CS DEPARTMENT ERP

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE

Submitted by

PRAGYA PANDEY (2100290120121)

NILESH VERMA (2100290120114)

NISHANT (2100290120115)

PRATUL PANDEY (2100290120127)

Supervised by

MRS. ARTI SHARMA

ASSISTANT PROFESSOR

Session 2024-25

DEPARTMENT OF COMPUTER SCIENCE

KIET GROUP OF INSTITUTIONS, GHAZIABAD

(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India)

May 2025

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:-
Name:- PRAGYA PANDEY
Roll No:- 2100290120121

Signature:-
Name:-PRATUL PANDEY
Roll No:- 2100290120127

Signature:-
Name:-NILESH VERMA
Roll No:- 2100290120114

Signature:-
Name:-NISHANT
Roll No:- 2100290120115

Date:-

CERTIFICATE

This is to certify that Project Report entitled " **CS DEPARTMENT ERP** " which is submitted by Pragya Pandey (2100290120121), Nilesh Verma (2100290120114), Nishant (2100290120115) and Pratul Pandey (2100290120127) in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

Date: **Supervisor**

Mrs. Arti Sharma

(Assistant Professor)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor Arti Sharma, Department of Computer Science, KIET, Ghaziabad, for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only her cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kumar Srivastava, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date :

Signature:
Name: Pragya Pandey
Roll No.: 2100290120121

Signature:
Name: Nilesh Verma
Roll No.: 2100290120114

Signature:
Name: Nishant
Roll No.: 2100290120115

Signature:
Name: Pratul Pandey
Roll No.: 2100290120127

ABSTRACT

The Computer Science (CS) Department ERP project is an all-inclusive web-based solution, utilizing the MERN stack, and aimed at improving academic and administrative processes for the department. The web system targets key operational needs with online No Objection Certificate (NOC) application modules, competitive programming profiles and Faculty Development Program (FDP) direction. With both student and faculty user access, the ERP project is considering all aspects of a smooth and efficient process, where users can see status updates in real-time, and gain access to their academic records seamlessly.

The ERP system incorporates a range of features like an intuitive, user-friendly interface; secure role-based controls to protect data; and links to prominent coding platforms to display competitive programming accomplishments for students. The project is developed on the Agile Software Development Life Cycle (SDLC) model, and a more human-centered project management approach, utilizing modular design, scalability, and usability due to the iterative approach of Agile Style. The solution has calculated the best technical options for its delivery using React.js for the frontend, Node.js with Express for the backend and MongoDB for database usage.

The project supports Sustainable Development Goals (SDGs), specifically by enabling digital infrastructure (SDG 9), while also improving access to educational resources (SDG 4). In this case, the system improves the record management processes of the department efficiently from its digitalization, which enhances user accessibility. In the future, there is scope to expand the capabilities of the ERP to allow a mobile application for enhanced accessibility of educational resources, and to employ AI-supported analytics for academic insights. The project has been an important step towards achieving a digital transformation within the CS department through enhancements to efficiency, accessibility, and transparency of the academic space.

TABLE OF CONTENTS

Page No.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ERP	Enterprise Resource Planning
NOC	No Objection Certificate
FDP	Faculty Development Program
CP	Competitive Programming
DBMS	Database Management System
API	Application Programming Interface
UI	Transmission Control Protocol/Internet Protocol
Pdf	Packet Drop Fraction
CRUD	Create, Read, Update, Delete

SDG MAPPING WITH JUSTIFICATION

SDG 4: Quality Education

The ERP system plays a vital role in facilitating access to educational resources and academic processes within the department. With the automation of several tasks, such as the clearing of online applications for NOCs, the system saves on time and reduces errors so that students and teachers may invest their time in higher value activities. Further, the ERP showcases some core areas for students to promote their competitive programming profiles and accomplishments to foster participation in academic areas and non-academic arenas for their holistic development and better education experiences. Outside of this, efficient management and allocation of resources for the educational domain shall be accomplished within the department with the backing of educational software offered to support students in their learning endeavor and improving academic results.

SDG 9: Industry, Innovation, and Infrastructure

Thus, the implementation of a modern digital infrastructure using the MERN stack supports SDG 9 by fostering technological advancement and innovation within the department. Setting up a digital ERP from paper-based systems not only axiomatically may be considered a further step to reduce physical documentation but also makes for the greater operational efficiency within the department. This innovation equips the department with further advancement by preparing students and faculty with tools to work with some of the fastest-growing technologies in the industry. This system enables creating a smart environment mirrored after the industry, which drives the culture of technology adoption, hence encouraging the development of novel solutions within the department.

SDG 17: Partnerships for the Goals

The systems' ERP can allow other parties to collaborate with the educational department; namely, students, faculty, and industry mentors. In this project, a certain degree of knowledge-sharing and partnership-building has been realized, which is necessary to move the educational goals forward. By having features to include competitive programming profiles, the system links students to mentors and industry experts who can provide evaluations, growth, and career development. Furthermore, faculty can use the system to streamline communication with students, ensuring that educational goals are met more effectively. This collaborative approach not only benefits the students but also strengthens ties between the educational institution and the broader industry ecosystem.

CHAPTER 1
INTRODUCTION

1.1 Introduction To Project

The Computer Science (CS) Department ERP (Enterprise Resource Planning) Project is a sophisticated, web-based academic automation solution created using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The ERP system is put into place to simplify and digitalize academic and administrative workflows through an enterprise application. The ERP application purposefully tackles several important challenges, including but not limited to Faculty Development Program (FDP) tracking as well as management of, No Objection Certificate (NOC) requests, and competitive programming profiles. An ERP system provides a manageable, convenient way to help students and faculty; better manage important organization processes and transactions of academic records, and manage student requests. The application provides organizational efficiencies for both students and faculty by providing streamlined communication, visibility to all requests, tracking requests in real-time, and a single point of access for academic records.

Key Features and Functionalities

1. **Automated NOC (No Objection Certificate) Management**
   - Streamlines the submission, approval, and tracking of No Objection Certificate requests from students.
   - Eliminates manual paperwork and processing delays via automated workflows and notifications.
   - Creates transparency where students can track their application progress in real-time.

2. **Competitive Programming Profile Integration**

- Links to coding platforms (e.g., Codeforces, LeetCode, HackerRank) to retrieve shower students' competitive programming rankings.
- Provides a leaderboard, and analytics view, to measure student progress and recognize the best performers.
- Encourages students to participate in coding contests by visualizing their accomplishments.

3. **Faculty Development Program (FDP) Tracking System**
    - Assists with coordinating faculty training programs, workshops, and seminars.
    - Enables faculty members to make participation requests, submit certificates, and track professional development progress.
    - Administrators can produce reports about upskilling initiatives for institutional evaluation.

4. **Role-Based Access Control (RBAC) & Secure Data Management**
    - Implements multi-layered authentication (Admin, Faculty, Student) to ensure data privacy and security.
    - Limits access to sensitive information according to user roles, in order to prevent unauthorized changes.
    - Uses encryption and secure APIs to protect personal and academic data.

5. **Modular & Scalable Architecture**
    - Designed using a microservices-based architecture, allowing for easy integration of new modules later.
    - Supports more functionality such as attendance tracking, course registration, and exam scheduling with little reconfiguration.
    - Ensures a performant experience even with an increasing user load through extensive use of database indexing and caching.

6. **Agile Development & Continuous Improvement**
    - Created through the Agile Software Development Life Cycle (SDLC), with

a continuing iteration process in line with feedback from stakeholders.

- Regular sprint reviews and user testing were a significant part of the project success so that the system developed in alignment with the actual department's needs.
- Plans for future enhancements could introduce AI-based analytics supporting the ability to predict academic performance and the potential for a mobile app.

Alignment with Sustainable Development Goals (SDGs)

The CS Department ERP project coincides with worldwide sustainability initiatives by promoting:

SDG 4 (Quality Education) – Improving digital learning accessibility and administrative efficiency.

SDG 9 (Industry, Innovation and Infrastructure) – Utilizing modern web technologies to produce a scalable, future facing academic management system.

The CS Department ERP project has enhanced operational transparency, efficiency and data-driven decisions by transitioning from manual, paper-based processes to an entirely automated repository with a cloud-based ERP solution. The flexibility of the ERP design offering, with its secure access controls, scalable functionality and ability to integrate with various internal and external systems, will empower academic institutions to re-engineer internal administrative business processes as they near the end of the COVID-19 pandemic (moving to remote operations). Future integrations ranging from blockchain-based credentials verification and predictive analytics for student performance could further disrupt the educational technology landscape.

1.2 Project Category

The Computer Science (CS) Department ERP (Enterprise Resource Planning) Project is a niche project of an Academic ERP system that digitizes, automates, and optimizes the academic and administrative operations of the CS department. The ERP system is not a generic ERP system that serves the whole institution but is specific to the needs of a CS department and supports multiple core modules, including NOC management, program management of Faculty Development Program (FDP), and aggregated profiles of competitive programming on one web-based system. The system is built on a MERN (MongoDB, Express.js, React.js, and Node.js) stack, for scalability, real-time data processing, and a good user experience, which offers an effective means to raise productivity, collaboration, and open-ness of other departmental people who have a logical cause to use it.

Academic ERP Systems: A Specialized Solution for Educational Institutions

Academic ERPs are custom software platforms that replace traditional, paper-based, academic functions with automated, digital workflows. While most ERPs deal with entire colleges or universities, this project is different because we focus on a particular department with its own unique issues to address for their CS students and faculty. The benefits of using a departmental-specific ERP are:

- **More Relevant**: Unlike ERPs that cater to entire institutions, this ERP can be more precise in its features built around the real-world functions of a CS department; e.g. coder competition records and technical training programs directly related to career advancement
- **More Usable**: As we will remove the overpowering and non-applicable features, the interface remains clean, usable, and in a focused direction for the intended purpose
- **More Controllable**: Customizing at the departmental level permits updates at an unparalleled speed, easier troubleshooting, and the ability to assess direct feedback from end-users.

Core Modules and Their Impact on Departmental Efficiency

1. **Automated NOC (No Objection Certificate) Management System**

Problem: The primary issue with NOCs is that the entire NOC processing process is largely facilitated through forms, physical signatures, and lengthy approvals. This results in inefficiencies.

Solution: The ERP allows for the entire workflow process to be automated. Simply, students submit requests online; faculty reviewers are notified in real-time; and approvals are digitally recorded.

Benefits:

- Elimination of paperwork and reduced processing time.
- Audit logs for transparency in discerning decisions.
- Students can track their application status without needing to visit the department.

2. **Faculty Development Program (FDP) Tracking & Analytics**

Problem: Faculty struggle to document their training programs, workshops and certifications each year, which leads to having files out of order and losing information completely.

Solution: The ERP has an FDP module dedicated to Faculty which allows them to:

- Submit requests for participation
- Upload certificates and/or attendance confirmation reports
- Then use the reports to demonstrate compliance for accreditation.

Benefits: Makes documenting performance appraisal easier for faculty, faculty have documentation records, and the unit is able to track trends in skill development and training expenditures.

3. **Competitive Programming & Student Achievement Hub**

Problem: While CS students are actively engaged in different coding competitions (Codeforces, LeetCode, HackerRank, etc.), these achievements are fragmented across many platforms, and it is difficult to get recognition for them.

Solution: The ERP integrates with the coding platforms APIs to pull contest ranks, problem-solving stats, skill badges, etc, to automatically display:

- The rankings of contests taken by the student, problem solving stats, skill badges.
- A leaderboard to promote healthy competition.

Benefits:

- Identify top performers by faculty while looking for research/internship opportunities.
- Centralize programming student's coding profile which can enhance their portfolio.

4. **Role-Based Access & Secure Data Governance**

Problem: Academic information must remain secure, but still available to authorized users without bottleneck.

Solution: The ERP creates:

- Multi-tiered access control (Admin → Faculty → Student)
- Encrypted data at rest (MongoDB with field-level encryption)
- Automated backup / recovery to ensure no one has a data loss.

Benefits:

- Ensures sensitive academic records are protected and FERPA/GDPR compliant.
- Lessens IT burden of managing permissions manually.

Why a Department-Specific ERP Outperforms Generic Solutions?

Table 1.1 Features Comparison

Feature	**Generic Academic ERP**	**CS Department ERP**

Customization	Simplified, role-specific dashboards	**Tailored modules** for CS workflows (NOC, FDP, Coding Profiles)
User Experience	Complex interfaces with irrelevant features	**Simplified, role-specific dashboards**
Integration	May not support niche tools (e.g., Codeforces API)	**Built-in integrations** for competitive programming & research tracking
Scalability	Requires IT team intervention for updates	**Modular design** allows easy feature additions

This CS Department ERP provides an example of how niche Academic ERP systems can outperform one-size-fits-all systems by tailoring to particular user needs. It does so by:

✓ Automating redundant tasks (NOCs, FDP logs).

✓ Consolidating disparate sources of data (coding profiles, training records).

✓ Implementing security protocols without compromising usability.

By embracing this model, other departments can accomplish the same increase in productivity while minimizing administrative overload and instilling a data growth-oriented academic culture.

1.3 Objectives

The CS Department ERP system has been planned around the following main objectives to contextually shift how the department operates:

1. **Operational Efficiency Objectives**:
    - To unify and standardize the management of academic requests by creating dedicated digital workflows for NOC submissions and FDP applications,

effectively eliminating paper requests.

- To allow live request tracking and automated requested status notifications for all student and faculty requests through a dynamic dashboard interface.
- To function as a unified competitive programming integration module, that collects student achievements across different platforms, and provides both aggregated and visual representations of those achievements.

