

Smart Campus Analytics Platform with AI Chatbot and ERP System

Samala Chandrasekharreddy¹, K. Srinivas Rao²

¹Department of Computer Applications, Aurora Higher Education and Research Academy, Hyderabad, India

²Asst Prof Department of Computer Applications, Aurora Higher Education and Research Academy, Hyderabad, India

Abstract

Institutions of higher learning are progressively reliant upon digital infrastructures for the administration of student records, attendance, examinations, assignments, fees, and various administrative operations. Nevertheless, once these functions become dispersed among distinct applications, both students and faculty members may encounter obstacles in retrieving information and in tracking academic progress. The present study delineates the design and implementation of a Smart Campus Analytics Platform incorporating Artificial Intelligence (AI) Chatbot alongside Enterprise Resource Planning (ERP) capabilities. Within this platform, academic and administrative workflows are consolidated into a single web-based application serving students, faculty members, and administrators. Among its principal features are student and faculty management, attendance tracking, academic records, assignment management, examination information, fee records, leave requests, timetable information, and campus notices. A central component is the Attendance Analytics Chatbot, which assists students in interpreting attendance records, computing their present attendance percentage, and ascertaining the minimum number of further classes needed to attain a designated attendance threshold. For the frontend, the system employs React and TypeScript; for the backend, Node.js and Express; for information management, MongoDB-compatible data models; and for conversational assistance, an optional integration with the Gemini language model. To regulate access to application functions, authentication and role-based authorization are implemented. This paper discusses the pertinent literature, system architecture, development methodology, workflow, implementation, and approach to functional evaluation. The proposed platform illustrates how ERP functions, academic analytics, and conversational assistance may be combined to enhance access to campus information and to facilitate academic decision-making.

Keywords: Smart Campus, Educational ERP, Artificial Intelligence, Academic Analytics, Attendance Management, Chatbot, Student Information System

1. Introduction

The swift advancement of Information Technology, Artificial Intelligence, cloud computing, and data analytics has fundamentally reshaped the manner in which educational institutions administer their academic and administrative operations. An expanding number of colleges and universities rely on digital platforms to store student records, track attendance, conduct examinations, disseminate announcements,

and synchronize institutional activities. Such technologies contribute to diminishing manual effort while enhancing the availability of academic information.

Enterprise Resource Planning (ERP) systems offer a centralized framework for governing institutional information and processes. Smart campus platforms build upon this notion by interlinking academic services, administrative operations, and digital technologies within a shared environment. Earlier investigations have examined intelligent campus management, big-data-driven platforms, data governance, Internet of Things (IoT) architectures, and educational ERP systems [1]–[12]. These investigations underscore the significance of integrated information management within higher education. Notwithstanding such progress, students may still be required to traverse several modules in order to ascertain their academic standing. Attendance data, for instance, are frequently displayed as a percentage or as a roster of recorded classes. Students may find themselves computing by hand the number of extra classes they need to attend so as to reach the required attendance percentage. Manual computations of this kind can prove cumbersome and may result in inaccuracies.

The proposed Smart Campus Analytics Platform with AI Chatbot and ERP System tackles these difficulties by consolidating selected academic and administrative functions into one web application. It grants distinct levels of access to students, faculty members, and administrators in accordance with their respective roles. The system further incorporates a chatbot interface designed to render academic information and supported assistance more readily reachable via natural-language queries.

The Attendance Analytics Chatbot constitutes a core component of the proposed platform. It draws upon existing attendance records together with a defined attendance threshold in order to compute the present attendance percentage and the quantity of additional classes needed to attain the target. This computation rests on attendance data and mathematical rules, rather than on a trained machine-learning prediction model.

The principal contributions of this work comprise the integration of academic and administrative ERP functions, role-based access to campus information, formula-driven attendance analytics, and conversational assistance delivered through an optional AI service equipped with a local fallback mechanism. hatbot, Student Information System

2. Related Work

Scholarship concerning intelligent campus management has explored the ways in which Artificial Intelligence may underpin campus operations, automation processes, and decision-making activities. An intelligent campus management approach powered by AI was investigated by Zhou et al. [1]. The creation of a university smart campus platform grounded in big-data technologies was analyzed by Huang [2]. The incorporation of smart campus components aimed at enhancing user experience was addressed by Elbertsen et al. [3]. Such investigations furnish pertinent groundwork for the construction of interconnected and user-centric campus services.

