

Smart Student Fee and Document Status Management System

Prof. S N V P Ravi Teja¹, Rohitha Sai Santoshi Gunnam², Rajeswari Panduri³, Nanda Gopal Kurru⁴,
Sharuka Jyothi Kasagana⁵

*Department of Electronics and Communication, Sasi Institute of Technology and Engineering,
Tadepalligudem*

Abstract— *The Smart Student Fee and Document Status Management System provides a safe way for schools to keep track of student records. This system helps schools get rid of outdated ways of keeping records and makes sure all student information is in one place. It is very important for educational institutes to have accurate and up-to-date information about student fees and documents, and also to take care of all student-related tasks, including tracking fees, checking document keeping, attendance records and looking at records. It follows a student from the time they join the school to the time they graduate and gives information for reports, such as whether a student is ready for a job or not. There are pages for people who handle fee status, academics and job opportunities. Students can make reports about student money and academic performance, which helps educational institutions make decisions based on facts, reduces the need for work and improves communication between the management and the students.*

Keywords — *Web-based application, Automation, Fee Tracking, Document Verification, MERN Stack, Secure Authentication.*

I. INTRODUCTION

Educational institutions around the world are facing a lot of pressure to manage and secure huge amounts of student data. This data includes administrative tasks and complex financial transactions. Usually, these processes like collecting fees, verifying documents and keeping records have been done using manual and paper-based methods. Research shows that these traditional methods are slow, labour-intensive, and prone to human error, which leads to a lot of repeated data and delays.[1] To solve these problems, people have suggested using system designs like Unified Modelling Language to manage large amounts of institutional data accurately.[2] Switching to administration has shown a lot of improvement in how institutions perform and how satisfied users are, especially in

vocational and higher education settings. New system architectures that use REST APIs and micro-services have also changed the sector by reducing work and keeping all school records in one place.[3] Apart from registration, the modern academic landscape needs modules to track Student Attendance and manage Backlog Records. These are necessary to keep a transparent student academic profile.[4] To build systems, people prefer using the MERN stack, which includes MongoDB, Express.js, React and Node.js. This is because it offers a modular design that can handle different types of data.[5] However, with access to data, security becomes a bigger concern. Implementing Role-Based Access Control and JSON Web Token authentication is essential to keep information private and only accessible to authorised personnel.[6] This technological framework supports a performance digital environment that can handle multiple tasks at the same time without slowing down.[7] Today's educational systems are expected to do more than just track administration.[8] They should include Placement Analysis features that evaluate student career eligibility and placement readiness based on their academic history. This gives both students. Coordinators have a strategic advantage.[9] One of the critical areas of automation is tuition fee management. Automated systems ensure real-time tracking of payments, instant receipt generation, and complete financial transparency, which are vital for accurate institutional bookkeeping.[10]

The proposed Smart Student Fee and Document Status Management System integrates all these features into a unified and secure platform. It provides dashboards for specialised Administrators, such as those in the Accounts office, Academic cell and Placement department. This way, every stakeholder can securely. Verify information relevant to their role, ultimately bridging the gap

between the college administration and the student body. The Student Fee and Document Status Management System is designed to make things easier for educational institutions to manage Student Attendance, Backlog Records and other tasks. The system is also meant to provide a transparent way to manage student data and fees.

II. OBJECTIVE

The main goal of this system is to get rid of paperwork. It wants to use a digital platform to do tasks that the institution has to do, which aims to show student fees and balances in a timely manner, and will help to reduce mistakes that people make. Students can check attendance, fee status, academic performance & documents which are uploaded by the administration. Students can view the job placement offers if they are eligible for certain company. This keeps the student information private. It also helps the educational institutes for better communication between the administration management and the students to clarify their changes.

III. LITERATURE SURVEY

The internet and new web technologies have really changed how schools and universities are run, especially when it comes to keeping track of student records and money. People have been trying out ways to make things work better for educational institutes:

In 2024, Pasaribu et al. [1] proposed a web-based Student Information Management System using UML modelling and black-box testing. The system manages student profiles, academic records, and achievements using a centralized database. It improves data accuracy, reduces redundancy, and enhances the administrative efficiency. However, this system lacks advanced analytics and intelligent decision-support features.

In 2023, Das et al. [2] proposed a student information system that integrates enrollment, grade management, and attendance tracking in a single platform. The system reduces manual record keeping and improves the transparency of the process. Real-time notifications enhance communication among users. However, deployment requires technical infrastructure and regular