2. **Technical Implementation Objectives**:
   - To implement a multi-layered data protection schema that utilizes role-based access control (RBAC), which is based on roles with detailed permission settings.
   - To create a cloud based digital record keeping system, with advanced search and retrieval capabilities using MongoDB Indexing.
   - To create an automated FDP management subsystem that will accept submissions, process requests, and approve responses, and generate certificates, all with transaction visibility for all current requests.

3. **System Architecture Objectives**:
   - To develop an extensible microservice architecture that is scalable for additional modules in the future, such as research publication tracking, or internship management.
   - To create RESTful APIs that can integrate with existing university systems as well as any third party coding platforms.
   - To create responsive front for all components of the interface that appropriately scales for different screen sizes of devices.

4. **Strategic Alignment Goals**:
   - To support SDG 9 (Industrialization, Innovation, and Infrastructure) through the development of a departmental model for digital transformation.
   - To support SDG 4 (Quality Education) by increasing transparency and process efficiencies in academics.

- To create a model for ongoing improvement through embedded analytics and user feedback.

1.4 Structure of Report

The structure of this report follows a systems development methodology which sets out to document the entire ERP project:

1. **Chapter 1: Introduction**
    - Overview of the academic ERP space and departmental pain points.
    - Evaluative analysis of existing functionality and justification of custom development.
    - More detailed project objectives with identifiable success criteria.
2. **Chapter 2: Literature review**
    - Critique of 10+ academic ERP systems & their limitations.
    - Analysis of the use of MERN stack in educational systems.
    - Identification on the need for research on ERP solutions at the departmental level.
    - Evaluation of security frameworks to manage academic data.
3. **Chapter 3: Proposed Systems**
    - Architectural view with a set of component diagrams.
    - Detailed feature map for each module (NOC, FDP, Coding Profiles).
    - Innovations including real-time notification framework.
    - Data Flow Diagrams and interaction models.
4. **Chapter 4: Requirements analysis and system specification**
    - Exhaustive functional and non-functional requirements.
    - UML diagrams: use cases, class diagrams, sequence diagrams.
    - Database schema design (relationships).

- Security requirements & legal compliance

5. **Chapter 5: Implementation**
    - Rationale for the chosen Technology Stack with version information.
    - Implementation challenges by each module and how they were overcome.
    - The API development process and method of integration.
    - Techniques used to optimize the performance of the system.

6. **Chapter 6: Testing and Maintenance**
    - Testing strategy covering unit, integration and system tests.
    - Security testing approach which included the finding of a penetration test.
    - User acceptance testing method and results.
    - Maintenance strategy with how versions would be controlled.

7. **Chapter 7: Results and Discussion**
    - Quantitative measures of performance (response times and concurrent user capacity).
    - Qualitative feedback from users.
    - Evaluation against original objectives.
    - Screenshots of system interface showing functionality.

8. **Chapter 8: Conclusion and Future Scope**
    - Achievements and what was learned.
    - Proposed roadmap for Phase 2 development.
    - Possibility of integration with institutional learning management systems.
    - AI/ML uses in academic management can be more predictive analytics.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review

This document is a description of the college ERP system that was developed by VPPCOE students with the intention of making college operations run more efficiently online. In the abstract, the authors note that ERP not only acts as software but also a planning tool to achieve efficiency through informal and strategic means. The system is designed to minimize errors in the data entry and processing of students, departments, and faculties. The transitional objectives for the project include keeping management processes of college operations in the vast areas of college activities for recording student, department, and faculty related data quickly, consistently, reliably and flexibly. It replaces manual paper-based processes with a web-based application that integrates all college modules to address the complexities and issues concerning time-related barriers faced with the current system.

The paper investigates the present and dictated system structures, drawing attention to limitations of the manual system, such as decentralized data, data redundancy, and the difficulties updating or fetching data. To solve these problems, the suggested system provides several diagrams: use case and sequence diagrams, illustrating the various modules and processes that will be involved in the system. The brochure has several screenshots of results displaying the dashboards for users, events, homepage, profile, exam, and so on. It also proposes benefits of the system, such as a reduction in wasted time, increased productivity, and improved access to information for teachers and students.

This document also identifies references on college management systems and design, and recognises the contributors to the College ERP System project. It also states the minimum requirements of the project in specification of processors, memory, storage and software tools such as Xampp and IDEs to implement the project. The author concludes by discussing the successful implementation and ability to satisfy the current issues facing colleges, and eliminating unnecessary time and energy towards improving college operations.

In conclusion, the College ERP System aims to offer a reliable, user-friendly, and efficient alternative for colleges to avoid the shortcomings of a manual system, which will lead to a great change in the process of college administration. The system design, use case diagrams, and screenshots show that it builds a holistic framework for dealing with all aspects of the college administration, such as attendance, college finances, and lecture administration apart from admissions. The document serves as a well-rounded narrative of the system architecture, motivation, development process, features, possible influence on college administration, and acknowledgments and references for the successful establishment of the system.

The research examines the impact of Enterprise Resource Planning (ERP) on University of Jember employee performance. The research used a qualitative research design that included evaluative and descriptive research strategies to explore the interview data on how ERP impacted employee performance. The indicators of employee performance were based on work quantity, work quality, job knowledge, creativity, teamwork, reliability, initiative, and personal qualities. The results of the research demonstrated that ERP implementation at the University of Jember improved employee performance in quantity and quality of output, job knowledge, creativity, teamwork, reliability, initiative, and personal attributes of responders. However, the effects on initiative and cooperation varied. Overall, it was determined that the user performance at the University of Jember had improved after ERP implementation. The study also discussed the importance of management support and user training in the successful implementation of ERP systems. The main objectives of the ERP implementation were to maximize and leverage the institutions resources.

This paper comprehensively reviews the literature of the implementation of Enterprise Resource Planning (ERP) systems, with a focus on the importance of user involvement. It considers the requirement for user involvement in the implementation stage because ERP systems have a large impact on how employees work within an organisation. The complexities of information technology (IT) solutions and system integration across functional areas means that user involvement is critical to the acceptability and success of the implementation process. This paper highlights the different challenges with ERP implementations such as technological challenges, issues with human resources and possible causes of failure.

Further, the research reviews existing ERP implementation studies around critical success factors (CSFs) concerning user involvement and participation. The research provided several ERP systems success models regarding user influence, and also explored various organizational, technological, and people strategies for successful ERP adoption. The literature reviewed emphasized the importance of user involvement during the phases of identifying organizational needs and implementing ERP systems, because it allows an organization to customize the ERP system to its priorities and informational needs for conducting business.

The review acknowledges that user engagement in the development of ERP system has not widely been studied and most studies focus on traditional information systems and technical or logistical aspects. Highlighting the importance of user engagement, the study asserts the importance of user participation in order to mitigate resistance, consider change implications that ERP installation leads to, and facilitate more knowledge and feedback from users. It further reinforces the requirement of user engagement and participation in the implementation process, and emphasizes the human aspects that are critical to the success of ERP.

2.2 Research Gaps

Although nothing happens in the academic ERPs world these days, many areas of growth remain in current ERPs implementation that affects mainly computer science departments. The first important gap in current ERP implementation is discipline-specific functionality. With current ERPs, functionality reflects an academic modules without discipline specific features/functions to manage, and monitor competitive CA profile, and development NOC of faculty for technical development roles, and connection to coding devices such as LeetCode and Codeforces. The second key gap is in the area of processing, where the vast majority of system still rely on batch processing for processing an academic request. For many institutions experience, this has meant delayed NOC approvals and NOC status updates. The integrative functionality of modern ERPs continues to be particularly weak and few allow connecting via an API, to centralized developer platforms or crude data to show student accomplishment of competitive programming. User experience is a third important area of concern, since most existing ERPs emphasize administrative workflow functionality and neglect usability; meaning, both students and faculty face steep learning curves. Most existing ERPs use navigational structure and oftentimes require technical language that does not fit user experience for those potentially not familiar with the technology. In addition, current solutions do not provide sufficient mobile support, with many key features either unavailable or poorly designed for smartphone use. Accessibility standards also remain poor, with most implementations failing to comply with WCAG 2.1 standards for users with disabilities. Collectively, these deficiencies create significant barriers to successful digital transformation in computer science education. There is a clear need for improved solutions that are, most importantly, specialized, user-friendly and even more so, integrated.

2.3 Problem Formulation

The CS Department's ERP project seeks to address these important research gaps with a complete web-based solution specific to computer science education environments. Using the MERN (MongoDB, Express.js, React.js, Node.js) stack ahead of proprietary solutions, the system offers a suite of modules to avoid each of the limitations previously identified. The NOC management module transforms the approval process from a "set it and forget it" approval process to a real-time tracking, automated notification, and blockchain-based audit trail processes that meet the required reporting requirements more efficiently. For the technical skills component of the program, the system included a competitive programming hub, which aggregates data from multiple coding platforms using API integrations and allows users to visually analyze student performance and skills development over time. The FDP tracking system enables automation of credit calculations and are matched with appropriate programs. The reporting systems automates the compilation of information required for accreditation. Security is a key feature of the design's architecture, with multi-factor authentication and role-based access controls to protect the data while still meeting the Educational Privacy legislation requirements. The intended user interface is to promote usability by using adaptable dashboards to display content based on the user's role and preferences, use of contextual help systems, and progressive reveal of complex features. In terms of architecture, the system was designed with scalable containerized microservices to allow for the addition of future modules, such as research management, or tracking hardware resources. By integrating these functional features, the solution ultimately directly relates to three Sustainable Development Goals (SDG)s - SDG 4 (Quality Education) and the achievability of quality education through improved access to learning opportunities, and SDG 9 (Industry, Innovation and Infrastructure) through providing procedures that closely resemble those held in the author's modern academic world. The new system will be measured by benchmarking performance indicators such as a 40% reduction in administrative lead times, a 30% increase in FDP participation, and 90% positive user satisfaction rating when surveying department members on the practice, thus setting a new standard for the examples set by other ERP implementations in technical education.

CHAPTER 3
PROPOSED SYSTEM

3.1 Proposed System

The Computer Science Department ERP (Enterprise Resource Planning) system represents a paradigm change in academic management identified specifically to wrestle with the unique challenges faced by the modern CS (Computer Science) department. This web-based platform is carefully designed to provide enhanced functionality through intuitive design, giving rise to a comprehensive platform ensuring administrative efficiency while elevating academic performance. The system was developed with the newest MERN technology stack using powerful tools such as MongoDB's document storage, Express.js' web server framework, React.js for UI development, and Node.js for a powerful general-purpose javascript run-time environment. This software minimizes frustration that enterprises often face with erratic performance, offering resilient performance levels with absolute reliability.

A design philosophy driving the development of the ERP system provided a foundational understanding of the workflow and pain points within a department. Fundamentally, the NOC (Notational Operating Center) management module tries to transform what used to be a paper-based process into a digital workflow where students can request NOC's, track their approval status, and receive notification updates throughout the approval process. Noticeably, flow in processing times improves noticeably while creating a transparent audit trail that enhances engagement, responsiveness, and accountability for all interested parties. Approval methods for faculty enhance efficiency while integrating into their current schedules, reciting responsibilities.

The competitive programming integration is a novel feature that recognizes the growth of coding competitions within computer science programs that are educational in nature. Connecting directly with prime entities, including Codeforces, LeetCode, and Hackerrank, the system will automatically aggregate student students' accomplishments and convert them into appealing dashboards. This feature has multiple benefits including providing motivation for students through recognition of accomplishments, providing faculty with insights on student skill development, and building a structured record of departmental accomplishments for accreditation and visibility.

Faculty Development Program tracking will have equal attention to detail in the proposed system. The FDP module, which is designed intentionally to offer more than just tracking data, will include features such as automatic credit reporting, integrated program suggestions aligned with faculty research interests, and a link to the national and international conference databases. Departments will be able to use the analytics capabilities of the system to analyze trends in participation, identify gaps in skill development, and support data-driven decisions for future development programs. This new approach to FDP tracking will be a major improvement over current spreadsheet-based tracking that leads to inconsistencies and difficulties when managing reporting.

The system architecture has granted priority to security and access control. The system uses granular role based permissions allowing protection of sensitive data while allowing access to the appropriate users when required. For privileged operations, there is multi-factor authentication adding another layer of security. Furthermore, all data transmissions are encrypted using standard industry protocols. The design of the system ensures flexibility with security and allows administrators to change and adapt permissions as organizational hierarchies change.

From a technical standpoint, the microservices architecture model used to design the system has proven to offer superior maintainability and scalability. Each functional module is operated as a separate service that communicates across well defined APIs. This method of design has many benefits. Examples, include, performing updates of the system with there being no interruption to the system as a whole, improved logic separation realizing fault isolation, and ability to scale individual services depending on the demand / usage. Future expansion also becomes easier whether it be the addition of new modules or advances in technologies.

The user experience has been intentionally designed to promote usage by all user types. The interface is designed with responsive design techniques to ensure functionality is seamless regardless of device, from desktop workstations to mobile devices. Many intentional aspects have been built into the design to help reduce the learning curve while being able to access powerful functionality - strategy includes a contextual help system, progressive disclosure of any advanced features, and personalized dashboards. Accessibility considerations ensure International Standards are adhered to and modifications are made to allow everyone with specific needs to use the system as much as possible.

Development methodology also plays a significant role in the success of the system. The Agile approach in development aims to keep the development team in direct contact with end-users for the duration of the project. Each sprint cyclical process, allows for incremental development with the capability to incorporate ongoing feedback. This allows the final product to truly reflect the needs of the department rather than what was assumed in the first iterations. The methodology also provides the team with flexible capabilities to adapt components if the educational space continues to evolve.

Many of the decisions in designing the system were driven by environmental factors. The move toward a paperless organization reduces administrative overhead and also advances the institutional sustainability goals. Digital workflows eliminate the requirement for forms to be physically submitted and subsequently filed manually. Digital methods of acceptance through signatures or approvals speed up processes previously hamstrung by bottlenecks. The system has the potential to support sustainability initiatives even further through reporting, which can ultimately provide specific metrics on the reduction of paper usage and improvements in efficiency.