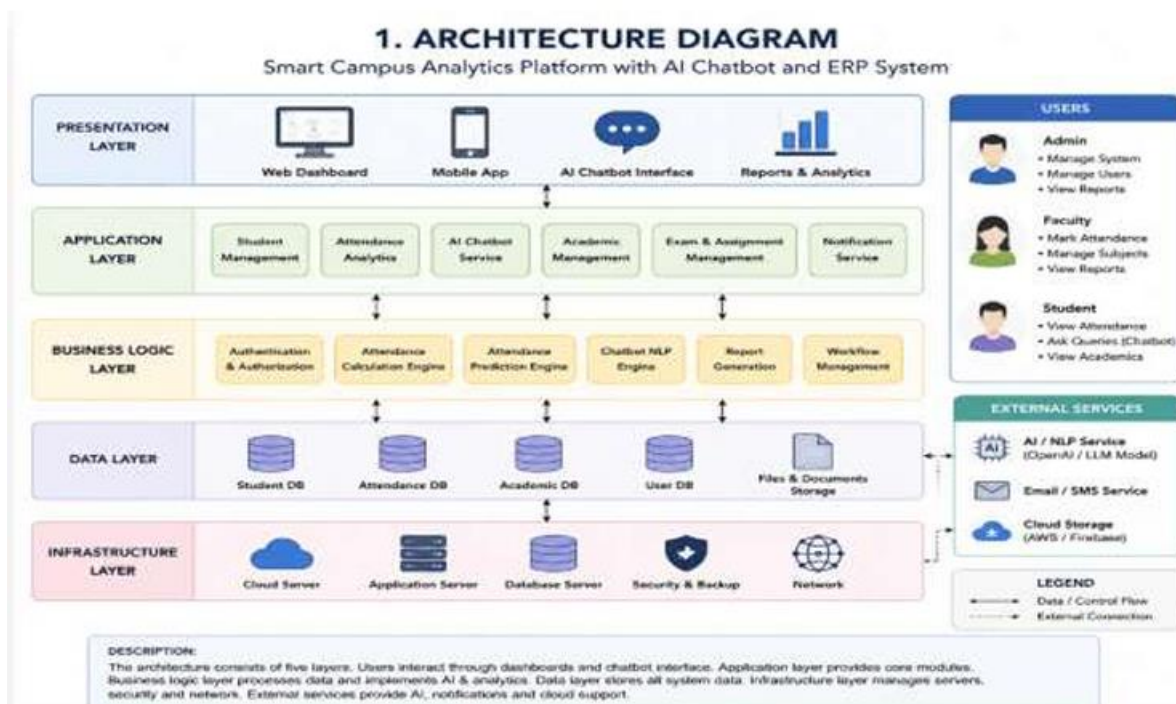
Within smart campus scholarship, data governance alongside information security likewise constitutes a significant domain of inquiry. Data governance within a university smart campus setting was scrutinized by Chen and Liu [4], whereas smart campus architecture together with data security safeguarding was

addressed by Ni and Zeng [5]. A versatile IoT architecture for a university campus was put forward by Zarpellon et al. [6], while the deployment of a smart-city model to a conventional university campus through big-data architecture was probed by Villegas-Ch et al. [7]. Information sharing, infrastructure integration, monitoring, and secure management are the concerns foregrounded by these lines of inquiry. Scholarship on educational ERP centers on the consolidation of academic and administrative processes. ERP implementation featuring a feedback module within higher education was analyzed by Patil et al. [8]. A web-based ERP portal for college administration built with React and Firebase was addressed by Mehrotra [9]. Unisphere, an ERP system designed for student management, was introduced by Pawar et al. [10]. An integrated education management system employing cloud computing was studied by El-Seoud et al. [11], whereas ERP system functionalities within higher education were analyzed by Noaman and Ahmed [12].

The body of literature examined attests to the significance of centralized information management, campus integration, and administrative automation. Nevertheless, the works encompassed by this project's literature review reveal a demand for tighter integration among ERP functions, student-oriented analytics, and conversational academic support. At the application level, the platform put forward here tackles this demand by furnishing a shared interface for chosen ERP modules, attendance calculations, dashboards, and chatbot assistance. This remark pertains to the scope of the studies reviewed and should not be taken to suggest that all extant campus systems are devoid of such capabilities.

3. System Architecture

Fig. 1. Architecture of the proposed Smart Campus Analytics Platform.



The platform being proposed is built upon a client-server architecture that operates over the web. It comprises a presentation layer, an application and API layer, a data layer, and a chatbot layer. Every layer

carries out a distinct role while interacting with the remaining layers so as to deliver integrated campus services.

3.1 Presentation Layer

The presentation layer is constructed with React and TypeScript. It furnishes the user interface intended for students, faculty members, and administrators. This interface encompasses dashboards and screens covering attendance, academic records, assignments, examinations, fees, leave requests, timetable information, notices, and administrative management.

Tailwind CSS serves the purpose of interface styling, whereas Recharts underpins chosen data visualizations. Lucide icons are employed to supply interface symbols. The presentation layer renders information and permits users to submit requests in accordance with the permissions assigned to them.

3.2 Application and API Layer

The application layer relies on Node.js and Express to manage API requests along with application logic. It handles requests originating from the frontend, validates data that arrives, coordinates access to stored records, and issues suitable responses.

Authentication is carried out by means of JSON Web Tokens (JWT), whereas bcrypt serves the purpose of password hashing. Role-based authorization distinguishes the functions of students, faculty, and administrators. The purpose of these mechanisms is to keep users from reaching operations that their roles do not permit.

3.3 Data Layer

The data layer employs Mongoose-compatible schemas intended for MongoDB-compatible storage. It accommodates structured data pertaining to users, courses, attendance, marks, assignments, examinations, fees, leave requests, timetables, and notices.

