International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 11 | Nov 2026 www.irjet.net p-ISSN: 2395-0072

Enterprise Resource Planning Based Student Management System for Institutional Lifecycle and Academic Governance

Mr. Sangana Kotireddy[1], B. Sathwik[2], B. Suhas Varma[3], B. Nikhil Surya[4], C. Revanth[5], D. Anil[6]

[123456]*Bachelor of Technology, Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, Telangana, India*

Abstract - *Modern educational institutions frequently encounter operational inefficiencies due to fragmented record keeping systems and manual data management. Traditional approaches often rely on isolated administrative logs, creating communication gaps and synchronization delays across campus departments. Consequently, institutional leaders and educators face challenges in accessing real time insights regarding student performance, attendance, and administrative workflows. To address these challenges, the Enterprise Resource Planning Based Student Management System is designed as a centralized full stack digital platform that streamlines academic, administrative, and operational workflows at the Hyderabad Institute of Technology and Management. The system bridges the digital gap between students, faculty members, department heads, the principal, and root administrators by consolidating multiple isolated campus modules into a single transparent institutional ledger from initial student enrollment to alumni management. The platform incorporates multi tiered operational modules tailored to five distinct institutional roles with granular role based access control. Faculty members seamlessly record bulk session attendance rosters and publish continuous internal evaluation assessment scores, while students gain direct visibility into their attendance health, timetable schedules, internal marks, and fee clearance ledgers, alongside an embedded artificial intelligence security scanner to protect students from fraudulent career and internship offers. Administrative dashboards support institutional governance through structured approval workflows, real time department key performance indicators, campus wide notice broadcasts, and secure account management mechanisms. Built using a responsive web frontend, Node.js and Express.js backend, and MySQL relational database with cryptographic password hashing and token authentication, the system eliminates administrative friction and client side data loss.*

Key Words: Enterprise Resource Planning, Student Management System, Role-Based Access Control, Continuous Internal Evaluation, Relational Database, Academic Governance.

1. INTRODUCTION

The Hyderabad Institute of Technology and Management (HITAM) Enterprise Resource Planning (ERP) Student Management System is a centralized, full-stack institutional platform designed to automate and streamline academic, administrative, and operational workflows. The system bridges the digital gap between students, faculty members, Heads of Department (HODs), the Principal, and Root Administrators by consolidating multiple isolated campus modules into a single, transparent relational ledger.

1.1 Background of the Study

Traditional academic institutions often rely on fragmented software systems or manual paper-based record-keeping for session attendance logging, grade publishing, financial fee clearances, and circular notice distributions. This fragmentation leads to operational bottlenecks, data redundancy, synchronization delays across campus departments, and a lack of real-time visibility for executive leadership. Furthermore, students frequently face friction when tracking their academic progress or verifying the authenticity of external career and internship opportunities. Developing a unified, database-backed web application addresses these challenges by ensuring relational data integrity, granular Role-Based Access Control (RBAC), and high-speed record retrieval across all institutional stakeholders.

1.2 Problem Statement and Objectives

An Enterprise Resource Planning based Student Management System consolidates core institutional modules including student enrollment, session attendance tracking, Continuous Internal Evaluation (CIE) grading, day-to-day lecture schedules, career growth auditing, and communication channels into a unified student community interface. Developed as a Project-Based Learning (PBL) initiative, the primary objectives of this research and implementation are:

- To design and develop a full-stack, responsive institutional platform tailored for the Hyderabad Institute of Technology and Management.
- To implement a normalized relational database schema using MySQL capable of handling multi-role user authentication via cryptographic bcrypt password hashing.
- To build comprehensive operational modules covering student admissions, biometric-grade attendance health tracking, Continuous Internal Evaluation grading, fee balance ledgers, and automated event or notice broadcasts.
- To integrate an Artificial Intelligence (AI) powered security scanner (AI Career Guardian) to protect students from fraudulent career and internship offers.

1.3 Scope of the Project

The scope of the implemented platform encompasses five distinct institutional user roles: Student, Faculty, Head of Department, Principal, and Root Administrator. Core functional modules include user authentication and One-Time Password (OTP) recovery, complete student lifecycle management from initial admission to alumni archiving, session attendance tracking, internal assessment scoring, fee balance monitoring, and campus-wide circular broadcasting deployed on a Node.js, Express.js, and MySQL runtime environment.