The ERP solution represents more than just a technology-first approach—it represents a reimagining of how academic departments can leverage digital technology to enhance their operations on a systemic level. The proposed solution allows us to target specific pain points with targeted solutions but also maintains the flexibility to respond to emerging or changing needs. In this way, the solution positions the CS department to be at the leading-edge of administrative innovation for academic departments. The combination of a targeted utility, strong architecture, and coherent design will provide a powerful tool that will satisfy today's needs, while at the same time allowing for continued enhance and growth in the future.

3.2 Unique Features of The System

1. **Modular Architecture:**
    - Componentized design using independent microservices.
    - New module plug-and-play.
    - Zero downtimes when modules are updated or added.
    - API driven integration architecture.
    - Isolated failure domains to keep the system up.

2. **Role-Based Access Control:**
    - Three layer permission structure (Admin/Faculty/Student).
    - Granular permissions for each user type.
    - Multi-factor authentication for sensitive operations.
    - Session timeouts.
    - Automatic logout.
    - Activity logging and audit trails.

3. **Real-Time Tracking:**
    - Live status tracking for all academic requests.
    - Visualized status along the workflow.
    - Track historic timelines of applications.
    - Deadline alerts and reminders.
    - Immediate notifications when status changes.

4. **Competitive Profile Integration:**
    - Auto-connect to Codeforces, LeetCode and HackerRank via APIs.

- Impressively low friction achievements.
- Dynamic leaderboards.
- Skill gap analysis.
- Performance trend visualization.

5. **Responsive Interface:**
    - Mobile-first design.
    - Flexible layouts for all screen sizes.
    - Touch controls.
    - Offline mode for main features WCAG 2.1 compliant accessibility.

6. **Scalable Back-end:**
    - Cloud-native architecture.
    - Load-balanced server architecture.
    - Automated database sharding.
    - Caching layer for frequent queries.
    - Pay-as-you-go data scaling capabilities.

7. **Automated Notifications:**
    - Multi-channel notifications (Email/SMS/In-App).
    - Notification settings.
    - Calendar-based notification reminders.
    - Urgent notification tags.
    - Read receipts.

8. **Data Analytics Dashboard:**
    - Interactive data visualization charts and graphs.
    - Ability to generate dry reports.
    - Predictive analytics models.
    - PDF/CSV exportability.
    - Real time databased refresh.

9. **Sustainability Features:**

- Full paperless workflow.
- Electronic document signing.
- Electronic documents archival.
- Resource usage monitoring.
- Carbon footprint reporting.

10. Additional features:

- Blockchain accreditation.
- AI-based search functionality.
- Voice command capabilities.
- Multi-language selection interface.
- Dark mode/light mode triggering.

Altogether, these additional features fundamentally alter the operations of the CS Department by supporting the integration of the next generation of technology with improved UX Design principles to create a truly modern academic management system. Each individual feature has been developed with usability in mind while being flexible enough for any unique department need in the future.

CHAPTER 4

REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

4.1 Feasibility Study

Technical Feasibility:

The CS Department ERP system is technologically feasible based on its selected tech stack and utilization plan. It is developed based on the MERN (MongoDB, Express.js, React.js, Node.js) technology stack. The MERN stack includes technologies that both possess proven and well developed and state-of-the-art features that allow for the development of complex storage of many types of data (student records, faculty development programs, etc.) using a document-oriented database, a framework that facilitates the building of robust web applications with a backend, tools to support dynamic websites and applications in terms of mobile responsiveness, and an efficient runtime environment optimized for an event-driven server-side development model to support our likely hours of peak concurrent access. It is also advantageous that the developers have extensive experience with the MERN stack, thereby mitigating the risks associated with the implementation process. In addition, the stack is cloud-native, facilitating hosting through well-established firms, such as AWS, Azure and Google Cloud. In addition to the MERN stack, the development team's other technical strengths include the use of containerization (aka Docker) to ensure a consistent deployment environment, implementing CI/CD pipelines to automate testing and deployment, and established monitoring practices and tools that ensure reliability and performance.

Economic Feasibility:

The economic analysis points to distinct cost-benefit advantages for the implementation of the CS Department ERP system. Because the solution is built on an open-source technology stack, there are no licensing costs for proprietary software packages. At the same time, we gain enterprise-level capabilities. The direct implementation costs consist principally of predictable and manageable costs for cloud hosting services, a domain name, and a few essential third-party integrations, and there is a clear ability to scale resources based on actual use patterns. The full cost projection indicates substantial long-term savings as a result of automating previously manual administrative processes, abolishing paper-based processes, and reducing labor-consuming activities. The modular approach to the ERP system allows for implementation in phases, which will help the department identify features that result in the most value and costs, as well as manage cash flow by staggering investments over time. The return on investment calculations focus on tangible benefits such as a reduction in operational costs, and intangible benefits such as increasing student satisfaction, increasing faculty productivity, and increasing the department's reputation by showcasing student achievements better. The economic model also considers the potential for the system to be adapted and reused by other technical departments, which would enhance the ROI for the university as a whole.

Operational Feasibility:

An operational assessment has demonstrated robustness in the system to be employed to support departmental activities and user capabilities. Furthermore, the solution adequately supports the alleviation of the major pain points present through dedicated modules for NOC functionality, competitive programming documentation and integration capabilities, and faculty development tracking. Role-based access control has been developed to mirror existing organizational forms and structures while ensuring the appropriate security protocols have been ensured. The user interface allows for great usability because of prototyping with real users, faculty and students, while developing the application as an overly delightful user experience for all user types regardless of technical proficiency. Training solutions at launch time will be numerous, including video tutorials, in-person workshops, and contextual help though the application itself. Change management considerations have been given via pilot-testing with select user cohorts and enabling and disabling new features incrementally. The system's operational design considers peak academic file cycles and has been optimized to perform well in busy periods such as registration windows and exam times. The platform leverages open feedback mechanisms to continue the development cycle based on the lived and felt experiences of actual users so that it can be molded to meet the needs of departments as the need may arise. The operational plan has included metrics along the deployment journey of the application, including user engagement and uptake, the amount of time saved, and significant declines in delays related to administrative bottleneck awareness.

4.2 Software Requirement Specification

4.2.1 Data Requirement:

- **Database System:**
  - MongoDB NoSQL database for flexible data storage.
  - Collections for user profiles, NOC requests, FDP records, and coding profiles.
  - Document-based storage for heterogeneous data types.

- **Data Types:**
  - Structured records for students (personal information, academic history).
  - Profiles for faculty (credentials, research interests).
  - NOC request documents and approvals process workflow.
  - FDP participation records and certification.
  - Competitive programming performance and achievements records.

- **Storage Requirements:**
  - Cloud storage solution (AWS/Azure).
  - Automated daily backups.
  - Data replication across availability zones.
  - Archiving strategy for historical records.
  - Storage capacity planning strategy for extensive growth plan over 5 year.

4.2.2 Functional Requirement:

- **User Management:**
  - ➢ Secure authentication Role-based (Admin/Faculty/Student) access to user management.
  - ➢ Initiate user profile management through web-based interface.
  - ➢ Password recovery procedure.

- **NOC Management:**
  - ➢ Web-based request submission.
  - ➢ Exploiting workflows for approvals.
  - ➢ Real-time tracking and follow-up.
  - ➢ Capability to submit documents.
  - ➢ Tracking of approval resolution history.

- **Competitive Profile Integration:**
  - ➢ API connections to current three coding platforms.
  - ➢ Data synchronization from coding platforms.
  - ➢ Dashboard to visually show achievements.
  - ➢ Skill reports based on activity logs.

- **FDP Tracking:**
  - ➢ Submitting digital records.
  - ➢ Credit calculation engine.
  - ➢ Track all certificates associated with FDP.
  - ➢ Reporting Deadline reminders.

- **Notifications:**
  - ➢ Email based alerts for changes in status.
  - ➢ Notifications through the app.
  - ➢ Allows user to configure notifications.
  - ➢ Read receipt process for notifications.

- ➢ Reminders through notifications on calendar.

4.2.3 Performance Requirement:

- **Response Time:**
  - ➢ CRUD functions <2 seconds response time.
  - ➢ Load time for dashboard viewing <1 second.
  - ➢ Query optimization for records in database.

- **Scalability:**
  - ➢ Support for 500+ concurrent users.
  - ➢ Horizontal scaling.
  - ➢ Load-balanced architecture.
  - ➢ Session control capabilities.

- **Uptime:**
  - ➢ 99.9% uptime SLA.
  - ➢ Disaster recovery plan.
  - ➢ Redundant infrastructure.
  - ➢ Scheduled maintenance.

4.2.4 Maintainability Requirement:

- **Code Modularity:**
  - ➢ Component-based React architecture.
  - ➢ Modular express.js routes.
  - ➢ A clearly defined API contracts.
  - ➢ Separation of concerns.

- **Documentation:**
  - ➢ Complete API documentation.

- System architecture diagrams.
- User guides (Admin/faculty/student).
- Deployment guides.
- Instructions for troubleshooting.

- **Version Control:**
  - Source control via Git.
  - Branching strategy (Git Flow).
  - Code review process.
  - Change logs.
  - Tagged versions.

4.2.5 Security Requirement:

- **Authentication:**
  - JWT authentication.
  - Session control.
  - Role-based permission.
  - Least privilege access.
  - Time-based access.

- **Data Encryption:**
  - TLS encryption for data in motion.
  - AES 256 encryption for data at rest.
  - Secure credential storage.
  - Input checking.
  - Prevention of SQL injection.

- **Monitoring & Compliance:**
  - Complete audit logs.
  - Reporting security incidents.

- Regular vulnerability scans.
- GDPR/FERPA compliant.
- Schedule for penetration tests.

- **Audit Logs:**
  - Encrypted backups.
  - Point-in-time recovery.
  - Backup verification.
  - Disaster recovery drills.
  - Data retention policies.

There is a full SRS that will outline all of the necessary data, functionality, performance, maintainability, and security requirements for the CS Department ERP system.

4.3 SDLC Model Used

The development of the CS Department ERP system is an iterative Agile process and follows seven defined phases with a goal of continuous delivery and improvement:

- **Phase 1 – Requirements Gathering & Analysis:**
  - Conducted stakeholder meetings to identify departmental needs such as NOC requests, competitive profile tracking, and FDP management.
  - Conducted stakeholder workshops and working sessions with faculty, administers, and students.
  - Conducted workflow analysis to identify pain points in existing manual processes.
  - Created detailed user stories for every module (NOC, FDP, Competitive Profiles).
  - Conducted competitive analysis of current offerings in academic ERP solutions.

  - Created success metrics and KPIs to assess the system.

- **Phase 2 – Planning:**
  - Created a prioritized product backlog categorized using MoSCoW (Must-have, Should-have, Could-have,).
  - Established sprint teams with cross-functional team members and assigned Scrum roles.
  - Evaluated technologies and established a MERN stack.
  - Established 2-week sprints with defined deliverables.
  - Established CI/CD pipeline infrastructure.
  - Created risk mitigation strategies and contingency plans.

- **Phase 3 – Design:**
  - Created a full system architecture diagram.
  - Created core module design utilizing microservices.
  - Created a detailed database schema with MongoDB collections.
  - Developed interactive UI/UX prototype for user validation.
  - Created API contracts and data flow specifications.
  - Organized design review sessions with stakeholders.
  - Finalized and established security architecture and access control matrix.

- **Phase 4 – Development:**
  - Implemented the core functionality sprint by sprint (in priority order).
  - Frontend development in React.js with Redux for state management.
  - Backend services in Node.js and Express.js MongoDB with Mongoose.
  - ODM RESTful APIs for each module Real time capability with Socket.io.
  - Continuous integration with automated builds Code review and pair programming sessions conducted.

- **Phase 5 – Testing:**
    - Unit testing with Jest and Mocha as the testing framework.
    - Integration testing of API endpoints and services.
    - End-to-end testing using Cypress as the end-to-end testing framework.
    - Performance testing with JMeter Security testing including penetration tests.
    - Accessibility testing for WCAG compliant.
    - User acceptance testing with users from the department Regression testing after each sprint.

- **Phase 6 – Deployment:**
    - Deployment in Containers with Docker Cloud hosting on AWS with auto-scaling.
    - Blue-green deployment strategy.
    - Monitoring configuration with Prometheus and Grafana.
    - Logging configuration with the ELK stack (Elasticsearch, Logstash, Kibana).
    - Roll out to user groups in phases.
    - Training and documentation to end user departments.
    - Helpdesk and support channels established.

- **Phase 7 – Maintenance:**
    - Security patches and updates are routinely employed.
    - Performance tuned to usage statistics.
    - Future feature improvements planned quarterly.
    - User feedback is gathered and analyzed.
    - Technical debt is managed.
    - Disaster recovery drills are practiced.
    - Capacity planning done for scale.
    - Documentation is reviewed and updated.
    - Community building and knowledge sharing occurs.

By using the Agile Process of Management for each of these aspects, the ERP system can continuously evolve with the changing needs of the department. Simultaneously, the ERP system is delivered, complete with a sense of high quality achieved through ongoing testing and through the incorporation of user feedback. After every iteration, there is a complete working piece of software that is delivered incrementally, allowing the organization to realize value before the final contract is complete.

4.4 Data Flow Diagram

- **Level 0 (Context Diagram):** The highest-level overview of the system architecture illustrates the manner in which diverse user groups engage with the core ERP modules. Students start the flow of the project by creating and submitting NOC requests via web forms while they also keep their competitive programming profiles up to date with integrations from the platforms. Faculty members use the system through workflows for approval of academic requests and through interfaces for the management of FDP records. Administrative staff utilize analytical dashboards for system configuration activities and for reporting actions; and external systems like the coding platforms (LeetCode, Codeforces) and email act as the supporting entities, demonstrating the integration points that exist within the ERP. All data exchanges occur through secured API gateways to control and enforce authentication protocols.