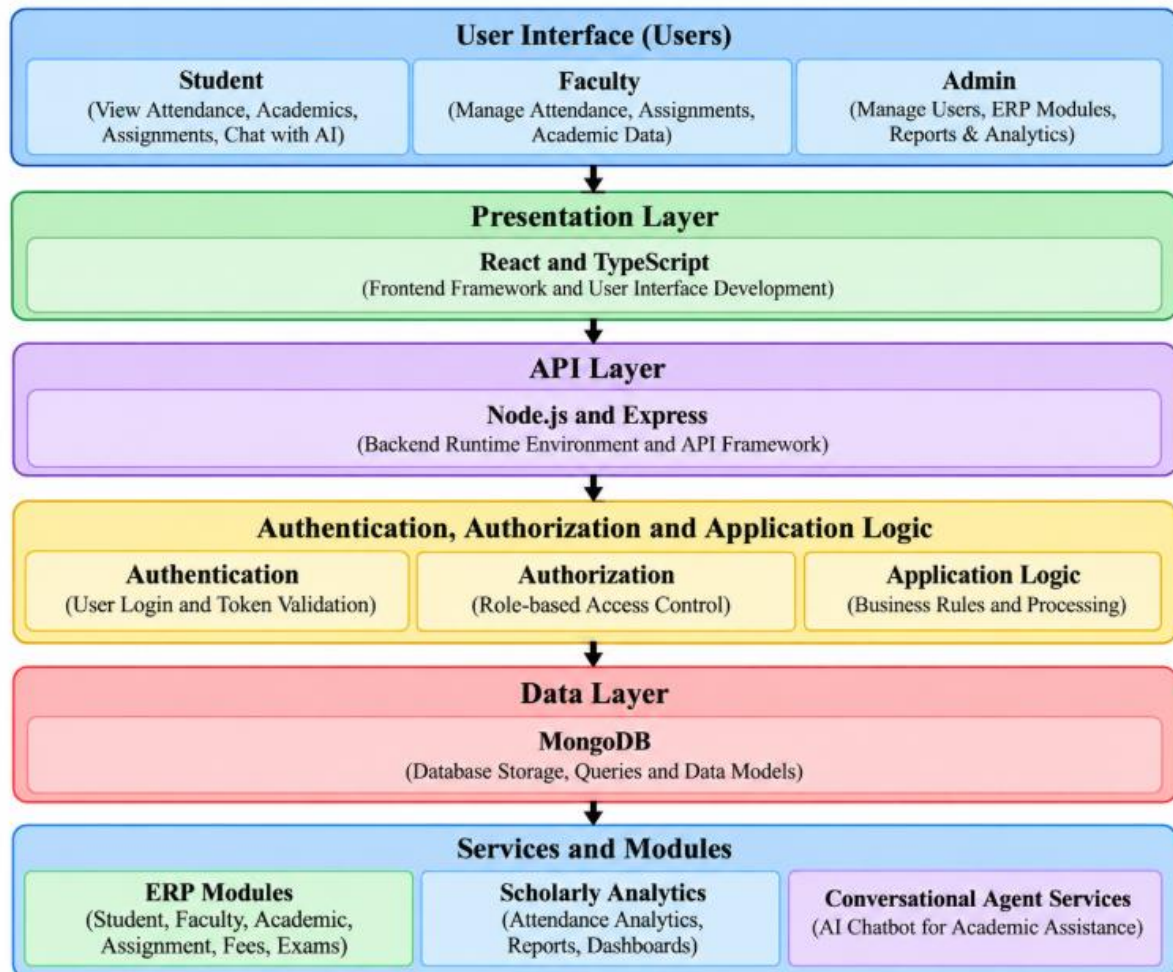
In a development configuration, the application is able to employ an in-memory persistence fallback whenever a database connection is not available. Seeded or demonstration records might likewise be employed throughout development and ought not to be construed as actual institutional records.

3.4 Chatbot Layer

The chatbot offers a conversational interface for academic queries that are supported. Where the Gemini service is configured and operational, it may be employed to produce natural-language responses. A local rule-based fallback accommodates a selected set of requests whenever the external service is not available. In the case of attendance-related questions, the calculation underpinning the response relies on attendance counts together with the configured target percentage. The chatbot conveys the outcome in a conversational format, whereas the mathematical computation continues to be separate from language generation. The mere deployment of a language model does not, in itself, demonstrate that the system undertakes machine-learning-based attendance prediction.

3.5 Architecture Overview

fig:2 System Architecture of the Proposed Smart Campus Analytics Platform with AI Chatbot and ERP System.



4. Methodology

The project adopts an application-development methodology that encompasses requirement identification, module design, implementation, integration, and the planning of functional verification. The development process draws upon the reviewed literature concerning smart campus systems, educational ERP platforms, data analytics, and AI-based student assistance.

4.1 Requirement Identification

Students, faculty members, and administrators constitute the primary user groups. These requirements were categorized into academic management, administrative management, analytics, and conversational assistance.

Students need access to their attendance records, marks, assignments, examination information, fee records, leave status, timetable, and academic assistance. Faculty members need access to the pertinent

teaching and attendance functions. Administrators need facilities for managing institutional records, users, and supported campus operations.

4.2 Module Design

The application is structured into interconnected modules covering authentication, user management, student and faculty records, attendance, marks, assignments, submissions, examinations, fees, leave requests, timetable information, notices, and chatbot assistance.

Shared data structures and role-based checks are employed to sustain consistent access to application functions. This modular approach enables the organization of diverse campus workflows within a single common application.

4.3 Attendance Calculation

Attendance analytics constitutes one of the principal functions of the proposed platform. It helps students understand their current attendance status and determine the minimum number of additional classes required to achieve a specified attendance threshold.

Let:

- A denote the number of classes attended.
- C denote the total number of classes conducted.
- T denote the target attendance percentage.
- x denote the minimum number of additional classes that must be attended.

