International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

CampusFlow: EdTech Training & Admission Management Portal

Ishika S. Gangan, Vaibhav K. Gawande, Atharva A. Dethe, Vansh S. Zade, Hrushikesh P. Dhore, Prof. Samrudhi S. Mamarde
Department of Computer Science and Engineering
P. R. Pote Patil College of Engineering & Management
Amravati, Maharashtra, India

***Abstract*—Modern educational hubs and skill-development academies deal with intricate administrative tasks spanning enrollment drives, batch mapping, schedule distribution, attendance tracking, grading workflows, fee tracking, and multi-tier communication. Operating these channels through piecemeal software or paper-heavy processes often results in records fragmentation, operational fatigue, and delayed updates. To resolve these fric-tion points, we introduce CampusFlow—an integrated digital ecosystem designed to centralize core academic and operational workflows. Developed using a robust web stack comprising React, Node.js, Express.js, and MySQL, the portal incorporates JSON Web Tokens (JWT) and Role-Based Access Control (RBAC) to enforce strict boundary checks across distinct user hierarchy levels including Super Admin, Staff, Trainers, Support Teams, and Students. Functional capabilities include prospective student tracking, registration processing, batch allocation, coursework assignments, internal billing, mock interviews, and automated performance summaries. By unifying these disjointed administrative modules into a single interface, CampusFlow streamlines operational throughput and simplifies end-to-end student lifecycle management.**

***Index Terms*—EdTech Portal, Administrative Automation, Student Lifecycle Management, Role-Based Access Control, Web Application Architecture, MySQL Data Store**

I. Introduction

The rapid digitalization of specialized skill training and higher education has amplified the organizational demands placed on training facilities. Beyond routine curriculum delivery, modern institutes must orchestrate candidate intake, schedule faculty shifts, record attendance metrics, manage billing status, and facilitate continuous communication across diverse user groups.

Relying on traditional methods—such as manual logbooks, standalone spreadsheets, or disconnected third-party applications—frequently triggers data discrepancies across departments. Repeatedly copying records between systems increases manual errors and creates communication bottlenecks between administrative, academic, and accounting units.

Deploying a centralized web infrastructure resolves these operational gaps by hosting core administrative workflows under a unified database architecture. A unified web platform bridges the gap between initial lead intake and final academic assessment, providing management with accurate insights while enabling learners to monitor their personal academic milestones through dedicated accounts.

This paper presents CampusFlow, a web platform created to bring administrative and instructional workflows together. Built around a consolidated relational database with role-driven authorization, the system manages candidate onboard-ing, batch coordination, attendance tracking, assignment submissions, fee accounting, interview preparation, and real-time alerts.

System security is prioritized through JWT-driven session validation paired with granular Role-Based Access Control (RBAC), ensuring that data access aligns with specific organizational duties [12]. The primary contribution of this work is an end-to-end operational framework that manages the entire student lifecycle within a secure, consolidated environment.

II. Related Work

Academic literature contains numerous studies on digital educational platforms, focusing on student analytics, learning management systems (LMS), cloud platforms, and access control models.

A. Performance Analytics Platforms

Recent studies explore computational frameworks to monitor student progress and predict academic risks [1], [2]. While effective for academic evaluation, these models generally omit enrollment, accounting, and operational workflows.

B. Learning Management Systems

Standard LMS platforms excel at resource distribution and online assessment submissions [3]. However, they rarely support lead conversion, fee handling, or batch scheduling.

C. Cloud and Security Infrastructure

Cloud deployments demonstrate the benefits of central storage and anytime access [4], while RBAC models provide clear authorization boundaries for educational data [5]. Additional research covers blockchain verification, college record systems, and educational data mining [6]–[11].

While prior studies target isolated functional areas, CampusFlow integrates enrollment, academic tracking, finance, mock interviews, and administrative reporting into a single operational workflow.

III. Research Gap and Problem Statement

Educational institutes often experience operational fragmentation by using separate tools for prospective lead tracking, course registration, tuition accounting, batch scheduling, attendance, and career preparation. This disconnected approach forces staff to duplicate data entry and maintain parallel records across multiple systems.

The underlying research gap is the lack of a cohesive, lightweight platform that connects lead management, fee accounting, academic progress, and interview evaluation within one unified data scheme. CampusFlow resolves this gap through a single web portal that maintains continuous data visibility from candidate inquiry to final course completion.

IV. Proposed System

CampusFlow uses a modular client-server framework where a web frontend connects to backend services and a centralized MySQL database. Access levels are strictly partitioned across six primary operational roles: Super Admin, Admin, Sales Executive, Trainer, Support Executive, and Student.

A. System Architecture

The proposed CampusFlow platform follows a layered client-server architecture consisting of role-based users, a security layer, a web application layer, and centralized data and service components.

Fig. 1 illustrates the overall system architecture of CampusFlow. The architecture separates user interaction, security enforcement, application services, and centralized data management.