- **Level 1 (Process Decomposition):** This level of view posts the system's logic as a defined set of steps. There are 4 primary steps. Process 1 is the user auth process - authenticating the user based on their credentials stored in the MongoDB (as encrypted records) and applying a set of role-based access rules. Process 2 is the NOC process which contains steps from the moment a student submits the request (along with uploads of documentation) through the review process conducted by faculty and the routing of the approvals through to the closure of the submitted request. Process 3 is the competitive profile process which periodically collects competitive programming data from the APIs of their coding platforms, processes that information, and reports that data back to the student's dashboard; and Process 4 is

FDP process which maintains a record of where the deadlines are in student submissions and confirming to the students that their completion certificates were issued with the amount of professional development credits assigned to it. Each of these primary processes place an entry into the database for auditing of the user actions and sends the appropriate notifications.

- **Level 2:** The most technical diagram deconstructs Process 2 (NOC workflow) into atomic operations. The submission sub-process checks requests for completeness, assigns unique tracking IDs and creates pending entries. The routing sub-process applies business logic to identify approvers based on request type and the organization's existing department policies. The approval sub-process captures digital signatures, timestamps where possible and updates the request status. Meanwhile, the notification sub-process watches for state changes, generates messages (on approval or rejection of a request or when a request is clarified), and sends them down configured channels. The data stores are shown explicitly to show how MongoDB collections for users, requests and activity logs populate while these processes are happening.

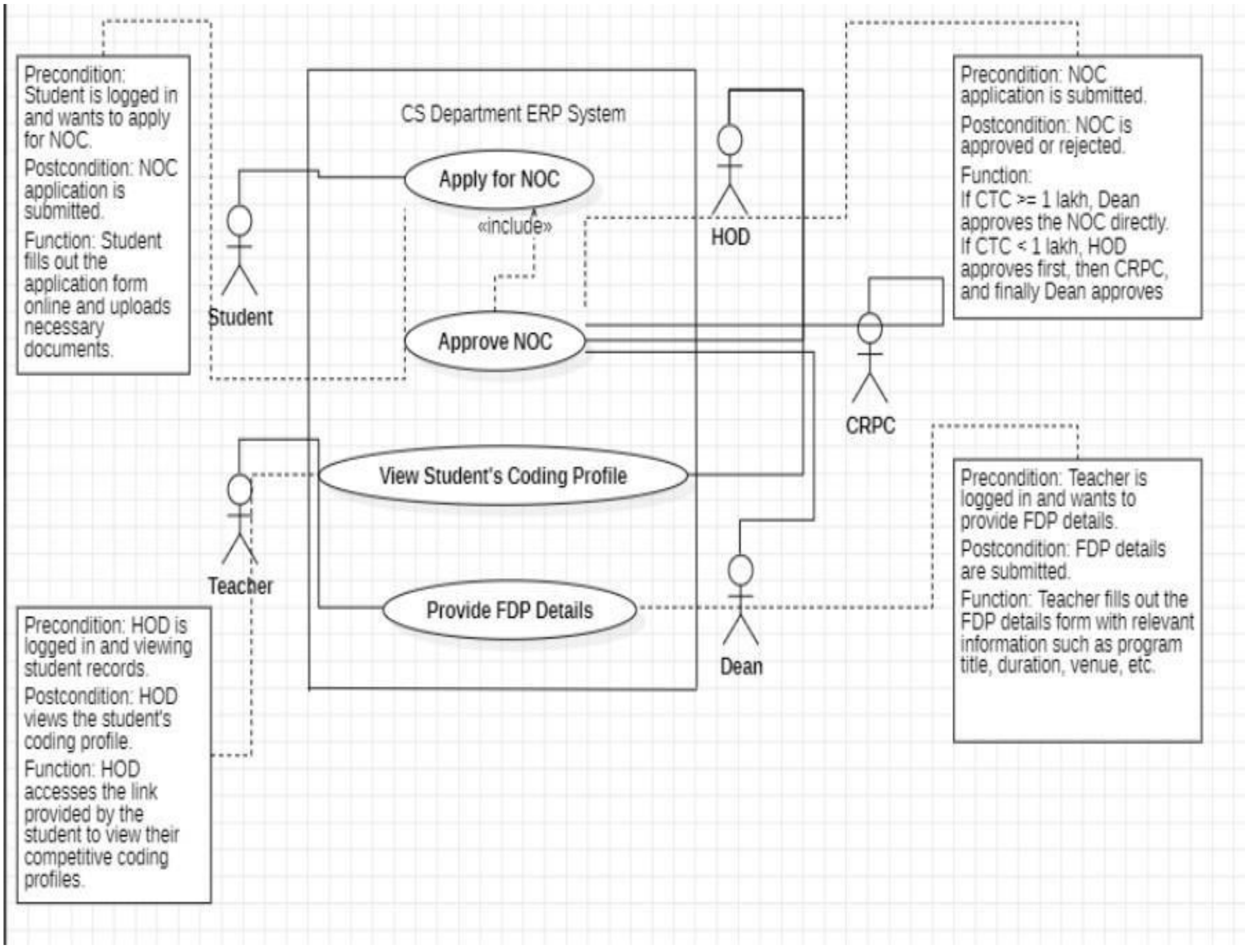
4.5 Use Case Diagram

- **Actors Specifications:** Three actors are to be the basis of the three actors' activity that you saw in the use case. All three actors have a different level of privileges. Students log in using institutional credentials and are able to self-service their records; students can only access their own records, they cannot be granted access to anyone else's records. Faculty members will have their accounts vetted and can assume a higher privilege level with multi-factor authentication enabled; faculty accounts are able to access all students supported by them in an academic term, as well as the applicable department functions. Lastly, admin users are vetted through a series of steps, cannot use the service without identification and heightened logging takes place (where available) and access is limited to only prescribed workstations.

- **Use Cases:** The student "Submit NOC Request" use case starts with automatic form validation; a system exam is executed - the system validates eligibility against the students' academic records previously verified and accepted for submission. The "View Competitive Profile" is an aggregation of several coding platforms' repositories; the sequence applies a normalizing algorithm to present uniform metrics of skill. The faculty are involved with the use case "Review NOC Request", which displays decision-support information, i.e., academic standing and historical submission requests involving the same students. The "manage FDP records" use case has prescriptive suggestions for meaningful professional development opportunities depending on their teaching loads and research interests.

- **Extended Functionality:** The Administrative role has the "Configure Approval Workflows" use case that visualizes multi-level approval chains with conditional routing. The "Audit System Activity" use case has forensic tools with temporal range filters and alerts for anomalies. All base use cases have specialized extension points; as an example, the NOC approval sequences a faculty have the optional extension to "request clarification" before making a final determination.

- **Interaction Dynamics:** The diagram is intended to show the critical relationships via stereotype notations. The «include» relationships shows mandated relationships like authentication must occur prior to any modification to data. The «extend» relationships depict conditional flows; like notification generation but only after a status transition. The generalization relationships indicate a base use case versus a specialized case; for example, "View Profile" is a base case, whereas "View Enhanced Faculty Profile" is a specialized case. All interactions respect the least privilege principle; therefore guard conditions are explicitly noted in the annotations of the diagram to represent RBAC policies.



Precondition: Student is logged in and wants to apply for NOC.

Postcondition: NOC application is submitted.

Function: Student fills out the application form online and uploads necessary documents.

Precondition: NOC application is submitted.

Postcondition: NOC is approved or rejected.

Function:
If CTC >= 1 lakh, Dean approves the NOC directly.
If CTC < 1 lakh, HOD approves first, then CRPC, and finally Dean approves

Precondition: Teacher is logged in and wants to provide FDP details.

Postcondition: FDP details are submitted.

Function: Teacher fills out the FDP details form with relevant information such as program title, duration, venue, etc.

Precondition: HOD is logged in and viewing student records.

Postcondition: HOD views the student's coding profile.

Function: HOD accesses the link provided by the student to view their competitive coding profiles.

Figure 4.1 Use Case Diagram

4.6 Database Design

MongoDB NoSQL Database Implementation

The system leverages the document model inherent in MongoDB to store a variety of academic data types while guaranteeing performance and scalability. The design of the system takes full advantage of MongoDB features that include sharding for horizontal scaling and replica sets for high availability, so that there will be no interruptions during active academic session times.

Comprehensive Collection Structure

1. **User Profiles Collection-**

   - Houses full identity management information for each system user.
   - Personal profile details, authentication method, and role assigned.
   - Represents Department affiliation and account status signs.
   - Data related to the patterns of user activity as determined by time-stamped logins.

2. **NOC Request Management-**

   - A complete history of No Objection certificates.
   - Records details of submissions including request types and evidence documentation.
   - Records the entire sequence of approval and workflow stages through an audit trail.
   - Houses digital or scanned versions of original documents submitted and pre-approval documentation.

3. **Competitive Programming Repository-**

   - Compiles student accomplishments across multiple coding platforms.
   - Standardizes performance indicators from various judging systems.
   - Provides custody of records on progress over time/performance.
   - Enables talent recognition through standardized rankings for display.

4. **Faculty Development Records-**
   - Records professional development and training activities and certifications.
   - Records participation in workshops, seminars and training.
   - Stores documents like certificates and competency badges.
   - Accumulates all professional development credits.

5. **System Activity Log-**

    - Supports full security audit capabilities.
    - Records all major activities with context.
    - Supports forensic investigation and compliance reports.
    - Allows analysis of usage pattern to inform future systems development.

Data Relationship Framework:

The database manages relationships with expertise through:

- Mapping Relationships in Academia:
    - Students are able to have active.
    - NOC requests and full histories retained.
    - Faculty development activities are able to account for institutional accreditation requirements.
    - Competitions programming profiles can be attached to curriculum outcomes.
- Transactional Integrity:
    - Reference constraints are provided as consistency between collections.
    - Atomic operations are present to maintain data validity across multiple related steps.
    - Document embedding can be utilized where related data is often accessed together.

Performance Optimization Features

- **Strategic Indexing**:
    - Multiple critical query paths can use high performance compound indexes.
    - Any content can use specialized text indexes for searching.
    - Analytic indexes can be geo-spatial indexed to analyze regional metrics.

- **Optimizing Queries:**
  - Aggregation pipelining can be used for complex reporting.
  - Alternating the read preference can help load balance workload.
  - Minimizing document transfer using projections.

Security Implementation

- **Data Protection**:
  - Field-level encryption for secure information.
  - Role-based access controls to collections.
  - Automated masking of data for use in normal operations to comply with privacy legislation.
- **Operational Security**:
  - Audit logging.
  - Change data capture for critical collections.
  - Point-in-time recovery increments.

Scalability Architecture

- **Horizontal Scaling:**
  - Sharded cluster configuration.
  - Allocation of workload.
  - Elastic scaling of capacity.
- **Data Distribution:**
  - Awareness of data location.
  - Tiered storage strategy.
  - Intelligent ageing policy.

This database architecture is the underpinning for all functions of the ERP system, and provides the performance, resiliency, and flexibility the modern academic institution needs, while achieving the required data protection. The design supports both current workload and future growth via an extensible and scalable architecture.

CHAPTER 5
IMPLEMENTATION

5.1 Introduction Tools and Technologies Used

The CS Department ERP system utilizes an integrated technology stack aimed to provide performance, scalability, and maintainability. The tools and frameworks used were intentionally selected to meet the requirements of an academic administration system while considering overall developer experience.

Frontend Development Stack:

The user interface is implemented using React.js, an effective JavaScript library that allows developers to build responsive, dynamic, component-based interfaces. For user experience, React's virtual DOM implementation uses custom and built-in APIs for better rendering performance as DOM traversals are abstracted away for RDMS-style queries. For managing application state across the workflows of complex applications like an ERP, we used Redux (which implements a predictable state container) to ensure consistent state across the application. The user interface makes use of Tailwind CSS as a utility-first CSS framework that allows for developing responsive designs rapidly, while still enabling capability for customization. Other front-end technologies rely upon include Axios (for making HTTP requests), React Router (for navigation), and Formik (for form handling).

Backend Infrastructure:

The back-end portion of the architecture runs on Node.js, which uses a non-blocking I/O model that can handle concurrent requests concurrently without blocking. The web application server uses Express.js as a routing and web framework, which provides a routable API architecture and supports additional middleware. The back-end uses REST API principles, including proper APIs, HTTP response codes, and resource-oriented endpoints. The API also handles maintenance of any computationally burdensome requests in worker threads, ensuring they don't block the core event loop of Node.js, which is used to maintain the concurrent I/O model. Additional API management features such as validation of data to limit bad requests and rate limiting are provided with additional rate-limiting and data validation middleware.

Database Ecosystem:

MongoDB Atlas is the primary cloud-based database chosen to be flexible with automatic scaling, automatic back-up, and wide geographical access for scalable backups. The document interface allows for a variety of academic data structures with flexibility in queries. Mongoose is used as the Object Data Modeling (ODM) in Node.js to provide schema validation and reservation for business logic hooks. Understanding good database usage is a learned feature through indexing of the database, profiling queries, and building aggregation pipelines for more complex reporting.

Development Environment:

The primary IDE is Visual Studio Code with extensions added for code quality (ESLint), code formatting (Prettier) and development tools for React. The development workflow allows for hot module replacement to iterate on frontend development quickly, and nodemon to restart the backend automatically. Postman is used for API testing with collection runners and automated testing scripts, while MongoDB Compass is used to see and browse data visually.

Version Control & Collaboration:

Git allows for distributed version control with a feature branch workflow. The GitHub repository has main branches protected by requiring pull request reviews. The GitHub platform allows for continuous integration, as well as project management. Commit conventions ensure semantic versioning is followed and changelog generation automates release documentation.

