ERP-Based Integrated Student Management System

A Web-Based Centralized Platform for Academic and Administrative Automation

Rutuja Gandhale
TYBCA — 22356030081

Awantika Gajare
TYBCA — 22356030080

Siddhi Gaikwad
TYBCA — 22356030079

Guide: Dr. Namrata Gadgil | Dept. of BCA, School of Computational Sciences, JSPM University Pune — A.Y. 2025-26

ABSTRACT

The ERP-Based Integrated Student Management System is a comprehensive web-based application designed to centralize and automate academic and administrative activities within educational institutions. Traditional systems handle attendance tracking, fee management, result processing, and student-teacher communication through isolated, manual processes—leading to inefficiencies, data inconsistencies, and delays. This paper presents a unified ERP platform integrating all these functionalities into a single system. Modules include secure role-based login, student and teacher dashboards, attendance monitoring, fee management, result analysis, document storage, internship tracking, and a real-time chat facility. The system is built using HTML, CSS, and JavaScript for the frontend; PHP for backend logic; and MySQL for database management, hosted via XAMPP. All ten test cases passed, demonstrating correctness, security, and reliability. The system reduces manual workload, improves data accuracy, and supports the digital transformation of higher education institutions.

Keywords: *ERP, Student Management System, PHP, MySQL, Web Application, JSPM University*

1. INTRODUCTION

Educational institutions generate and manage enormous volumes of data related to students, faculty, fees, examination results, and administrative records. In many colleges and universities, these activities are handled manually or through disconnected software tools, creating silos of information that impede efficiency and accuracy.

This paper presents the ERP-Based Integrated Student Management System—a centralized, web-based platform designed and implemented at JSPM University Pune for the Bachelor of Computer Application programme. The system integrates seven functional modules under a single authenticated interface, enabling students, teachers, the General Fee Manager (GFM), parents, and administrators to interact with accurate, real-time data.

The need for such a system is underscored by four primary pain points identified during requirement analysis: (i) excessive manual data-entry and paperwork; (ii) fragmented records across departments; (iii) limited self-service access for students; and (iv) growing institutional demand for digital transformation in the education sector.

2. LITERATURE SURVEY

A survey of existing student management and ERP solutions reveals that most commercially available systems address only a subset of the required functionality. Standalone attendance tools do not integrate with fee or result systems, forcing administrators to maintain parallel records. Open-source LMS platforms such as Moodle focus on content delivery rather than administrative automation.

Prior academic projects at similar institutions highlight gaps in communication features and multi-role access. The proposed system addresses these limitations by providing a unified dashboard for five distinct user roles, a built-in inter-user chat facility, and an internship tracking module—none of which are commonly found in open-source alternatives at the same integration level.

3. SYSTEM ARCHITECTURE

3.1 Three-Layer Model

The system follows a classic three-layer client-server architecture. The Presentation Layer—built with HTML, CSS, and JavaScript—renders dynamic dashboards and forms in any standards-compliant browser. The Application Layer, implemented in PHP, contains all business logic: authentication, session management, data validation, and module-specific processing. The Database Layer uses a normalized MySQL schema with seven tables linked by student_id as a foreign key.

This separation of concerns enhances security (the database is never exposed to the client), simplifies maintenance (each layer evolves independently), and improves scalability (additional modules plug into the Application Layer without restructuring existing code).

3.2 User Roles

Five user roles are supported: Student, Teacher, GFM (General Fee Manager), Administrator, and Parent. Role-based session variables restrict each user to their authorized module set. Administrators enjoy the broadest privileges—announcements, notices, and full student record management—while parents have read-only access to their ward's attendance, grades, and fee status.

4. FUNCTIONAL MODULES

Table 1 summarizes the seven core modules integrated into the system.

Module	Function
Login / Auth	Secure role-based access
Dashboard	Summary of all modules
Attendance	Track & calculate %
Fee Mgmt	Paid / pending details
Results	Marks & grade display
Documents	Upload & store files
Chatbot	Query support for users

Table 1: System Modules and Their Functions

The Login module validates credentials against the admin or students table in MySQL. On success, PHP creates session variables encoding the user's ID and role, after which the user is redirected to their role-specific dashboard.