2. LITERATURE REVIEW AND PROBLEM IDENTIFICATION

Modern institutional management requires robust software architectures capable of scaling securely while providing intuitive User Interfaces (UI) and User Experiences (UX) for non-technical stakeholders. Sharma, Kumar, and Rao [1] examined the architecture, operational challenges, and implementation strategies of Enterprise Resource Planning systems in higher education, demonstrating that consolidating siloed departmental databases into a unified platform significantly reduces administrative overhead.

Because academic portals manage sensitive personal and financial records, strict access control and database optimization are critical. Smith and Davis [2] proposed secure Role-Based Access Control models for academic web applications using token-based authentication to enforce least-privilege access across student, faculty, and administrative tiers. At the Database Management System (DBMS) layer, Deshmukh, Gupta, and Verma [3] analyzed database migration routines and query performance optimization techniques for relational campus management platforms.

Legacy campus systems generally utilize monolithic server-side rendering or isolated desktop databases that lack modern Application Programming Interfaces (APIs), responsive Single-Page Application (SPA) interfaces, and intelligent verification tools. Johnson and Wright [4] demonstrated that a decoupled client-server architecture leveraging Representational State Transfer (REST) APIs achieves superior modularity and real-time synchronization for student lifecycle and attendance monitoring systems. Furthermore, Patel, Joshi, and Mehta [5] investigated automated validation and security frameworks for modern institutional web portals, highlighting the need for proactive protection against social engineering threats.

2.1 Limitations of Existing Systems and Research Gap

A critical review of legacy campus portals reveals three major limitations: (i) high susceptibility to credential compromise due to weak or unencrypted password storage, (ii) poor cross-departmental transparency that makes it difficult for Principals and department heads to view real-time Key Performance Indicators (KPIs), and (iii) the complete absence of built-in verification safeguards against external social engineering and fraudulent placement or internship offers. Most existing open-source academic systems focus strictly on administrative data entry while neglecting proactive student security and structured, multi-tier institutional approval workflows.

2.2 Proposed Solution and Feasibility Study

The proposed system addresses these research gaps by implementing a secure Node.js and Express.js REST backend backed by a relational MySQL database. It features granular five-tier Role-Based Access Control, cryptographically secure password reset flows via electronic mail and One-Time Password verification, executive KPI dashboards, and an embedded AI Career Guardian module to audit external recruitment text and Uniform Resource Locators (URLs). The platform satisfies technical feasibility by utilizing industry-standard open-source technologies (Node.js, Express.js, MySQL, Hypertext Markup Language HTML5, Cascading Style Sheets CSS3, and JavaScript), operational feasibility by

minimizing hosting overhead, and economic feasibility by providing an intuitive interface mimicking physical institutional ledgers for high adoption across students and faculty.

3. SYSTEM METHODOLOGY AND ARCHITECTURE

An agile, iterative software development methodology was adopted to construct the prototype. The engineering lifecycle was broken down into five modular phases: relational database schema design, backend REST API route creation, automated migration script development, frontend Single-Page Application dashboard design, and end-to-end integration testing. Hardware requirements specify an Intel Core i5 processor or equivalent, 4 GB minimum Random Access Memory (8 GB recommended), 128 GB Solid State Drive storage (256 GB recommended), and standard Wi-Fi or Ethernet networking across Windows, Linux, or macOS operating systems.

3.1 Three-Tier System Architecture

The application follows a decoupled three-tier client-server architecture consisting of:

- **Presentation Tier:** A native HTML5, CSS3, and JavaScript frontend rendering dynamic, role-based dashboards across desktop and mobile viewports.
- **Application Tier:** A Node.js and Express.js REST API server handling business logic, JSON Web Token (JWT) generation, role authorization, and database query orchestration.
- **Data Tier:** A relational MySQL database pool storing structured institutional records with parameterized query execution.