maintenance. In 2023, Saud et al. [3] developed a cross-platform student information system using both Flutter and Firebase. The system includes authentication, attendance tracking, and grade management. Cloud-based storage improves scalability and accessibility via mobile devices. In 2022, Falebita [4] proposed a secure web-based student information management system with authentication and data protection mechanisms. The system enhances confidentiality and protects academic records of students. It improves security compared to traditional systems. In 2022, John et al. [5] developed a student information management system for enrollment, attendance, and grade tracking. Report generation helps to monitor academic performance. However, its scalability and real-time communication features are limited. In 2021, Agarwal et al. [6] proposed a comprehensive student information system for managing academic and institutional data. The system improves data storage and retrieval and administrative efficiency. It reduces manual documentation and enhances record-keeping. In 2021, Al - Hunaiyyan et al. [7] investigated the user experience in student information systems. This study emphasized usability, accessibility, and responsive design. An improved interface design increases system adoption and user satisfaction. However, enhancing the user experience increases the complexity of development. In 2021, Chandima et al. [8] proposed a web-based student information system to replace manual processes. The system manages academic records and improves accessibility for both administrators and students. Centralized storage reduces redundancy and improves efficiency. However, reliable Internet and server infrastructure are required. In 2019, Heryanda et al. [9] implemented a student information system for discipline and database management in Indonesia. In 2018, Krishna et al. [10] developed a Student Information System with dashboard analytics to analyze student performance. The system uses data mining techniques to generate reports and assist in decision-making.

Despite all these developments, it is still hard to make one system that can do everything a school needs, from keeping track of fees and documents to tracking attendance and helping with careers and placement analysis also keeping all the information safe for both the educational institutions and the students.

IV. PROPOSED METHODOLOGY

This system is made using an organised method. This method makes sure the system is good for the college and keeps the data safe. They want to make sure the system can track fees, check documents, and watch over the students' academic work. This means different departments like Accounts and Academics have their sections which are built using the MERN Stack, which is a group of tools that includes MongoDB, Express.js, React and Node.js. The system also has a feature that helps with placements and academic records based on how well the students are doing in their studies. To keep

everything, the system uses Role-Based Access Control, which is like a secure door that only lets certain people in. This door is locked with keys called JSON Web Tokens. The system makes sure all important work, like updating fees and checking documents, is done in a way. All of this is done to make sure the system is safe, easy to use, and works well. The end result is a system that's easy to understand works very well and gives a complete view of a students time, at the college. The Smart Student Fee and Document Status Management System helps get rid of mistakes that can happen when people do things by hand.

Flow Diagram:

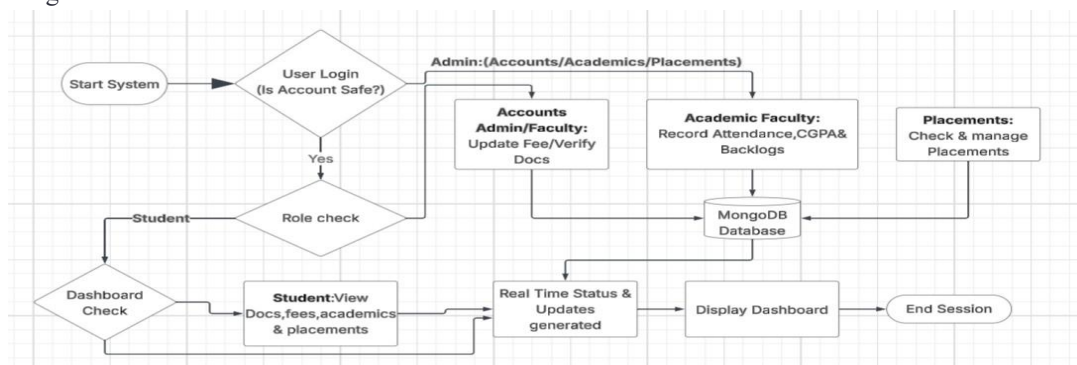


Fig 1: Flow Diagram

Block Diagram:

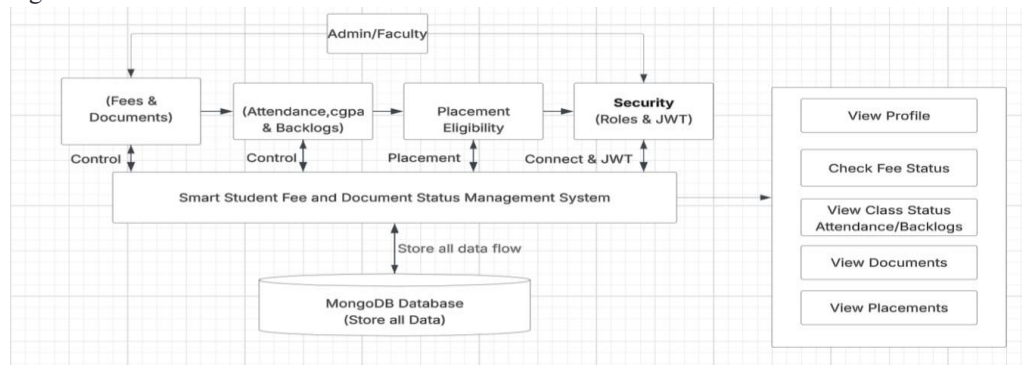


Fig 2. Block Diagram

Technologies Used:

Node.js: It is used as the backend runtime environment to handle server-side operations. It supports asynchronous processing and improves the system performance for multiple user requests.

Express.js: It is used to build RESTful APIs and manage routing. It simplifies backend development and enables communication between the frontend and the database.

React.js: Used to develop the frontend user interface. It allows the creation of reusable

components and provides a fast, responsive single-page application.