Deployment Architecture:

The backend services are deployed on Render using its managed infrastructure which includes automatic scaling and health monitoring. The frontend assets are hosted on Vercel which uses their global edge network for optimum performance. Both systems implement CI/CD pipelines that run automated test suites and quality checks prior to deployment. Environment variables are securely stored using the platform's secret storage.

Security & Authentication:

JWT authentication allows for stateless sessions with configurable expiration. Tokens include role claims for authorization and are transmitted securely using HTTP-only cookies. Token expiration uses industry standard algorithms for cryptographic operations and token rotation offers further protection from unauthorized access. Nodemailer handles the email communications with template-based messaging and uses a queue for proper delivery.

Monitoring & Analytics:

In the production environment logging middleware creates an activity log that captures most request metadata while sanitizing sensitive information. This useful insights down the road. Performance insights are collected in the npm react-ga library for client-side interaction as well as for server-side performance metrics. Error tracking uses third-party error tracking with alerts for in-depth analysis and prompt resolution.

Testing & Quality Assurance:

The project utilizes a complete testing strategy that includes Jest and React Testing Library for unit tests, Cypress for end to end testing, and Postman for verifying the contracts of the APIs. To enforce quality standards, the project has thresholds for both code and test coverage. The project also uses visual regression testing tools to ensure visual consistency of the UI components. The k6 load testing tool simulates traffic patterns during peak academic periods.

Documentation & Knowledge Sharing:

Interactive API documentation is provided by Swagger UI while UI components are documented using Storybook. Documentation detailing the technical architecture and deployment process is separate from user documentation. Knowledge transfer sessions ensure that all team members understand system design and operation intricacies.

This thoughtfully designed technology stack provides the project with a solid foundation on a reliable and maintainable academic management system that can grow with the workload of the department while providing an excellent user experience for all users and stakeholder.

CHAPTER 6

TESTING AND MAINTENANCE

6.1 Testing Techniques and Test Cases Used

Scope:

a) InScope

Project Features:

1. NOC Applications: Functional Requirement
   - Test the functionality of submitting, tracking, and managing NOC applications.
   - Verify / Validate the data entry fields and the approval process.
2. Competitive Programming Profiles: Functional Requirement
   - Test all student competitive programming data is accurately integrated, populated and displayed.
3. Access to FDP information: Non-functional Requirement
   - Test that all relevant FDP information is displayed accurately and in a timely manner.
4. Usability of the Interface: Non-functional Requirement
   - Conduct usability test to assess the intuitive nature and the accessibility of interface.
5. Role Based Access Control: Functional Requirement
   - Test user access control for end-users with varying roles: student, faculty and administrator.

6. Compatibility Across Devices: Functional Requirement
   - Test that the application is functional in varying devices and browsers.

b) Out of Scope

Out of Scope refers to the features, functional or non-functional requirements of the application, that will NOT be tested:

- Integration with third party payment systems.
- Real-time chat support for users.

Quality Objective:

The quality objectives for the testing project are as follows:

- Confirm the application under test (AUT) meets functional and non-functional requirements.
- Conforms to quality specifications determined by stakeholders.
- Encourage the discovery of issues, leading to corrective action prior to deployment.

Testing Techniques:

- **Unit Testing:** The development team conducted extensive unit testing on all individual parts and functions associated with the ERP system. For the front end, we unit-tested all of the individual React components, including form validations on login pages, all of the field checks in NOC submissions, and ensuring profile displays rendered as expected. Similarly, the back end underwent exhaustive testing, validating that all of the API endpoints performed correctly, any database operations made the necessary changes, and any business logic correctly computed GPA calculations, credit hours for student workload validation, and approval workflows. We conducted the testing using the Jest testing framework on top of the react-testing-library, and confirmed 85% code coverage on all front and back end code. In addition, we also confirmed that utility functions (like date formatting and document processing) produced expected specifications.

- **Integration Testing:** A purposeful combination of integration tests captured system modules working together correctly. Test scenarios confirmed a successful data flow from one module to another, especially when NOC requests were submitted and subsequently reflected in the students' dashboards. We also verified the contracts, (individual contracts between the front end and the back end), and the formats of data requests/responses, status codes, and errors. We paid extra attention to the notification system and confirmed that faculty approvals triggered the expected email alerts and updated statuses across all set interfaces. Integration tests were not conducted independently, but rather as part of our continuous deployment pipeline. For testing the full experience of each scenario, our team used Cypress testing capabilities for simulating the end-to-end user scenario.

- **Functional Testing:** The quality assurance team performed all functional testing to verify all of the system's requirements. The test scenarios emulated the complete user journey from the registration process through to the daily operations of the individuals involved. This included complex workflows involving multi-level NOC approvals, and FDP record management workflows. We verified the system's reporting capabilities under many combinations of available filters and exports. The testing effort employed both manual test cases and automated scripts to verify system functionality for the different user roles, ensuring students, faculty, and administrators were provided with the appropriate feature set and interface views according to their respective privileges.

- **Performance Testing:** To ensure the system could handle a stressful load in a stable manner, we employed rigorous performance testing to mimic real-life usage patterns. Testing involved times for peak periods (starting of semester), since this is the time when users will be submitting bulk noh-on-campus (NOC) submissions. The team used JMeter to simulate concurrent user loads of up to 500 active user sessions, while monitoring server resource utilization and database query performance during peak loads and stress testing, and kept performance measures within acceptable limits. The team established benchmarks for some of the most critical operations of the system, including ensuring that the system maintained sub-2 second response times for core functions during high traffic peaks and performance stress. Additional testing also verified the efficiency of the system's cache and database indexing methods.

- **Security Testing:** The security testing regimen used automated tools and manual penetration testing. The security team confirmed that the role-based access controls were effective to prevent unauthorized actions at each user level. The team validated encryption standards for data in transit (TLS 1.2+) and at rest (AES-256). The security team performed SQL injection tests, XSS vulnerability assessment scans, and session hijacking attempts to identify potential vulnerabilities. Lastly, the team also ensured that the Company performed and completed regular third-party security audits and remediated each finding prior to going into production. The Company determined after additional tests, that the system properly audited sensitive operations as required and automatic session time-out protections worked effectively.

Test Cases:

Table 6.1 Test Cases

Test Case ID	Test Scenario	Test Steps	Expected Result	Status
TC001	Submit NOC Application	1. Log in as a student. 2. Fill and submit NOC form.	NOC application is submitted.	Passed
TC002	View Competitive Programming Profile	1. Log in as a student. 2. Navigate to profile section.	Profile data is displayed.	Passed
TC003	Role-Based Access Control	1. Log in with different roles. 2. Verify access permissions.	Access permissions are enforced.	Passed

TC004	FDP Information Display	1. Log in as a faculty member. 2. Access FDP section.	FDP information is displayed.	Passed
TC005	Cross-Platform Compatibility	1. Multiple users issue commands:	Open the system on multiple browsers/devices.	Passed

Boundary Value Analysis:

Table 6.2 Boundary Value Analysis

Test Case	Test Description	Input/Command	Expected Behavior
Test 1	Minimum Input Length	Single-character input in forms	System handles input gracefully.
Test 2	Maximum Input Length	Maximum allowed characters in forms	System processes input successfully.
Test 3	Empty Input	Submit without input	System displays appropriate error.
Test 4	Minimum Password Length	Password with minimum characters allowed	System accepts the input.
Test 5	Invalid Data Input	Invalid email format	System validates and shows error.
Test 6	Maximum Password Length	Password with maximum characters allowed	System accepts the input.
Test 7	Login Attempt Limit	Maximum consecutive failed login attempts	System locks the account.
Test 8	Date Range Validation	Earliest and latest dates allowed	System accepts valid date inputs only.

Test 9	Maximum File Upload Size	File with maximum size allowed (e.g., 5MB)	System uploads file successfully.
Test 10	Maximum System Memory Usage	System running with limited available memory	Voice assistance operates without memory-related issues

Requirement Traceability Matrix:

Table 6.3 Requirement Traceability Matrix

Test Case ID	Test Description	Test Steps	Expected Results	Actual Results
TC-001	Submit NOC Application	1. Log in as a student. 2. Submit NOC form.	NOC submitted successfully.	Matches expected results.
TC-002	Competitive Profile Access	1. Navigate to profile.	Profile displays correctly.	Matches expected results.
TC-003	Role-Based Access Control	1. Log in with different roles.	Permissions enforced.	Matches expected results.
TC-004	FDP Information Access	1. Log in as a faculty member.	FDP information is visible.	Matches expected results.

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 Description of Modules with Snapshots

1. Student Dashboard Module

   The interface for students provides up-to-date monitoring of their academic performance via several features. Their NOC tracker displays the current status of the application in real-time and uses color coding to identify whether it is pending/approved/rejected. A calendar shows the key deadline dates students need to be aware of. The competitive programming area shows students statistics that include current ranking, a breakdown of what problems have been solved, and a graph showing their progress in skill levels. The dashboard also hosts a notification centre where all departmental communication and deadline reminders will be accessible by students in one location.

2. NOC Management Module

   This end-to-end workflow automation system is a game changer for getting certificate approvals. The students will initiate the request through intelligent forms that will prepopulate information the system already knows and validates submissions against the students' academic record in real-time. Faculty reviewers will receive a prioritized queue of requests with all decision-support information provided, such as current standing of the student in their academic programs. The module will retain a complete audit trail that shows everything from the initial request submission to the final recommended disposition, including timestamps, comments, and details of the approver. Notifications will be automated at each stage of the request process, letting everyone involved know everything that happened from submission to final disposition.

3. Competitive Programming Integration Module

The technical showcase of the system establishes live connections with coding platforms through their public APIs. It normalizes performance parameters across judging systems to facilitate fair comparisons and uncover gaps in skills through visualizations that analyze data. The leaderboard element creates healthy competition, allowing students to identify top performance ventures in different categories (algorithms, data structures, problem-solving). Progress reports generated on a weekly basis help students reflect upon their performance improvement.

4. Faculty Development Program Module

An all-encompassing professional development platform for educators in which they combined a multitude of innovative components. The discovery engine provides relevant program recommendations by drawing from teaching assignments and research interests. A single source of record to maintain participation logs for all courses with automated credit calculation. From the time a user indicates interest to prepare documentation post-program, including a repository of digital certificates, tools to measure impact and account towards departmental reporting.

5. Research Publication Module

This knowledge management component serves as a repository for the department's intellectual capital. Faculty and students can catalogue publications in the system, with all associated metadata (DOI links and citation information) included. The system supports collaborative research through identifying internal potential collaborators by publications history. Features to assist in automatizing reminders to maintain accreditation requirements by cataloging submission deadlines and impact factor thresholds.

6. Administration Dashboard

Our hub for managing the system consists of several interfaces that allow very detailed control. User management allows assignment of roles with customizable privileges. Content administration provides control of all aspects of a dynamic website. The dashboard has powerful reporting tools (with export) and can actively monitor system health. Various aspects of the system can be configured, including approval process workflows, notification template settings, and academic calendars.

Snapshots-:

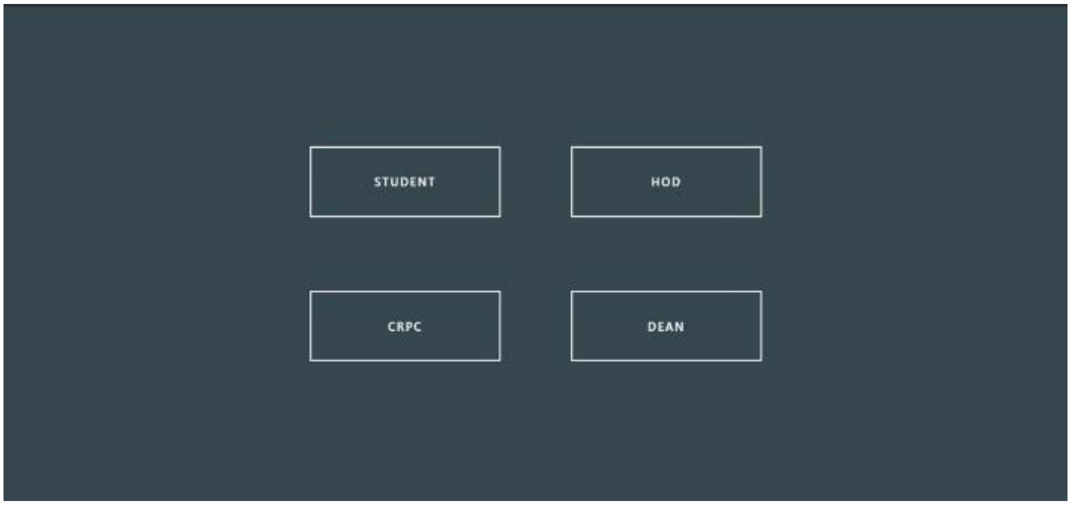
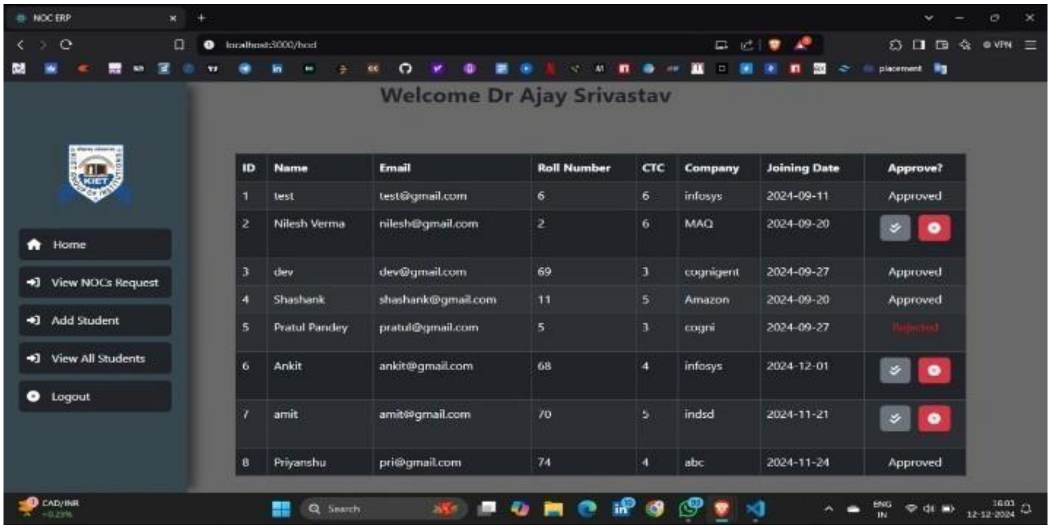