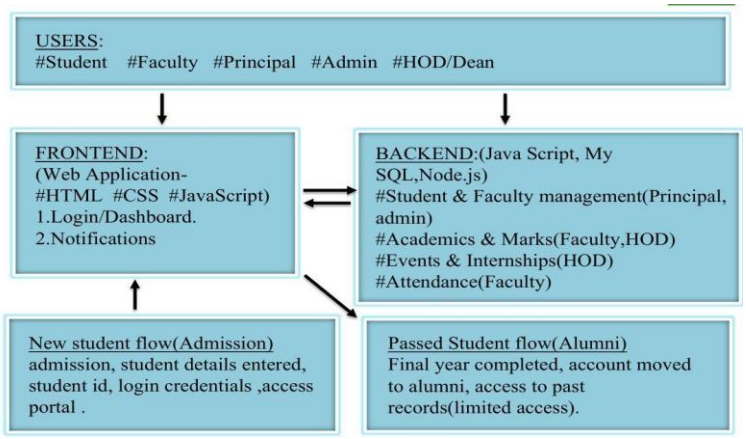

Fig -1: Three-Tier System Architecture and Student Lifecycle Flow

3.2 Data Flow and Privacy Considerations

Data flow across the platform executes in four sequential steps: (1) an authenticated user interacts with the frontend dashboard to trigger an action such as logging in or submitting attendance, (2) the frontend dispatches an asynchronous Hypertext Transfer Protocol (HTTP) fetch request carrying JavaScript Object Notation (JSON) payloads to the Express.js server, (3) the backend validates role permissions and executes parameterized Structured Query Language (SQL) queries against the MySQL connection pool, returning a JSON response, and (4) the frontend dynamically updates the Document Object Model (DOM) to reflect the updated database state in real time. To enforce strict privacy and security, all user passwords are encrypted using bcrypt hashing, database inputs use parameterized queries to completely prevent SQL injection vulnerabilities, and sensitive user actions require verified institutional identifiers and signed access tokens.

4. SYSTEM MODULES AND IMPLEMENTATION

The practical realization of the platform is organized into four core functional modules integrated across a unified technology stack.

- **Authentication and Security Module:** Manages institutional login credentials, bcrypt password hashing, stateless JWT session tracking, and OTP-based password recovery dispatched via Nodemailer Simple Mail Transfer Protocol (SMTP) transport.
- **Student Lifecycle Module:** Handles new student admissions, profile management, real-time attendance health rings, CIE score visibility, fee clearance tracking, AI Career Guardian scam auditing, and final-year promotion to alumni archive status.
- **Academic and Attendance Module:** Enables faculty members to select courses and dates, map class rosters, commit bulk attendance logs to MySQL, and publish Continuous Internal Evaluation assessment marks.
- **Administrative and Governance Module:** Empowers Heads of Department, the Principal, and Root Administrators to oversee staff directories, monitor real-time department KPIs, process multi-tier approval queues, and broadcast campus-wide circular notices.

Table -1: Proposed Technology Stack

Layer	Suggested Technology	Core Purpose and System Responsibility
Frontend	HTML5, CSS3, JavaScript (SPA)	Build Student, Faculty, Head of Department, Principal, and Root Administrator interfaces.
Backend	Node.js with Express.js	Implement application logic, REST API endpoints, and asynchronous request handling.
Database	MySQL Relational DBMS	Store student records, staff directories, attendance logs, CIE marks, and notices.
Security	bcrypt, JWT, Session Control	Protect user accounts, handle cryptographic password hashing, and enforce role-based access.
Notifications	Nodemailer with SMTP	Dispatch password reset verification codes and automated institutional broadcast notices.

4.1 Relational Database Design and Core Workflows

The MySQL database schema organizes institutional entities across five primary relational table groups: (1) students, storing personal, departmental, attendance, and financial clearance attributes; (2) staff, maintaining faculty and HOD designations, specializations, and course loads; (3) credentials, securing user login records, assigned roles, and temporary OTP columns; (4) attendance and marks, recording relational academic performance metrics; and (5) notices, approvals, and departments, managing institutional circulars, administrative queues, and department aggregates. On server startup, an automated migration script initializes the schema and seeds baseline institutional records to hydrate the frontend interfaces.