The user layer consists of Super Admin, Admin, Sales Executive, Trainer, Support Executive, and Student roles. These users interact with the platform through authenticated communication. The security layer incorporates JWT authentication and RBAC middleware to validate user sessions and control access according to assigned responsibilities.

The web portal provides functional modules for lead and inquiry management, admission, batch management, attendance, assignments, mock interviews, invoice and installment tracking, reports and dashboards, and notifications. The backend REST API, implemented using Express.js controllers and services, processes requests and applies the required business logic.

The data and services layer uses a centralized MySQL database for persistent storage. Reports, analytics, notifications, and functional integration testing are supported through services connected to the centralized data layer.

B. Security and Access Policies

Security is enforced using JSON Web Tokens (JWT) for secure session management [12]. Upon login, the system evaluates the user's assigned role through RBAC middleware to grant or restrict access to specific API endpoints and frontend views.

The use of role-based authorization establishes a controlled access mechanism in which users can interact only with the resources and operations relevant to their assigned responsibilities.

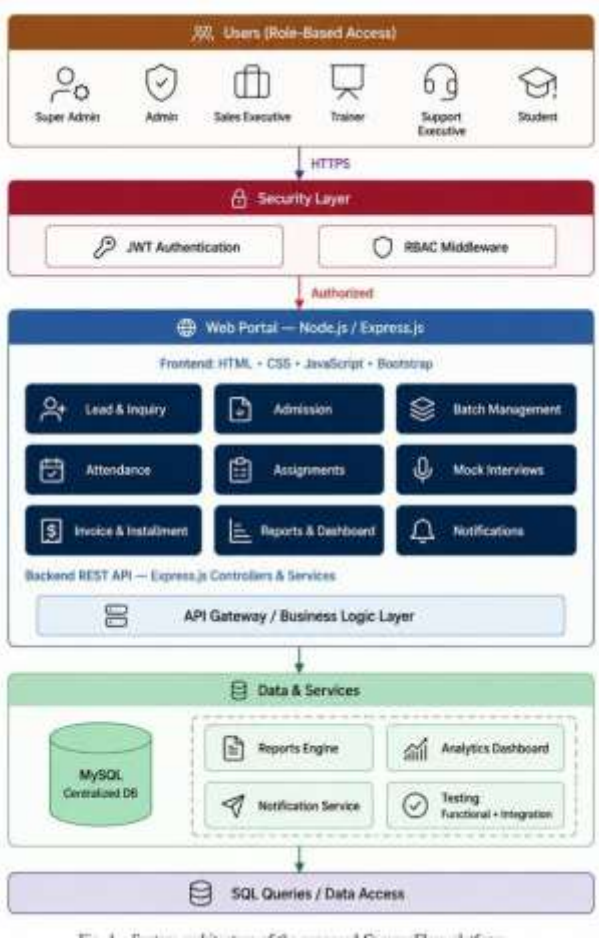


Fig. 1. System architecture of the proposed CampusFlow platform.

C. Core Functional Modules

CampusFlow integrates multiple operational modules within a single platform.

1) User and Course Management: This module handles account provisioning, profile status, and course structure definitions. It provides administrators with centralized control over user information and course-related configurations.

2) Batch and Admission Workflows: The admission workflow manages student registration and maps enrollments directly to active course batches. The system maintains admission status throughout the enrollment process.

3) Academic Tracking: The academic tracking module records student participation and manages assignment submissions. Attendance information is associated with corresponding batches and schedules.

4) Finance and Career Preparation: The finance module tracks internal fee payments and related financial records. The career preparation component organizes mock interview practice sessions to support student career readiness.

5) Dashboards and Reporting: The dashboard system provides students with a centralized view of their academic standing. Administrative users receive aggregated operational information and reports for monitoring institutional activities.

V. Implementation and Observations

The platform's underlying architecture was built and verified using React, Node.js, Express.js, and MySQL. The implementation follows a client-server approach in which the React-based frontend communicates with backend RESTful services developed using Node.js and Express.js.

Authentication and authorization mechanisms are incorporated into the backend through JWT-based session validation and RBAC policies. These mechanisms provide controlled access to application resources according to the assigned user role.

The centralized MySQL database maintains the relationships among users, courses, batches, admissions, attendance records, financial information, assignments, and other operational data.

Functional tests were performed to verify application workflows. The tests confirmed that API endpoints correctly process user requests, enforce authentication policies, and perform relational database updates without system errors.

The implementation demonstrates the feasibility of integrating academic and administrative operations within a single web-based environment.

VI. Discussion

By replacing isolated tools with a shared relational database model, CampusFlow minimizes duplicate data entry and establishes centralized visibility across operational activities.

The integration of admission, batch allocation, academic tracking, financial records, and career preparation allows information generated during one stage of the student lifecycle to remain accessible to relevant users in subsequent stages.

RBAC further strengthens the operational model by establishing clear security boundaries between administrative, academic, support, and student-level activities. Consequently, users can access functionality according to their assigned responsibilities rather than interacting with the complete system.