The current attendance percentage is calculated as follows:

$$\text{Attendance Percentage} = \frac{A}{C} \times 100, C > 0$$

To determine the number of additional classes required to reach the target attendance percentage, the following condition is used:

$$\frac{A + x}{C + x} \times 100 \geq T$$

When the target attendance percentage is below 100%, the minimum number of additional classes can be calculated using:

$$x = \max\left(0, \left\lceil \frac{TC - 100A}{100 - T} \right\rceil\right)$$

In this expression, T represents the target attendance percentage, and the ceiling operation rounds a positive fractional result up to the next whole number. If the current attendance already meets the target, the number of additional classes required is zero.

For example, if a student has attended 26 out of 50 classes, the current attendance percentage is 52%. If the target attendance percentage is 65%, the minimum number of additional classes required is calculated as follows:

$$x = \left\lceil \frac{65(50) - 100(26)}{100 - 65} \right\rceil$$

$$x = \left\lceil \frac{650}{35} \right\rceil = 19$$

Therefore, the student must attend 19 consecutive additional classes, assuming no further absences, to achieve an attendance percentage of at least 65%. This calculation is based on recorded attendance data and does not predict whether the student will actually attend future classes.

The platform incorporates this calculation into its attendance assistance workflow. When a student submits an attendance-related query through the chatbot, the system uses the available attendance records to calculate the current percentage and the additional classes required to reach the specified target. The result is then presented to the student in an understandable format.

Fig. 3 The workflow of attendance analytics and required-class calculation is illustrated in.

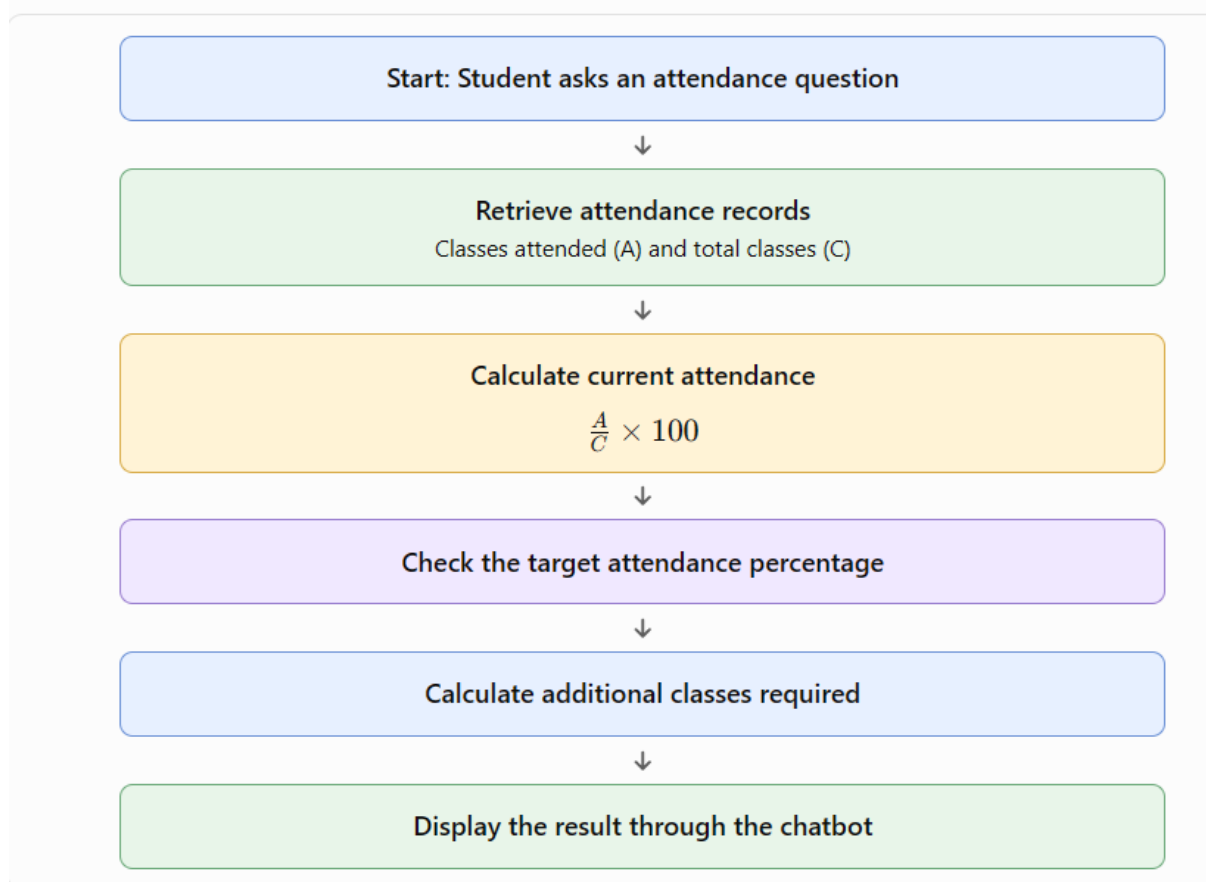


Fig. 3. Workflow of attendance analytics and required-class calculation.

4.4 Chatbot Integration

The chatbot serves as a natural-language gateway for the academic assistance that the system supports. A question is submitted by a user via the interface, after which the application handles that request in accordance with the functionality it supports. In cases where it is suitable, the system draws upon attendance records together with the calculation module in order to produce an answer.

In instances where the Gemini integration has been configured and can be accessed, it may be employed to generate conversational responses. Should the external service prove inaccessible, the local fallback is

capable of responding to the query types that are supported. The precise functionality that is available is contingent upon the configuration of the application and the requests that the implementation supports.