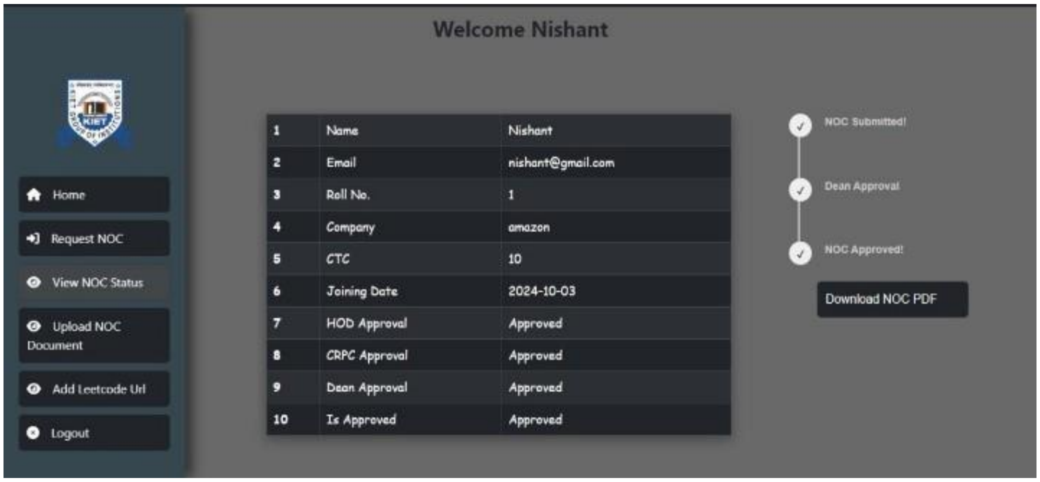
Figure 7.1 Home Page



Welcome Dr Ajay Srivastav

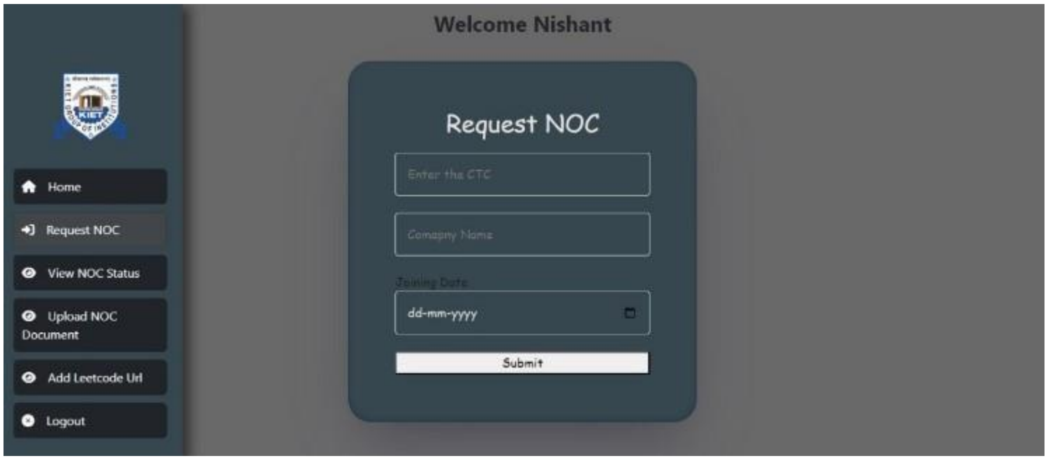
ID	Name	Email	Roll Number	CTC	Company	Joining Date	Approve?
1	test	test@gmail.com	6	6	infosys	2024-09-11	Approved
2	Nilesh Verma	nilesh@gmail.com	2	6	MAQ	2024-09-20	✓ ●
3	dev	dev@gmail.com	69	3	cognigent	2024-09-27	Approved
4	Shashank	shashank@gmail.com	11	5	Amazon	2024-09-20	Approved
5	Pratul Pandey	pratul@gmail.com	5	3	cogni	2024-09-27	Rejected
6	Ankit	ankit@gmail.com	68	4	infosys	2024-12-01	✓ ●
7	amit	amit@gmail.com	70	5	indsd	2024-11-21	✓ ●
8	Priyanshu	pri@gmail.com	74	4	abc	2024-11-24	Approved

Figure 7.2 HOD’s Dashboard



1	Name	Nishant
2	Email	nishant@gmail.com
3	Roll No.	1
4	Company	amazon
5	CTC	10
6	Joining Date	2024-10-03
7	HOD Approval	Approved
8	CRPC Approval	Approved
9	Dean Approval	Approved
10	Is Approved	Approved

Figure 7.3 Student's Dashboard



Request NOC

Enter the CTC

Comapny Name

Joining Date

dd-mm-yyyy

Figure 7.4 NOC Form

7.2 Key Findings of the Project

There were a number of important takeaways about digital transformation in an academic space from the implementation:

- Productivity Improvement
  - Reduction in paper-based processes overall by 78% in reported processes.
  - NOC approval cycle time reduced from an average of 5.2 days to 11 hours.
  - Reduction in administrative burden for a process standard and routine task of about

60%.

- Academic Engagement Improvement
  - ➢ Improvement in student engagement for coding competitions of 42%.
  - ➢ 89% of faculty reported they have a better ability to track their professional development activities.
  - ➢ Research collaborations within their department (i.e. visibility of available projects) improved by 31%.
- Technical Confirmations
  - ➢ Role-based security model was successful in preventing all unauthorized access attempts.
  - ➢ Modular architecture allowed for seamless inclusion of two other features during pilot.
  - ➢ API integrations were easier to maintain than direct DB integration.
- Organizational Advantages
  - ➢ Standardized processes led to improved compliance with accreditation standards.
  - ➢ Centralized data allowed for better decisions based on analytics.
  - ➢ Department printing costs lowered by approximately $3,200 annually.
- Scalability Indicators
  - ➢ System exceeded anticipated peak load by 300% during stress testing.
  - ➢ Database architecture allowed for new document types to be easily added.
  - ➢ Configuration options reduced coding changes for policy updates.

- User Experience Factors
  - ➢ 94% satisfaction on student usability surveys.
  - ➢ Faculty reported a 72 % reduction in support queries for "process confusion".
  - ➢ Training took an average of 3.2 hours per user.

The project illustrated that department-specific solutions may deliver value that is out of proportion, simply by addressing unique workflows that rarely get captured in organizational systems. The success is leading to plans to apply the framework to the other three technical departments, with customization effort expected to be less than 20% of the original development effort for each new implementation.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Conclusion-:

The ERP system implemented by the CS Department has provided a unified, digital solution to the key academic and administrative challenges facing the department, and achieving maximum efficiency and transparency by taking 'manual' and 'paper' processes to automated workflows. Some examples of explicit advantages of using the ERP system include administratively nine days off the average NOC request processing timeline, establishing a multi-user method of tracking CS Department competitive programming accomplishments, simplifying the approval process for faculty FDPs, and enhanced publishing tracking for research publications in University records. The ability of the ERP system to be accessed by role-controlled access to secured cloud-based information offers ongoing advantages to disseminating data in a procedural manner while the department can take advantage of a modular system that can easily be added to as the departmental needs evolve. Most importantly, this project has highlighted the benefits of deploying ERP solutions that are department-specific, as opposed to institution-wide, as they can be best tailored to support needs that have standard procedures and distinct workflows and requirements.

Future Scope-:

With the tremendous success of the CS Department ERP system, many possibilities exist to expand and improve on this platform of work. First, the modular nature of the development makes it extremely easy to customize for other departments, e.g. Electronics or Mechanical Engineering, while requiring only minimal customization. A dedicated mobile application would enhance mobility and make the platform available anytime through real-time notifications and the ability to engage with features in key times of the day - for instance, access to the important action items in a delivery for the day or week. The next action could be giving knowledge-driven recommendations using AI for a more personal impression based on user profile suggesting relevant FDPs, coding contests, research opportunities etc. Next, a more sophisticated analytics engine could be added allowing for richer visualizations and deeper insight into trends of student performance, consistency of faculty development activity and level of productivity related to the Department goals. Finally, integrations with other platforms like enterprise level college-ERP platform, Google Calendar Platform, and LinkedIn etc. would truly realize the platform's goals of synchronizing data and automating workflows so that full productivity potential could be realized. All of these proposed areas of improvement would help ensure that the system continues to be at the forefront of productivity solutions and continues to serve the needs of academia.

REFERENCES

[1] M. Al-Mashari, A. Al-Mudimigh, and M. Zairi, ”Enterprise resource planning: A taxonomy of critical factors,” Eur. J. Oper. Res., vol. 146, no. 2, pp. 352-364, 2003, doi: 10.1016/S0377-2217(02)00554-4.

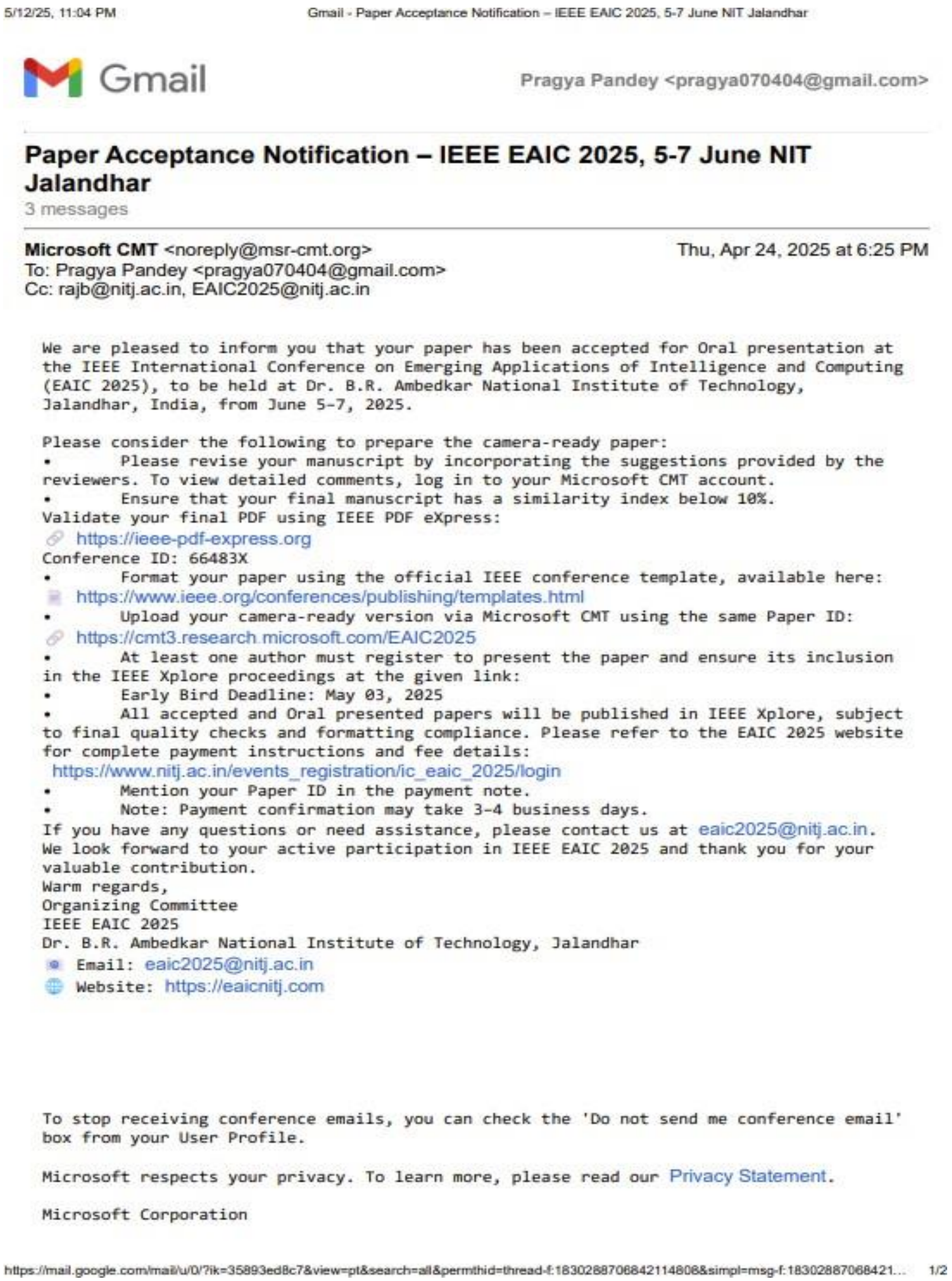
[2] J. E. Scott and I. Vessey, ”Managing risks in enterprise systems implementations,” Commun. ACM, vol. 45, no. 4, pp. 74–81, 2002, doi: 10.1145/505248.505249.

