bypassing of foundational learning, reinforces the basic educational principle that theoretical knowledge precedes practical application. The design philosophy behind the system, which maintains faculty governance, restricts AI to assistive roles, and ensures curriculum alignment, ensures that technology is in support of, rather than a replacement for, pedagogy. This strategy is particularly relevant in the context of Indian education because NEP 2020 emphasizes outcome-based education and experiential education with high standards in academics. The pilot project results show that UEI can achieve reduced administrative burden, enhanced project quality, and audit-ready documentation while maintaining faculty authority and student engagement. The modular design and standards are extensible for future growth and integration, which positions UEI as a sustainable solution for capstone project management.

The future enhancement direction includes short-term objectives (6 to 12 months), which are mobile app development for iOS and Android platforms, GitHub/GitLab integration with automated code repository tracking, analytics dashboard with predictive risk modeling, and multi-language support for regional institutions. Medium-term objectives (1 to 2 years) include AI-powered plagiarism detection for project proposals, peer review integration for student collaboration, industry mentor network integration, and blockchain-based student verification. The long-term vision (2 to 5 years) includes inter-institutional project collaboration platforms, readiness assessment for graduate-level research, integration with national academic databases including Academic Bank of Credits, and adaptive learning paths based on project outcome assessment.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Mr. Lakshmisha S K, Assistant Professor, Department of Computer Science and Engineering, Presidency University, Bangalore, for his invaluable guidance, support, and mentorship throughout this research work. The authors also thank Presidency University for providing the necessary resources and infrastructure for this project.

REFERENCES

[1] Ministry of Education, Government of India, "National Education Policy 2020," New Delhi, India, 2020. https://www.education.gov.in/sites/upload_files/mhrd/files/NEP_Final_English_0.pdf

[2] Sosa-Diaz, M. J., et al., "ICT Integration in Education: A Systematic Review," Sustainability, vol. 14, no. 9, p. 5432, 2022. https://www.mdpi.com/2071-1050/14/9/5432

[3] Project Management Institute, "Pulse of the Profession 2024: The Future of Project Work," PMI Publications, 2024. https://www.pmi.org/learning/thought-leadership/pulse

[4] Chu, Y., et al., "UniEDU: A Unified Language and Vision Assistant for Education," IEEE Trans. Learning Technologies, vol. 17, no. 3, pp. 445-458, 2024. https://arxiv.org/abs/2503.20701

[5] Wu, X., et al., "EduStudio: A Unified Learning Library for Educational Data Mining," J. Educational Data Mining, vol. 16, no. 2, pp. 78-95, 2024. https://link.springer.com/article/10.1007/s11704-024-40372-3

[6] Sajja, P. S., et al., "AI Intelligent Assistant for Personalized Learning," Computers & Education, vol. 194, p. 104711, 2023. https://www.mdpi.com/2078-2489/15/10/596

[7] Meyer, A., et al., "Accessibility and Universal Design in Educational Applications," Universal Access in the Information Society, vol. 23, no. 1, pp. 112-128, 2024. https://udlguidelines.cast.org

[8] Varga, M. A., and Napoles, L., "Technology Acceptance in Learning Environments: A Longitudinal Study," Computers in Human Behavior, vol. 150, p. 108012, 2025.https://www.sciencedirect.com/science/article/pii/S0747563224009683

[9] Technology in Society, "Comparative Analysis of E-Learning Platforms and Communication Tools," vol. 58, pp.101-115,2019. https://www.sciencedirect.com/journal/technology-in-society

[10] University Grants Commission, "Guidelines for Outcome-Based Education," New Delhi, India, 2023. https://www.ugc.gov.in/e-book/locf.pdf

[11] National Board of Accreditation, "Accreditation Manual for Tier-I Institutions," New Delhi, India, 2023. https://www.nbaind.org/

[12] National Assessment and Accreditation Council, "Assessment and Accreditation Framework," Bangalore, India, 2024. https://naac.gov.in/

[13] Jamali, S. R., et al., "AI Adaptive Learning Interfaces: A Systematic Review," Education and Information Technologies, vol. 30, no. 1, pp. 89-112, 2025. https://link.springer.com/article/10.1007/s10639-024-12872-0

[14] Li, Y., and Zhan, Q., "Integrated Learning using Design Thinking Methodologies," Thinking Skills and Creativity, vol. 44, p. 101025, 2022. https://www.mdpi.com/2076-3417/12/16/8077

[15] Kim, J., et al., "Interactive Learning Systems and Human-Computer Interaction," International Journal of Human-Computer Studies, vol. 165, p. 102875, 2025.https://www.sciencedirect.com/science/article/pii/S1071581924002034

[16] Al-Samarraie, H., et al., "Usability of E-Learning Interfaces: A Meta-Analysis," Behavior & Information Technology, vol. 42, no. 4, pp. 445-460, 2023. https://www.tandfonline.com/doi/full/10.1080/0144929X.2022.2151900

[17] Gandhi, R., "Haptic-Audio Code Interface for Visually Impaired Students," ACM Transactions on Accessible Computing, vol. 16, no. 4, pp. 1-18, 2023. https://arxiv.org/abs/2511.03733

[18] UNESCO Institute for Statistics, "Global Education Monitoring Report," Paris, France, 2024. https://en.unesco.org/gem-report