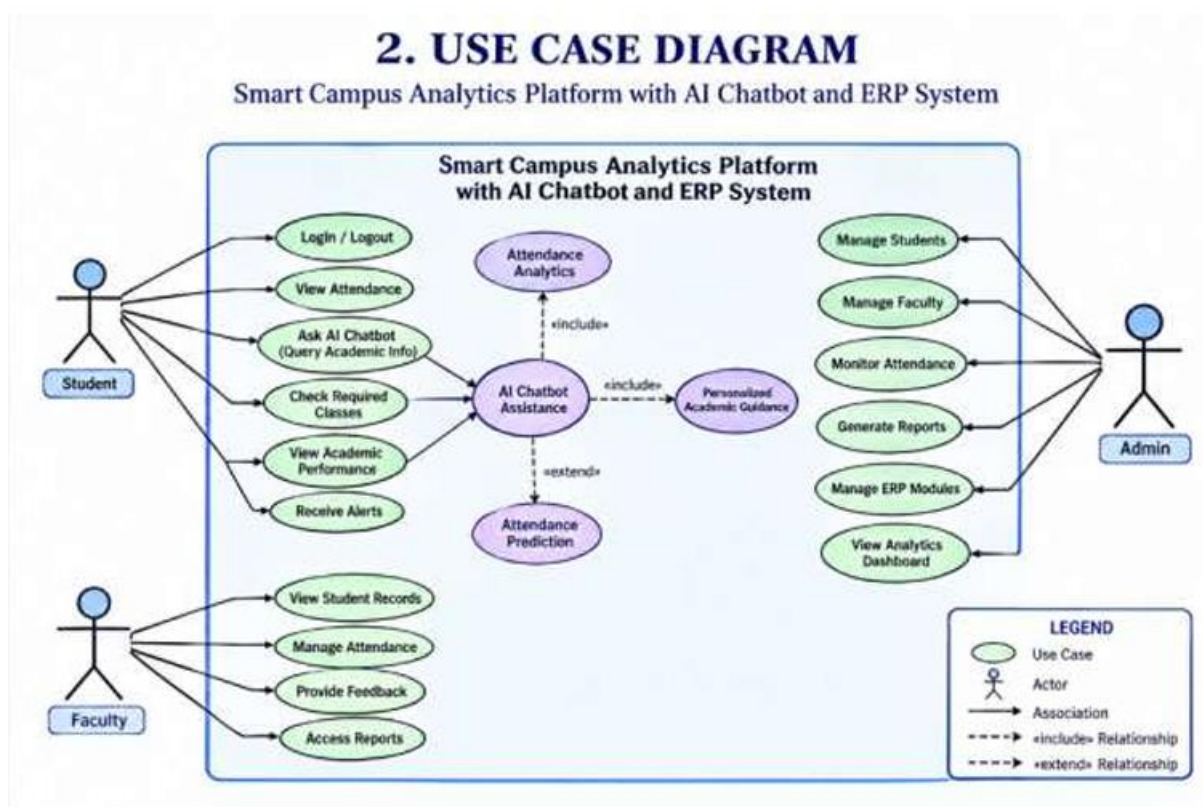
4.5 Functional Verification Approach

Functional verification ought to scrutinise authentication, restrictions tied to roles, data retrieval, attendance computations, chatbot responses, and standard ERP workflows. The test cases ought to encompass valid as well as invalid login attempts, attempts to access restricted operations, attendance computations at varying thresholds, and the submission or retrieval of academic and administrative records.

No independent benchmark measurements, results from user studies, or quantitative test logs were provided as evidence in support of this paper. Accordingly, the evaluation is articulated in terms of test scenarios and the observable behaviour that is expected, rather than through fabricated performance measurements or claims regarding accuracy.

5. System Workflow

Fig. 4. System Workflow of the proposed Smart Campus Analytics Platform



A standard system workflow is initiated at the point when a user logs in to the application. The credentials are validated by the backend, which then establishes an authenticated session. Subsequently, the interface presents the functions that are available in accordance with the user's role.

Students are able to retrieve their own academic and administrative information, which encompasses attendance, marks, assignments, examinations, fees, leave status, and timetable details. Teaching functions

allocated to faculty members are accessible to them, whereas administrators are able to manage institution-level records and the settings that are supported.

In the case of an attendance query, the application retrieves the pertinent attendance counts along with the target threshold. The current attendance percentage is determined by the calculation module and, where this is requested, the minimum number of additional classes needed to attain the target. The result is presented by the chatbot in natural language.

With regard to other ERP operations, the user chooses a module and either submits a request or retrieves an existing record. The request is processed by the backend, which applies the relevant validation and authorization checks and accesses the corresponding data. The returned information, or the updated status, is then displayed by the frontend.

Through a common interface, this workflow links academic and administrative functions together. Role-based authorization remains significant at every stage of the process, since users ought to access only those records and operations that their assigned roles permit.