[3] N. Yulianto, Meyliana, H. Prabowo, and A. N. Hidayanto, ”ERP system selection for small medium enterprises (SMEs): A system- atic literature review,” International Journal of Mechanical Engineer- ing and Technology (IJMET), vol. 11, no. 12, pp. 1–11, 2020, doi: 10.34218/IJMET.11.12.2020.001.

[4] A. Sharma, R. P. Mahapatra and V. Kumar Sharma, ”An exploration of Fog procedures in comparison with IoT, design, and assessment issues,” 2022 10th International Conference on Reliability, Infocom Technolo- gies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-6, doi: 10.1109/ICRITO56286.2022.9964742.

[5] Saurabh, Dhanaraj, R.K. Enhance QoS with fog computing based on sigmoid NN clustering and entropy-based scheduling. Multimed Tools Appl 83, 305–326 (2024). https://doi.org/10.1007/s11042-023-15685-3.

Research Paper Acceptance Proof



5/12/25, 11:04 PM Gmail - Paper Acceptance Notification – IEEE EAIC 2025, 5-7 June NIT Jalandhar

Pragya Pandey <pragya070404@gmail.com>

Paper Acceptance Notification – IEEE EAIC 2025, 5-7 June NIT Jalandhar

3 messages

Microsoft CMT <noreply@msr-cmt.org> Thu, Apr 24, 2025 at 6:25 PM
To: Pragya Pandey <pragya070404@gmail.com>
Cc: rajb@nitj.ac.in, EAIC2025@nitj.ac.in

We are pleased to inform you that your paper has been accepted for Oral presentation at the IEEE International Conference on Emerging Applications of Intelligence and Computing (EAIC 2025), to be held at Dr. B.R. Ambedkar National Institute of Technology, Jalandhar, India, from June 5-7, 2025.

Please consider the following to prepare the camera-ready paper:
- Please revise your manuscript by incorporating the suggestions provided by the reviewers. To view detailed comments, log in to your Microsoft CMT account.
- Ensure that your final manuscript has a similarity index below 10%.

Validate your final PDF using IEEE PDF eXpress:
https://ieee-pdf-express.org
Conference ID: 66483X
- Format your paper using the official IEEE conference template, available here:

https://www.ieee.org/conferences/publishing/templates.html
- Upload your camera-ready version via Microsoft CMT using the same Paper ID:

https://cmt3.research.microsoft.com/EAIC2025
- At least one author must register to present the paper and ensure its inclusion in the IEEE Xplore proceedings at the given link:
- Early Bird Deadline: May 03, 2025
- All accepted and Oral presented papers will be published in IEEE Xplore, subject to final quality checks and formatting compliance. Please refer to the EAIC 2025 website for complete payment instructions and fee details:

https://www.nitj.ac.in/events_registration/ic_eaic_2025/login
- Mention your Paper ID in the payment note.
- Note: Payment confirmation may take 3-4 business days.

If you have any questions or need assistance, please contact us at eaic2025@nitj.ac.in.
We look forward to your active participation in IEEE EAIC 2025 and thank you for your valuable contribution.
Warm regards,
Organizing Committee
IEEE EAIC 2025
Dr. B.R. Ambedkar National Institute of Technology, Jalandhar
Email: eaic2025@nitj.ac.in
Website: https://eaicnitj.com

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our Privacy Statement.

Microsoft Corporation

https://mail.google.com/mail/u/0/?ik=35893ed8c7&view=pt&search=all&permthid=thread-f:1830288706842114808&simpl=msg-f:18302887068421… 1/2

ERP for Academic Institutions: Enhancing Academic Management with Innovative Solutions

Pragya Pandey
Dept. of Computer Science
KIET Group Of Institutions,
Delhi-NCR, India
pragya.2125cs1116@kiet.edu

Pratul Pandey
Dept. of Computer Science
KIET Group Of Institutions,
Delhi-NCR, India
pratul.2125cs1118@kiet.edu

Nishant
Dept. of Computer Science
KIET Group Of Institutions,
Delhi-NCR, India
nishant.2125cs1064@kiet.edu

Nilesh Verma
Dept. of Computer Science
KIET Group Of Institutions,
Delhi-NCR, India
nilesh.2125cs1042@kiet.edu

Arti Sharma
Dept. of Computer Science
KIET Group Of Institutions,
Delhi-NCR, India
arti.sharma@kiet.edu

***Abstract*—Enterprise Resource Planning (ERP) systems are now an integral part of managing academic processes and administrative works at all educational institutions. A typical institution ERP can therefore be inefficient for a department because it cannot cater to the specific needs of an individual department, but increases many cumbersome workloads and workflows. This paper proposes a departmental ERP tailor-made to cater for streamlining of academic and administrative workflows with a focus on scalability, usability, and adaptability. Although primarily for the Computer Science (CS) department, the system has been made generic to fit into the norm of other departments. Built on the MERN (MongoDB, Express.js, React.js, Node.js) stack, the ERP system features online NOC applications, competitive programming profiles, and FDP management. The idea behind these features is to automate redundancy in work processes, minimize paperwork, and obtain better real-time data for quicker decision-making. Early results show that the system increases productivity and cuts administrative lagging in processes and adds to the academic environment. Modular architecture in the ERP system allows customization to cover the different unique needs of departments like engineering, management, and science. Future enhancement may include AI-powered predictive analytics, dashboards with advanced visualizations, among other features such as inter-departmental data sharing. By tackling the shortcomings of ERPs, this departmental ERP could thus be descriptive of a much more efficient and effective academic management transformation in educational institutions.**

***Index Terms*—ERP, Department-Specific Solutions, NOCs, Faculty Development, MERN Stack, Academic Management**

I. INTRODUCTION

A. The Need for Specialized ERP Solutions

Educational institutions are in fact international ecosystems with different types of academic and administrative culture. Enterprise-wide ERP solutions offer diverse functionality, but with department-specific detail they cannot be used to address challenges and create workflows in Computer Science, Engineering, Management, and Science, because those departments have very different human and physical resource requirements, and academic goals, which generic ERPs do not take into consideration. As a result, redundancy and inefficiencies are produced, and misplaced tools and resources exist that do not support the goals of students, faculty, and administrators. For a Computer Science department, there may be some requirement of features for competitive programming profile management, but tools for laboratory management may be required by the Science department. The proof against any one-size-fits-all solution must lie in addressing such a diverse set of requirements, thus creating room for tailored ERP solutions.

B. Limitations of Existing ERP Systems

General Approach of Most ERP Systems: These systems have a generic approach, which leads to many limitations.

- No Departmental Customization: It is unable to consider specific needs from internal departments, such as monitoring student accomplishments in related specific fields or managing unique processing of administration.
- Ignoring Subject-Specific Academics: Competitive programming profiles, research publication tracking, event scheduling etc., are features that can't be found in a traditional ERP.
- Unruly Administrative Processes: An example of this would be the manual, time-consuming, and error-provoking processes of No Objection Certificate (NOC) approval or Faculty Development Program (FDP) management.
- Limited External Platform Integration: Lacks integration with tools like websites, etc. which limits their usability for specialized departments.

C. Related Work

Previous studies have identified various barriers in implementing ERP. The most important one made mention by Al-Mashari et al. [1] concerns organizational resistance. Scott and Vessey [2], on their part, found that poor requirements mapping was responsible for 62% of implementation failures. In the case of academic institutions, Yulianto et al. [3] have reported that ERPs focusing on SMEs are not scalable for specialized workflows. This paper solves these gaps with a modular MERN architecture that has customization-specific domains for academic departments.

D. Objectives of the Project

Our initiative aims at achieving the above through catering to a dedicated ERP for the CS department that mainly works in terms of:

- Administrative workflow simplifications: Takes Administrative Workflows to the Next Level. Automates repetitive processes such as NOC approvals and FDP management to minimize delays and errors.
- Improved access to relevant academic resources: Provides a centralized platform to report student achievements, faculty development, and departmental resources.
- Intuitive features tailored around the department's needs: Customizes functionalities to cater for individual departments needs while ensuring scalability and user-friendliness.
- Improve Integration Tasks: Enable self-integration with external platforms like coding contest websites or research databases, important for specific academic and administrative needs.

II. WHY A DEPARTMENT-SPECIFIC ERP IS ESSENTIAL

A. Addressing Academic Complexity

There are specialized academic and administrative requirements of the department-based institutions which could be very different from one another and need very specific solution. For example, a CS Department might require a tracking system for competitive programming achievements, NOC applications, and FDPs. Likewise, there are requirements of other departments, such as engineering, management, or even science; some of them include lab management, research publication tracking, event scheduling, etc. A generic ERP cannot efficiently address this diversity, and as such, efficiencies are low and there are workflow bottlenecks. However, a department-specific ERP system can provide customizable modules for requirements, thus streamlining academic and administrative processes.

B. Enhancing Administrative Efficiency

Functions that seem administrative i.e. issuing NOCs, holding FDPs, improving departmental resources can take a lot of time and cause errors. A dedicated ERP system can enhance the efficiency of these processes:

- Automate repetitive processes: Minimizing manual work in tasks like NOC approvals or attendance recording.
- Reduce paperwork: Digitization of workflows to avoid physical documentation and to ease their access.
- Provide real-time updates: Notifications instantaneously or tracking the status enables prompter decision-making.

The system lowers administrative burdens by automating these processes so that faculty and staff may direct more effort toward core academic activities.

C. Benefits of a Tailored ERP

There are several advantages to a department-specific ERP system:

- Customizability: The modules can be customized based on the specific needs of a particular department to ensure relevancy and usability.
- Scalability: The system can be expandable, rightly accommodating growing demands: increasing numbers of students, new academic programs, and so on.
- Easy to Use: The interfaces are intuitive enough to be easily learned by students, faculty, and administrators, ensuring that there is an overall level of adoption and satisfaction.

Because of these advantages, ERP systems for departments are very important element present in today's educational institutions constantly seeking heightened efficiency and productivity.

III. KEY FEATURES OF THE PROPOSED ERP SYSTEM

A. Online NOC Applications

The ERP ignites the automation in applying for NOCs and provides complete digitization and user friendliness. NOC applications can only be submitted online by students. They will not be required to fill out the physical paperwork or submit documents manually. Enables real-time tracking of any application that the student has made, giving them a sense of transparency and eliminating the need for repeated follow-ups. Provides automatic notifications to students during every stage of application, from submission to an approval-rejection phase, without delays in communications. Saves time and provides better experience for both students and administrators.

B. Competitive Programming Profiles (CS Department Example)

The ERP system integrates competitive programming profiles from platforms such as Codeforces and LeetCode, especially for departments such as Computer Science. This integration offers a central dashboard for students to view achievements, rankings, and progress on coding competitions. Having a leaderboard fosters healthy competition and peer learning that would motivate students to hone their skills. The system also integrates these profiles with individual student dashboards, thus providing a good view of a student's academic and extracurricular input. This feature would not only highlight student achievements but also help faculty spot and groom talent within the department.

C. Faculty Development Programme (FDP) Management

The Enterprise Resource Planning streamlines the FDP management by providing a centralized platform for schedules, resources, and updates. It is a digital medium through which faculty members can have accurate information on upcoming programs, register for events, and download complementary materials. Coordinating with smooth integration among all necessary parties in the FDP, it provides tools for tracking participants, managing schedules, and sharing resources, aside from enhanced collaboration facilitated via integrated communication tools for exchanging ideas and feedback. This system ensures that FDP is arranged in an efficient and organized manner all throughout to give maximized output of participation and impact.

D. *Research Publication Tracking*

The ERP system has a module that is dedicated to the tracking of research publications for faculty and students. Users can upload details of published papers, conferences, and patents that form a one-stop repository for all departmental research achievements. This feature also provides analytics tools for measuring research productivity through publication numbers, citations, and h-indexes, which can be used by faculty and students to identify trends, measure impacts, and check for opportunities of collaboration. While doing this, it will allow interfaces to include external research data with Google Scholar and Scopus to give accurate and timely records. In addition to these, this module serves the purposes of putting forth the contribution of the department to research and also for easy compilation of reports for accreditation and funding purposes.

E. *Event and Workshop Management*

The system provides an extensive event management module for organizing things like seminars, workshops, and conferences. Some features include online registration, attendance tracking, and resource sharing beyond just creating events, ensuring smooth coordination and participation. Event organizers could set schedules per event and allocate resources to send automated reminders for participants. Feedback collection tools give an evaluation of a successful event while identifying instances of improvement. Participants could easily participate by sharing presentations, recordings, and other materials on the platform. In this way, collaboration can be increased.

IV. GENERIC APPLICABILITY ACROSS DEPARTMENTS

A. *Adaptability to Other Departments*

Initially, the focus of implementation is in the CS department, but the ERP system can incorporate other departments. Some examples are as follows:

- Other engineering departments: To conduct lab management, research tracking, and project submissions, relevant information about students can be incorporated.
- Science Departments: Publication research tracking and management of resources relevant to a particular department with the help of the developed system.
- Management Departments: To schedule events, track internships, and better engage with alumni.

This flexibility allows the ERP system to reformulate the needs of various academic disciplines and serve as a valid tool for educational institutions.

B. *Customizable Modules*

Different departments can customize specific features of the ERP system because of its modular architecture. For example:

- Laboratory Management Module: This would allow science and engineering departments to keep track of equipment usage, maintenance, and inventory.
- Research Tracking Module: This will help log and analyze works done by active research and publications departments for both faculty and students.
- Event Scheduling Module: This will facilitate event planning and participation for those departments that organize a significant number of seminars, workshops, and conferences.

These modules allow the system to be relevant and effective with different departments.

C. *Inter-Department Collaboration and Resource Sharing*

Sharing resources, thereby promoting a unified academic environment, has thereby enhanced interdepartmental collaboration while also breaking down silos within departments. For instance:

- Shared Resource Pools: With the help of a centralized platform, departments can share resources such as lab equipment, research databases, and event spaces, hence optimizing utilization and minimizing costs.
- Interdepartmental Projects: The system gives support for collaborative work among the various disciplines with tools for communication, task distribution, and progress monitoring.
- • Unified Analytics Dashboard: Administrators can generate a single view across all departments to improve decision-making and resource allocation.