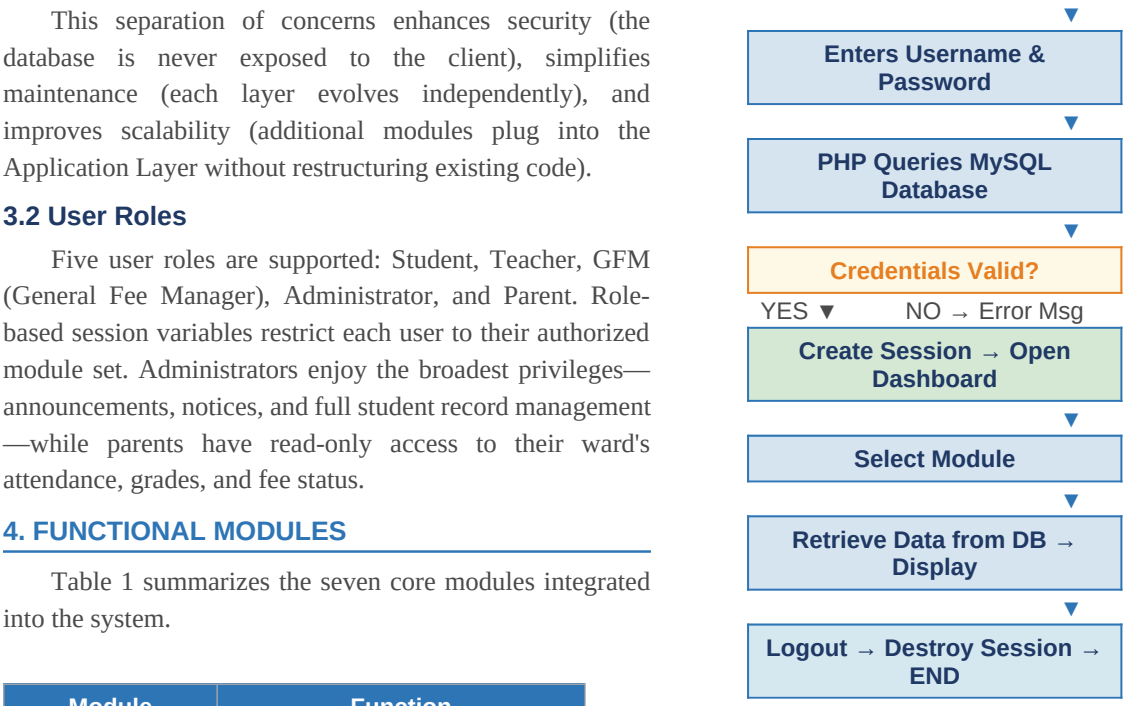
The Attendance module stores per-subject lecture counts and computes attendance percentage in real time. Students and parents can view the calculated percentage; teachers and administrators can create or update records.

The Fee Management module maintains total, paid, and pending amounts together with due dates. The GFM role was introduced specifically to give finance staff view and update rights without granting full administrative access.

5. SYSTEM DESIGN

5.1 Login Flow Diagram

Figure 1: Login Authentication Flow



5.2 Technology Stack

Layer	Technology
Frontend	HTML, CSS, JavaScript
Backend	PHP
Database	MySQL
Server	XAMPP (Apache)

Table 2: Technology Stack

6. TESTING & RESULTS

The system underwent six testing phases: unit, integration, system, validation, security, and UI testing. Ten formal test cases (TC-01 through TC-10) were executed; all passed without failure. Key results are summarized below:

TC	Module	Input	Result
TC-01	Login	Valid	✔ Pass
TC-02	Login	Invalid	✔ Pass
TC-03	Attendance	Open	✔ Pass
TC-04	Fee Module	Open	✔ Pass
TC-05	Result Module	Open	✔ Pass
TC-06	Doc Upload	PDF	✔ Pass
TC-10	Security	No auth	✔ Pass

Table 3: Selected Test Cases — All Passed

Security testing confirmed that unauthenticated URL access is redirected to the login page, and that session variables are destroyed on logout. No SQL injection

vulnerabilities were detected in form inputs during validation testing.

7. DISCUSSION

The implemented system demonstrably reduces administrative overhead. Tasks that previously required department visits—checking attendance percentage, fee balances, or examination results—are now available to students on demand through a browser. Administrators can update records and publish announcements in real time, improving institutional transparency.

The inclusion of a GFM role separates financial management from general administration, reducing privilege creep—a security best practice often overlooked in small-scale academic ERP implementations. The parent role similarly grants read-only visibility without burdening the student's privacy settings.

Limitations of the current implementation include dependency on local network connectivity (XAMPP server), absence of an online payment gateway, and lack of mobile-native support. These constraints are acknowledged and form the basis of the future scope.

8. FUTURE SCOPE

Several enhancements are planned for subsequent releases: (1) migration to a cloud-hosted server for remote access; (2) development of native Android and iOS applications; (3) integration of an online fee payment gateway; (4) biometric or face-recognition attendance; (5) AI-powered chatbot replacing the keyword-based prototype; (6) timetable and lecture scheduling module; (7) integration with cloud storage for document management; and (8) data-analytics dashboards providing predictive insights into student performance.

9. CONCLUSION

This paper presented the design, implementation, and evaluation of the ERP-Based Integrated Student Management System—a centralized, web-based platform that consolidates seven academic and administrative modules under a single authenticated interface at JSPM University Pune. The system successfully addresses the core problem of fragmented, manual record-keeping by providing real-time, role-segregated access to attendance, fee, result, document, and communication data.

All ten formal test cases passed, validating the correctness, security, and usability of the system. The three-layer client-server architecture ensures maintainability and scalability for future enhancements. The project demonstrates that open-source web technologies—PHP, MySQL, HTML/CSS/JavaScript—are sufficient to build a production-quality ERP system for educational institutions, substantially reducing development cost while meeting functional requirements.

REFERENCES

[1] Gandhale R., Gajare A., Gaikwad S. (2026). ERP-Based Integrated Student Management System. Project Report, JSPM University Pune.

[2] Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.

[3] Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems (7th ed.). Pearson.

[4] W3Schools Online Web Tutorials. https://www.w3schools.com

[5] PHP Documentation. https://www.php.net/docs.php

[6] MySQL 8.0 Reference Manual. Oracle Corporation. https://dev.mysql.com/doc/

[7] MDN Web Docs — JavaScript Reference. Mozilla Foundation. https://developer.mozilla.org