6. Implementation

Fig:5 login page of the proposed Smart Campus Analytics Platform

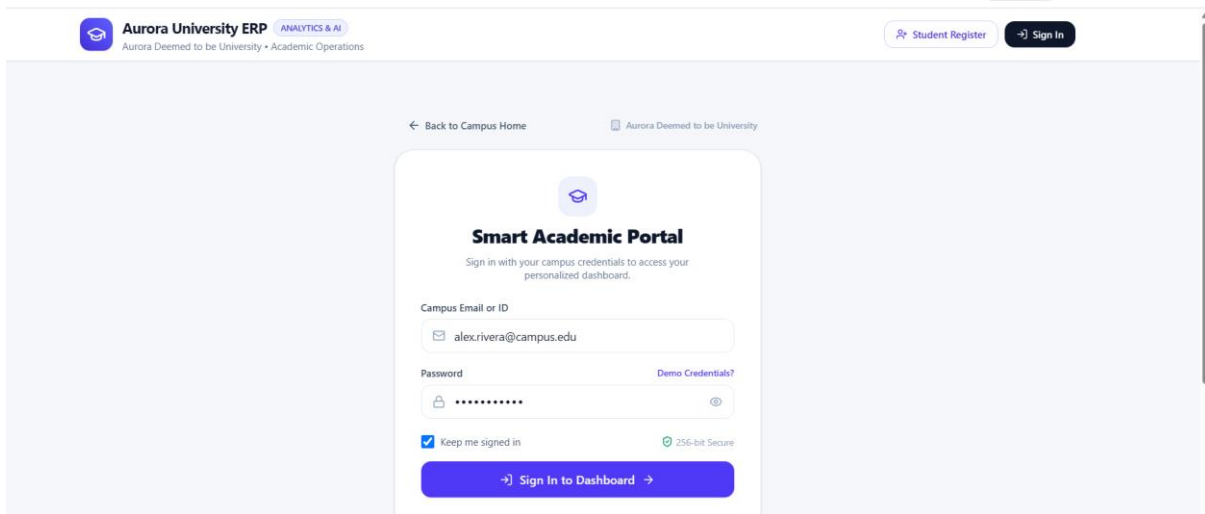


Fig:6 Student Dashboard Showing Attendance, CGPA, Enrolled Courses, and Examination Information.

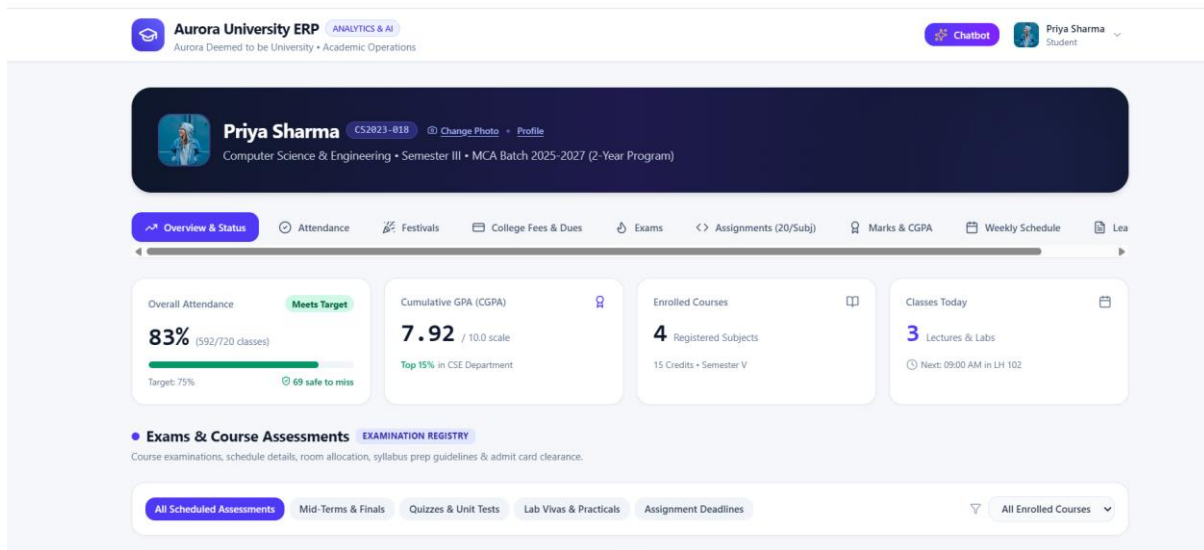
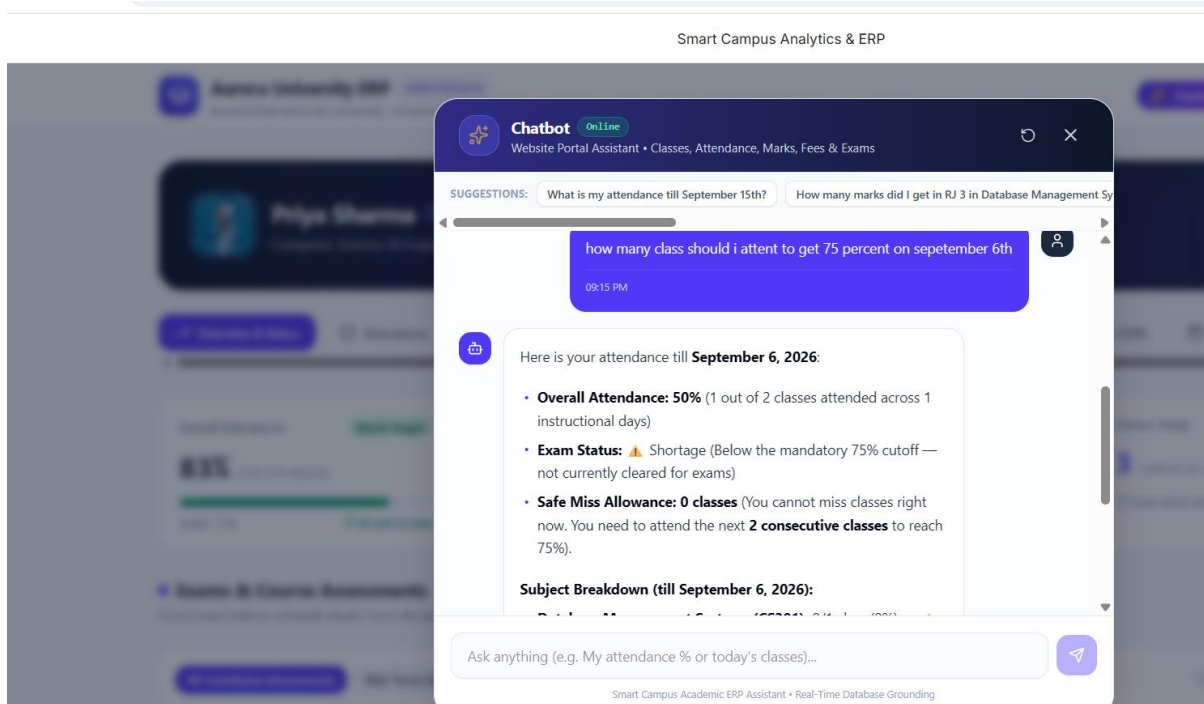