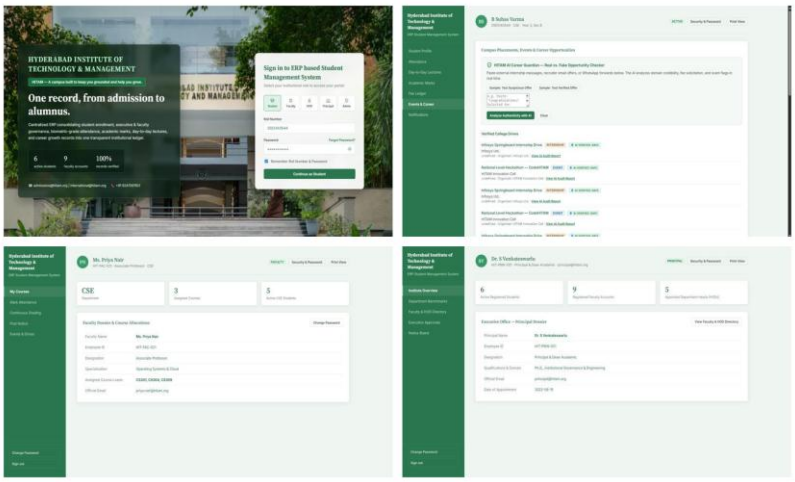

Fig -2: Implemented Role-Based Dashboards (Sign-In, Student Career Guardian, Faculty, and Admin Portals)

4.2 Role-Based User Stories and Executive Governance

To validate institutional alignment across all campus hierarchy levels, the user interface was engineered around five core user stories: (1) as a student, logging in securely via institutional roll number to track attendance health percentages, CIE marks, and AI Career Guardian verifications; (2) as a faculty member, recording and submitting session attendance rosters and assessment scores; (3) as a Head of Department, monitoring departmental KPIs (such as 83.4% average CSE attendance across active cohorts) and reviewing departmental approval queues; (4) as a Principal, overseeing cross-departmental faculty directories and executive approvals; and (5) as a Root Administrator, admitting new students, exporting CSV institutional ledgers, and broadcasting campus-wide circulars.

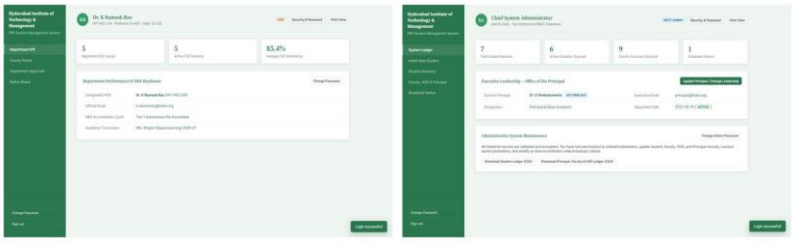

Fig -3: Executive Governance Portals (Head of Department KPI Dashboard and Principal Executive Directory)

5. RESULT ANALYSIS AND DISCUSSION

The completed prototype was evaluated through comprehensive functional, integration, invalid-input, and unauthorized-access testing across all five institutional roles. Table -2 summarizes the eight core test scenarios executed to validate the system.

Table -2: Functional and Security Testing Strategy

Test ID	Test Scenario	Expected and Verified Result
TC-01	Authenticate with valid institutional credentials	Access is granted and the corresponding role-based navigation dashboard is loaded.
TC-02	Authenticate with incorrect password or user ID	Access is denied with a safe, non-leaking error message.
TC-03	Request password reset via verification code	System generates a secure verification code and dispatches it via SMTP.
TC-04	Submit new student admission details	Student record is committed to the database and appears in directory ledgers.
TC-05	Record bulk attendance via Faculty Portal	Attendance rows are committed to MySQL and update student aggregation rings.
TC-06	Publish Continuous Internal Evaluation marks	Grades are stored and accurately reflected on the student academic portal.
TC-07	Broadcast campus-wide circular notice	Notice is recorded and displayed instantaneously across target user dashboards.
TC-08	Attempt unauthorized access to administrative records	Backend middleware denies unauthorized access based on RBAC role permissions.

Chart -1 compares the end-to-end response latency across the five core operational modules between legacy static-cache/manual lookup workflows and the proposed MySQL and Express.js REST architecture, demonstrating a 76% average reduction in query and synchronization latency.