The centralized architecture also provides a foundation for future reporting and analytics capabilities by maintaining operational information within a common data environment.

VII. Limitations and Future Scope

A. Limitations

The current system has several practical constraints. First, the platform is designed for single-tenant deployment and is therefore focused on the operational requirements of a single educational or training institution.

Second, the system requires an active internet connection because it is implemented as a web-based platform. Third, the current implementation does not include dedicated native mobile applications for Android or iOS devices.

Finally, the platform does not currently provide live payment gateway integration and focuses on internal fee tracking rather than direct online payment processing.

B. Future Scope

Future enhancements can extend CampusFlow beyond its current operational scope. A multi-tenant architecture can be introduced to support multiple educational institutions through a common platform while maintaining isolated institutional data.

Dedicated Android and iOS applications can improve accessibility for students, trainers, and administrative staff. Online payment gateway integration can further automate financial transactions and reduce manual payment processing.

Another important direction is the incorporation of AI-driven predictive analytics. Such capabilities could assist institutions in monitoring student performance, identifying potential academic risks, and generating more informative performance insights.

VIII. Conclusion

CampusFlow provides a practical digital solution for training facilities seeking to streamline administrative and educational operations. By consolidating lead intake, student enrollment, course logistics, fee management, and academic evaluations into a secure web application, the portal reduces administrative workload and enhances institutional efficiency.

The combination of React, Node.js, Express.js, MySQL, JWT authentication, and RBAC provides a structured technical foundation for managing the student lifecycle through a centralized platform.

By connecting operational workflows that are commonly handled through isolated tools, CampusFlow establishes a unified environment for information management, academic coordination, and administrative monitoring. The proposed architecture also provides a foundation for future expansion through mobile applications, payment integration, multi-tenant deployment, and AI-assisted analytics.

Acknowledgment

The authors would like to express their sincere gratitude to their project guide Prof. S. S. Mamarde , faculty members, and the Department of Computer Science and Engineering, P. R. Pote Patil College of Engineering & Management, Amravati, for their guidance, support, and valuable suggestions throughout the development of this project.

References

[1] N. Naveenkumar et al., "EduConnect": A Secure and Analytics-Driven Smart Student Bio Portal for Academic Monitoring," in *Proc. 2026 ICSSCNA*, pp. 1747–1753, 2026, doi: 10.1109/ICSSCNA68616.2026.11546900.

[2] G. Yuan, "Design and Evaluation of Intelligent Student Management System Based on Multi-Modal Data," *Systems and Soft Computing*, vol. 7, Art. no. 200370, Dec. 2025, doi: 10.1016/j.sasc.2025.200370.

[3] V. Kamble, S. Mane, and D. J. Waghmare, "AI Powered Student Management System," *Iconic Research and Engineering Journals*, vol. 9, no. 6, pp. 16–24, Dec. 2025.

[4] S. Geetha et al., "Full-Stack Based Student Management Sys-tem and Proactive Dropout Mitigation of School Students Us-ing Machine Learning Algorithms," in *Proc. AICDMB 2025*, doi: 10.1109/AICDMB64359.2025.11277582.
[5] S. N. Abd Wahab and Z. Abdullah, "SIS: Student Information System for SMKTMI with RBAC," *Applied Information Technology and Computer Science*, vol. 6, no. 2, pp. 813–832, 2025.
[6] P. Chinnasamy et al., "Blockchain-Based Electronic Educational Doc-ument Management with Role-Based Access Control Using Machine Learning Model," *Scientific Reports*, vol. 15, Art. no. 18828, 2025, doi: 10.1038/s41598-025-99683-5.
[7] A. Raj et al., "College Management System," *International Research Journal of Computer Science*, vol. 11, no. 4, pp. 179–185, Apr. 2024, doi: 10.26562/irjcs.2024.v1104.08.
[8] S. R. Pendela et al., "Design and Implementation of CampusFlow: A Cloud Native Application for Educational Institutions," *IJRASET*, vol. 12, no. 3, pp. 2185–2196, Mar. 2024, doi: 10.22214/ijraset.2024.59305.
[9] S. Vincent-Lancrin and C. Gonza´lez-Sancho, "Education and Student Information Systems," *OECD Digital Education Outlook 2023*.
[10] S. Singla et al., "College Management System Using Blockchain," in *Proc. 7th ICIIP*, pp. 176–181, 2023, doi: 10.1109/ICIIP61524.2023.10535113.
[11] H. Yang and W. Zhang, "Data Mining in College Student Edu-cation Management Information System," *International Journal of Educational Studies*, vol. 15, no. 3, pp. 279–287, 2022, doi: 10.1504/IJES.2022.124841.
[12] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, IETF, May 2015.
[13] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2016.
[14] R. S. Pressman and R. B. Maxim, *Software Engineering: A Practi-tioner's Approach*, 9th ed. McGraw-Hill, 2019.
[15] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. Pearson, 2020.
[16] R. Elmasri and S. Navathe, *Fundamentals of Database Systems*, 7th ed. Pearson, 2017.