Fig:7 AI Chatbot Interface Showing Attendance Analysis and Required-Class Calculation.



The AI chatbot provides students with attendance-related information, including current attendance percentage, examination eligibility status, and the number of consecutive classes required to reach the prescribed attendance threshold. It enables students to access academic information through natural-language queries.

React and TypeScript serve as the technologies through which the frontend is constructed. Interface styling is provided by Tailwind CSS, while Recharts is employed to render chosen analytical visualizations. On the backend, Node.js and Express are utilized to structure API routes, manage request handling, handle authentication, and organize application logic.

Data models compatible with MongoDB underpin the storage of structured academic and administrative records. Sessions based on JWT, together with bcrypt password hashing, constitute the mechanism underlying the authentication flow. The chatbot component accommodates an optional Gemini integration as well as a local fallback that handles particular categories of queries.

The scope of the implemented project encompasses student, faculty, and course management; the recording and viewing of attendance; computations of attendance percentage and required classes; analysis of marks and grades; workflows for assignments and submissions; reflective journals and lab reports; scheduling of examinations; records of fees and installments; leave applications and their approval status; timetable and calendar information; campus notices; and the management of user credentials.

Which individual features are available is contingent upon the present application configuration and the permissions granted to the user. In certain environments, development or seeded data may be present, and such data ought to be differentiated from live institutional data.

The calculation of the attendance requirement is deterministic, resting on stored attendance counts and the designated target threshold. The chatbot functions as a conversational means of access and does not constitute evidence that a separately trained predictive analytics model exists. In a comparable manner, dashboards are capable of presenting information and facilitating interpretation, yet their mere presence does not establish that academic outcomes have improved.

7. Assessment of Functionality

The system can be evaluated by checking whether its main functions produce the expected observable behaviour. Table 1 presents representative functional test scenarios for the proposed platform. The table defines evaluation criteria and does not represent measured results from an independently completed test campaign.

Table 1. Functional Evaluation Scenarios

Area	Test Scenario	ExpectedObservable Behaviour
Authentication	Sign in with valid and invalid credentials	Valid credentials permit access, while invalid credentials are rejected
Role-based access	Attempt to access a function outside the user's role	Restricted operations are blocked or unavailable
Attendance analytics	Calculate attendance from attended and conducted class counts	The displayed percentage matches the attendance formula
Required-class calculation	Calculate classes needed to reach a target percentage	The result is the minimum whole number satisfying the target
Academic records	Retrieve authorized marks, assignments, and examination information	The interface displays records available to the authorized user
ERP workflows	Submit or review a leave request or fee record	The application displays the applicable record or status
Chatbot	Submit a supported academic query	The system provides a response based on the supported functionality

Fallback behaviour	Test supported chatbot queries when the external model service is unavailable	The local fallback handles the query types it supports
Data access	Attempt to retrieve information outside the permitted scope	Access is restricted according to authorization rules

The attendance calculation should be checked with multiple input combinations, including cases where the student already meets the target, has a low attendance percentage, or needs a non-integer number of additional classes before rounding upward. These checks help establish whether the calculation satisfies the specified mathematical condition.

Further evaluation should record actual test results, defects, response times, and user feedback. Such measurements would be required before making quantitative claims about performance, reliability, usability, or improvements in student outcomes.

8. Discussion

By combining a selection of educational ERP capabilities, academic information, and conversational support, the proposed platform delivers them within a single unified application. Its principal practical value lies in merging campus workflows with an attendance calculation function designed around the student. Rather than obliging students to work out attendance requirements by hand, the system is able to ascertain, from the records at hand, the minimum number of further classes required to attain a given threshold.

A shared structure for retrieving campus information is likewise afforded by the integration of student, faculty, and administrative interfaces. Operations accessible to distinct user groups are kept separate through authentication and role-based authorization. An additional interface for supported academic questions is supplied by the chatbot, whereas the local fallback furnishes limited continuity whenever the external language-model service is not available.