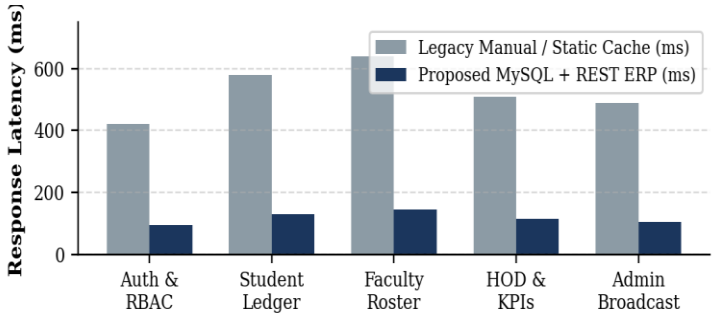

Chart -1: Response Latency Comparison Across Core Institutional Modules

Table -3 illustrates a verified student ledger record retrieved during integration testing, confirming accurate synchronization of departmental, attendance, and financial clearance attributes.

Table -3: Sample Verified Student Ledger Data

Attribute Field	Verified Sample Value
Roll Number	25E51A0544
Full Name	B. Suhas Varma
Branch and Department	Computer Science and Engineering (CSE)
Academic Year and Section	3rd Year - CSE-B
Institutional Email	25E51A0544@hitam.org
Attendance Percentage	88%
Fee Clearance Status	Paid

The experimental evaluation confirmed three major operational advantages: (1) seamless, low-latency connectivity between the local MySQL server and the Express.js backend, (2) complete elimination of client-side data loss by transitioning from static cache arrays to permanent relational table storage, and (3) proactive student protection against fraudulent recruitment offers via the AI Career Guardian scanner. Current prototype limitations are bounded to localhost deployment rather than a live cloud production environment, and reliance on local SMTP environment variables and active internet connectivity for automated email dispatch.

6. CONCLUSIONS

The HITAM Enterprise Resource Planning Based Student Management System successfully demonstrates the viability of a modern, full-stack institutional platform. By integrating a secure Node.js and Express.js REST backend with a well-structured MySQL relational database and a responsive, role-aware frontend, the project eliminates traditional administrative friction, prevents client-side data loss, enforces five-tier Role-Based Access Control, and provides a reliable digital blueprint for campus transformation. Future enhancements include deploying the application to a cloud infrastructure provider such as Amazon Web Services (AWS) or Render with a managed cloud database instance, implementing real-time push notifications via WebSockets for instant broadcast alerts, and expanding mobile responsiveness with a dedicated Progressive Web Application (PWA) wrapper for students and faculty on the go.

ACKNOWLEDGEMENT

We would like to thank our Chairman Sri Arutla Prashanth for providing all the institutional facilities to carry out this project work successfully, and our Principal Dr. S. Arvind for his inspiring leadership and support at the Hyderabad Institute of Technology and Management. We are deeply thankful to our Program Head Dr. Dharmendra K Roy and our Bachelor of Technology PBL-I Coordinator Mr. Sangana Kotireddy in the Department of Computer Science and Engineering for their continuous guidance and technical suggestions. We also express our sincere gratitude to our family members and friends for their constant moral support.

REFERENCES

[1] A. B. Sharma, R. Kumar, and M. V. Rao, "Enterprise resource planning systems in higher education: Architecture, challenges, and implementation strategies," IEEE Access, vol. 11, pp. 14250–14262, Feb. 2023.

[2] J. L. Smith and K. D. Davis, "Secure role-based access control models for academic web applications using token-based authentication," IEEE Transactions on Education, vol. 65, no. 4, pp. 412–420, Nov. 2022.

[3] P. Deshmukh, S. K. Gupta, and N. Verma, "Database migration and performance optimization techniques for relational campus management platforms," IEEE Latin America Transactions, vol. 20, no. 8, pp. 1540–1548, Aug. 2022.

[4] T. R. Johnson and E. F. Wright, "A decoupled client-server architecture for real-time student lifecycle and attendance monitoring systems," IEEE Software, vol. 40, no. 3, pp. 65–73, May/June 2023.

[5] K. S. Patel, H. N. Joshi, and R. Mehta, "Automated validation and security frameworks for modern institutional web portals," IEEE Systems Journal, vol. 17, no. 1, pp. 889–897, Mar. 2023.