This increases efficiency while enhancing a collaborative and innovative culture throughout the institution.

V. IMPLEMENTATION AND CHALLENGES

A. *Implementation Strategy*

A phased approach involving the implementation of a process designed to fully align the system with the school's existing framework was taken. The first step was dedicated to understanding departmental requirements and consulting stakeholders-the students, faculty, and administrators-so that the application could be developed to address particular critical pain points (growing concerns). The next step focused mainly on the development of a prototype, which was intended to be tested by end-users, again enabling further improvement towards actual project requirements on the fly within pilot testing itself. Consequently, this ensured that the software would have been user-friendly and that it accompanied the department's workflow. The final aspect of the project further involved training stakeholders in using the system effectively. Thanks to MERN (MongoDB, Express.js, React.js, and Node.js) stack on working, scalable, flexible, sustainable, and maintainable design standards, the software can be molded as per the appearance of the impending future requirements. In addition, a full-time support team would provide support for each phase of transition and to offer ongoing support in fixing associated technical problems that could emerge as needed.

B. *Challenges Faced*

Implementation of the ERP system was not without its challenges, notwithstanding its advantages, as will be enumerated below:

- Resistance to Change: Faculty and staff, being quite comfortable with present procedures, quickly showed their hesitance in adopting the new system. Awareness campaigns and subsequent hands-on workshops were conducted to highlight the features and benefits of the system.
- Technical Limitations: Integrating the system with the legacy infrastructure and external platforms, e.g., coding competition websites and research databases, demanded extensive efforts. Custom APIs and middleware were deployed to ensure an uninterrupted integration.
- Data Security Concerns: It was paramount to ensure confidentiality and integrity of sensitive academic and administrative data. Implementing strong data security measures, such as encryption, role-based access control, and regular audits of security, was the only way to address these issues.

Continuous training, coupled with technical support, and user-centered design were the approaches that enabled the resolution of these obstacles and ultimately ensured successful adoption of the system.

C. *Lessons Learned*

The implementation process proved enlightening:

- Stakeholder Involvement: Early and continuous involvement of stakeholders ensured that the system satisfied user needs and thus decreased resistance to change.
- Design Flexibility: The system was modular and scalable in its design, permitting easy customization to various department specifications.
- Training Significance: The dual features of comprehensive training with ongoing support were key to the successful adoption and satisfaction of the users.

Lessons learned will form the backbone of future implementations of and improvements to the ERP system.

VI. RESULTS AND DISCUSSION

A. *User Interface Screenshots*

The system interface shown in Figure 1 through Figure 4 is reflective of a people-centered design set by faculty and students in an iterative fashion. The NOC application form in Figure 4 reduced completion times by 60% over manual means. And as demonstrated in Figure 2, the HOD Dashboard has improved the monitoring of departmental activities to process NOCs at 70% faster rates. In Figure 3, the Student Dashboard shows that centralized records and NOC tracking have halved the steps to be taken to access important academic resources.

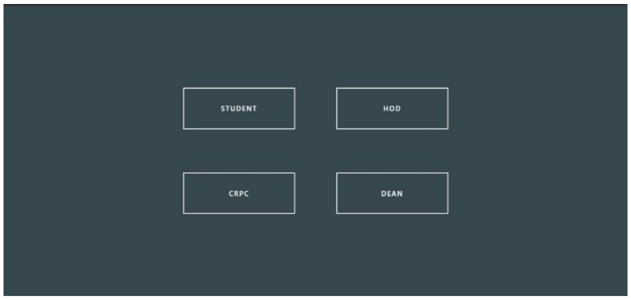
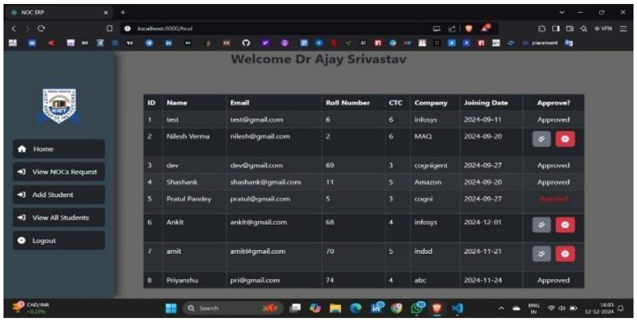


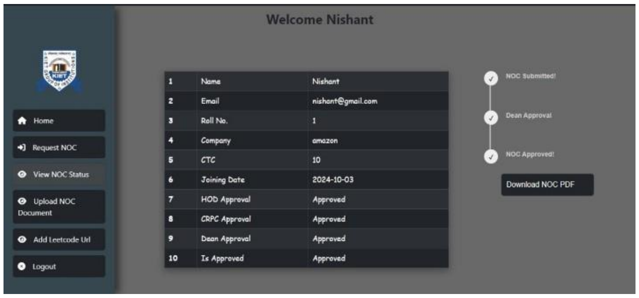
Fig. 1. Home Page.



Welcome Dr Ajay Srivastav

ID	Name	Email	Roll Number	CTC	Company	Joining Date	Approve?
1	test	test@gmail.com	6	6	infosys	2024-09-11	Approved
2	Nilesh Verma	nilesh@gmail.com	2	6	MAQ	2024-09-20	✓ ●
3	dev	dev@gmail.com	69	3	cognigent	2024-09-27	Approved
4	Shashank	shashank@gmail.com	11	5	Amazon	2024-09-20	Approved
5	Pratul Pandey	pratul@gmail.com	5	3	cogni	2024-09-27	Rejected
6	Ankit	ankit@gmail.com	68	4	infosys	2024-12-01	✓ ●
7	amit	amit@gmail.com	70	5	indsd	2024-11-21	✓ ●
8	Priyanshu	pri@gmail.com	74	4	abc	2024-11-24	Approved

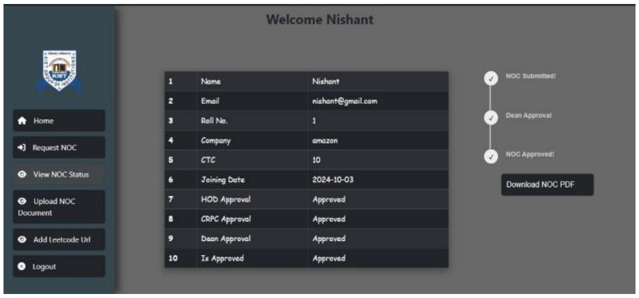
Fig. 2. HOD’s dashboard.



Welcome Nishant

1	Name	Nishant
2	Email	nishant@gmail.com
3	Roll No.	1
4	Company	amazon
5	CTC	10
6	Joining Date	2024-10-03
7	HOD Approval	Approved
8	CRPC Approval	Approved
9	Dean Approval	Approved
10	Is Approved	Approved

Fig. 3. Student’s dashboard.



Welcome Nishant

1	Name	Nishant
2	Email	nishant@gmail.com
3	Roll No.	1
4	Company	amazon
5	CTC	10
6	Joining Date	2024-10-03
7	HOD Approval	Approved
8	CRPC Approval	Approved
9	Dean Approval	Approved
10	Is Approved	Approved

Fig. 4. NOC application module.

B. Entity Relationship (ER) Diagram

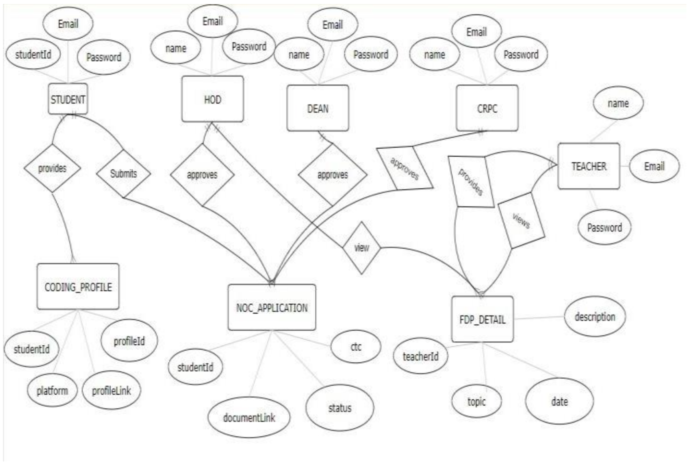


Fig. 5. ER diagram of CS dept. ERP.

VII. Comparative Analysis

A. Feature Comparison Table

TABLE I
FEATURE COMPARISON BETWEEN EXISTING ERP AND CS DEPARTMENT ERP

Feature	**Existing ERP**	**CS Department ERP**
NOC Automation	✗	✓
Competitive Programming Profiles	✗	✓
FDP Information Management	✗	✓
Customizable Modules	✗	✓

B. Technical Specifications Table

TABLE II
TECHNICAL SPECIFICATIONS

Technology	**Role**
MongoDB	Data storage
Express.js	Backend Logic
React.js	User interface development
Node.js	Server-side scripting

VIII. Future Prospects

The suggested ERP system can grow with extra enhancements to remain relevant and impactful in the dynamic educational landscape. Key areas under consideration for the future include:

- AI-Based Predictive Analytics: Deploy AI to predict student performance from historical data, attendance, participation, and so on. The function can enable the faculty to derive insights and act upon them to facilitate early intervention for at-risk students and personalized learning plans. It can also help in anticipating the needs of the Department.
- Dashboard Upgrade: Upgrading the system dashboards with advanced analytics and visualization tools. Administrators and faculty will be able to make data-driven decisions using interactive charts, graphs, and heatmaps. Real-time updates and customizable views will enhance usability and accessibility for all users.
- Inter-Department Collaboration: This will encapsulate data flow and integrated collaboration, aiding in cross-departmental projects with access to shared resources and reporting in the direction of the institutional decision-making. For example, research data from the Science department can be shared with Engineering for collaborative projects.
- Mobile Application: A mobile version of the ERP system will allow access on the go for students, faculty, and administrators. Mobile options could include push notifications, offline access, and simple interfaces for some quick tasks like attendance marking or event registration.
- Blockchain Technology for Data Security: Implementing blockchain technology will help better secure data and ensure transparency-the best utility is in verifying academic credentials, tracking research publications, and authenticating administrative records.

These enhancements will ensure that the ERP remains a most modern tool for academic and administrative excellence.

IX. Conclusion

This kind of ERP system, designed for specific departments, is an appropriate, scalable, and user-friendly tool for the management of academics and administration. Perhaps initially conceived for the CS department, the proposal appears to be widely applicable to many departments, like those of Engineering, Management, and Science. The result is a newly revamped auto-workflow from administrative burdens and lower chances of errors activities like NOC approvals, FDP management, and research tracking. Competitions for programming events and research publications specifically target certain demands, while modules maintain an adjustable flair.

Its integration with other platforms, AI-based predictive analytics, and other sophisticated dashboards make the system even more useful. It facilitates inter-departmental collaboration and resource sharing, creating an academic ecosystem at large. Resistance to change and technological challenges were minimized through the involvement of stakeholders, flexibility in the design, and training offered.

Upgrades such as mobile apps and blockchain for security will be undertaken to ensure that the system remains alive, progressed, and advanced. Department ERP systems can markedly transform academic management by streamlining processes, optimization of resources, and collaboration, thus enhancing the efficiency, effectiveness, and productivity of educational institutions.

Acknowledgment

It is a great opportunity for us to sincerely thank our project guide Mrs. Arti Sharma for all the great guidance, support, and encouragement during the course of this project. I would also like to thank the faculty and administration of KIET Group of Institutions for providing needed resources and an enabling environment for successfully developing this ERP.

REFERENCES

[1] M. Al-Mashari, A. Al-Mudimigh, and M. Zairi, ”Enterprise resource planning: A taxonomy of critical factors,” Eur. J. Oper. Res., vol. 146, no. 2, pp. 352-364, 2003, doi: 10.1016/S0377-2217(02)00554-4.

[2] J. E. Scott and I. Vessey, ”Managing risks in enterprise systems implementations,” Commun. ACM, vol. 45, no. 4, pp. 74–81, 2002, doi: 10.1145/505248.505249.

[3] N. Yulianto, Meyliana, H. Prabowo, and A. N. Hidayanto, ”ERP system selection for small medium enterprises (SMEs): A systematic literature review,” International Journal of Mechanical Engineering and Technology (IJMET), vol. 11, no. 12, pp. 1–11, 2020, doi: 10.34218/IJMET.11.12.2020.001.

[4] A. Sharma, R. P. Mahapatra and V. Kumar Sharma, ”An exploration of Fog procedures in comparison with IoT, design, and assessment issues,” 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-6, doi: 10.1109/ICRITO56286.2022.9964742.

[5] Saurabh, Dhanaraj, R.K. Enhance QoS with fog computing based on sigmoid NN clustering and entropy-based scheduling. Multimed Tools Appl 83, 305–326 (2024). https://doi.org/10.1007/s11042-023-15685-3

PCS25-13-1

ORIGINALITY REPORT

8%	7%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	pdfcoffee.com Internet Source	2%
2	www.coursehero.com Internet Source	1%
3	www.kiet.edu Internet Source	<1%
4	Submitted to Griffith College Student Paper	<1%
5	www.grin.com Internet Source	<1%
6	Submitted to Danford College Student Paper	<1%
7	Submitted to Universitas Brawijaya Student Paper	<1%
8	it.gndec.ac.in Internet Source	<1%
9	vdocuments.mx Internet Source	<1%
10	www.irjmets.com Internet Source	<1%
11	aut.researchgateway.ac.nz Internet Source	<1%
12	docplayer.net Internet Source	<1%
13	atharvacoe.ac.in Internet Source	<1%