Limitations also characterize the project. The precision of attendance calculations rests upon the accuracy and completeness of the underlying attendance records together with the configured threshold. Chatbot replies are contingent upon the capabilities of the chosen integration or fallback rules. The availability of the external AI service is contingent upon its configuration and connectivity. Seeded demonstration records ought not to be regarded as proof of live institutional deployment.

Furthermore, the attendance feature as currently implemented computes the classes needed to attain a target; it does not employ a trained predictive model to estimate a student's future attendance behaviour. To establish how effectively the platform performs within a genuine campus setting, quantitative evaluation, usability studies, and testing with representative institutional data would be required.

9. Conclusion

This study introduced a Smart Campus Analytics Platform incorporating an AI Chatbot alongside an ERP System, conceived to unify academic and administrative operations within a single web-based application. The platform integrates role-specific interfaces with management of students and faculty, attendance

monitoring, academic records, assignments, examinations, fee details, leave applications, timetable information, and campus-wide communication.

The Attendance Analytics Chatbot offers a pragmatic instrument for interpreting attendance data and determining how many further classes are needed to attain a given attendance threshold. Through the separation of deterministic computation from conversational response generation, the system establishes a transparent foundation for attendance-related support.

The platform illustrates an application-level strategy for linking ERP workflows, academic analytics, and conversational assistance. Subsequent efforts ought to encompass systematic functional testing, usability assessment, deployment using verified institutional data, and evaluation of system performance. Such undertakings would yield evidence for evaluating the platform's dependability and practical worth within higher education.

10. Scope for Future Work

There are multiple avenues along which the platform may be developed further. Through the use of suitable historical data and properly validated machine-learning models, predictive analytics could be incorporated in order to estimate academic risk or patterns of future attendance. Accessibility for users with differing language preferences could be enhanced through multilingual chatbot support and voice-based interaction.

Further enhancements might encompass real-time notifications, support for mobile applications, integration with institutional identity systems, more comprehensive audit logging, and expanded reporting capabilities. Consideration should likewise be given in future evaluation to data privacy, security, usability, scalability, and the accuracy of the information the chatbot returns. The existing formula-based attendance calculation and any predictive feature ought to be evaluated independently of one another.

11. Acknowledgement

The author wishes to convey heartfelt appreciation to Mr. Srinivas, the project guide affiliated with the Department of Computer Applications at Aurora Higher Education and Research Academy, for the direction and assistance extended throughout the course of the project's development. The author further extends recognition to those scholars whose published investigations concerning smart campus systems, educational ERP platforms, data management, and intelligent educational technologies served as the groundwork upon which the literature review was constructed.

12. Authors' Biography

Samala Chandrasekharreddy is a postgraduate student pursuing Master of Computer Applications (MCA) in Data Science in the Department of Computer Applications at Aurora Higher Education and Research Academy, Hyderabad, India. His academic work focuses on software development, data science, and AI-based applications.

K. Srinivas Rao is asst prof, Department of Computer Applications at Aurora Higher Education and Research Academy, Hyderabad, India, and serves as the project guide for the Smart Campus Analytics Platform with AI Chatbot and ERP System.

References

1. Zhou Q., Peng J., Ma W., “Design of Intelligent Campus Management System Driven by Artificial Intelligence,” Proceedings of the 2025 International Conference on Artificial Intelligence and Educational Technology, 2025.
2. Huang D., “Design and Development of University Smart Campus Platform Based on Big Data,” EAI Endorsed Transactions on Smart Cities, 2023.
3. Elbertsen M., Kok H., Salimi N., “Designing the Future Smart Campus: Integrating Key Elements to Enhance User Experience,” Journal of Science, Technology and Policy Management, 2025.
4. Chen Z., Liu Y., “Research and Construction of University Data Governance Platform Based on Smart Campus Environment,” Proceedings of the International Conference on Information Systems and Data Management, 2021.
5. Ni Q., Zeng Y., “Research on Smart Campus System Architecture Design and Data Security Protection Strategy,” Journal of Digital Education and Technology, 2025.
6. Zarpellon B.O., da Silva R.M., Rodrigues A.P., “Design and Implementation of a Smart Campus Flexible IoT Architecture on a Brazilian University,” IEEE Access, vol. 12, 2024.
7. Villegas-Ch W., Palacios-Pacheco X., Luján-Mora S., “Application of a Smart City Model to a Traditional University Campus with a Big Data Architecture,” Sustainability, vol. 11, no. 10, p. 2857, 2019.
8. Patil S., Wagh S., Jadhav S., Ghadage R., Suryawanshi P., Jagtap P., “Exploring the Implementation of ERP with a Feedback Module in Higher Education: A Case Study,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 14, no. 5, 2025.
9. Mehrotra T., “ERP Portal for College Administration Using React and Firebase,” South Asian Journal of Artificial Intelligence and Blockchain Technology, vol. 2, no. 3, 2025.
10. Pawar N., Patil P., Jadhav O., “Unisphere: The ERP System for Student Management,” International Review of Engineering and Technology, vol. 9, no. 5, 2025.
11. El-Seoud S.A., Taj-Eddin I.A., Seddiek M., “Integrated Education Management System via Cloud Computing,” International Journal of Interactive Mobile Technologies, vol. 11, no. 2, 2017.
12. Noaman A.Y., Ahmed F.F., “ERP Systems Functionalities in Higher Education,” Procedia Computer Science, vol. 65, pp. 385–395, 2015.