Tailwind CSS: CSS was used to style the user interface. It provides utility-based classes for quickly designing responsive and consistent layouts.

Vite: Used as the frontend build tool. It provides a faster development server startup and optimizes production builds.

MongoDB: It is a database that is used as an NoSQL database to store student data, attendance, and user information. It supports flexible schemas and

efficient data retrieval.

Postman: It is a tool to test the API endpoints. This helps verify request handling and responses and ensures proper communication between the frontend and backend.

V. WORK FLOW STEPS

- User identification is provided to Students and the Administrator so that the user can login by the creation of his/her account with the use of secure password and email. It has occurred.
- The user with their role is redirected to the particular dashboard, like Student, Accounts Admin, Academic Admin, Placement Admin and so on.
- The data input is done by three people: the Departmental admin enters student data, the Accounts Admin updates the fee status of the student; the Academic Admin enters attendance and backlog of.
- The students may upload automatically generated mandatory files and certificates through their own dashboard for verification.

- The core of the system processes the input data, so this data will always be real-time across students. Student data are automatically monitored through Emergence Analysis. It includes the CGPA, percentage of attendance, and the number of backlogs.
- These data may or may not be analyzed. Students can check their live fee payment status, document verification status, academic progress and much more.
- Placement Alerts: Based on the assessment of student eligibility, alerts will be given, with respect to upcoming jobs for eligible students.

VI. RESULTS AND DISCUSSIONS

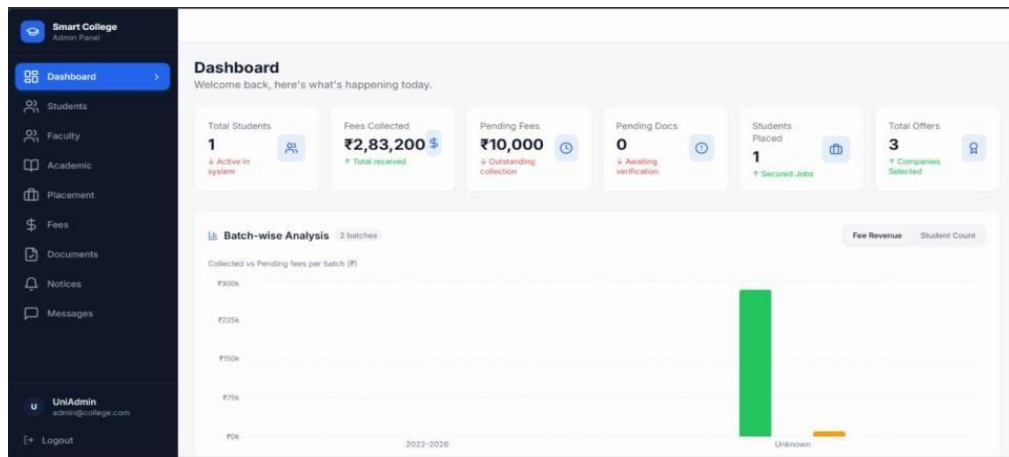
1. Login:

To log in as an admin, faculty, or student, select “sign in.” If you are new, click on “Request Approval.” The registration form will appear. Enter the required details and click on “Submit Request.” While registering, choose whether you are a “Student” or “Faculty.” Invalid logins are not allowed.

2. Admin Access:

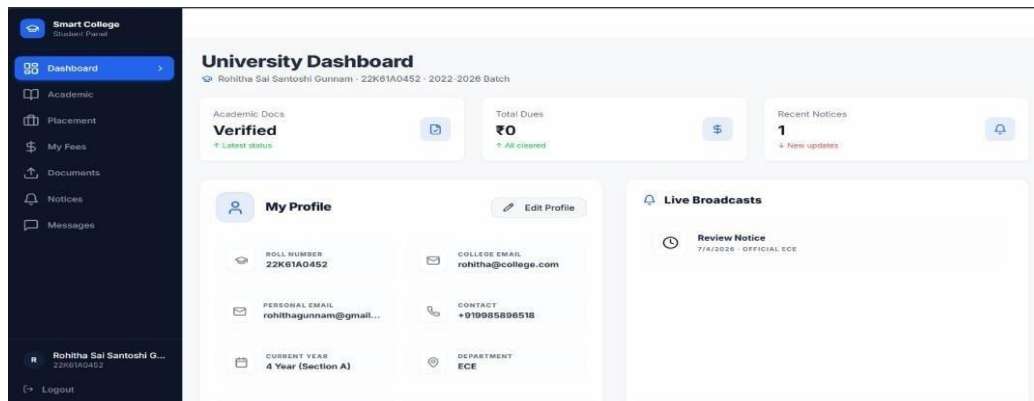
In admin dashboard, it shows total students, fees collected, pending fees, pending docs, students placed, and total placements. It shows the data in an analytic way, like a column-wise graph. The admin approves students, faculty, and fee information.

Check the documents and upload them to the website that are related to the students. Admins can see the status of academics and placements. The notice module is used to send important messages. The message module is used for talking between students and administrators.



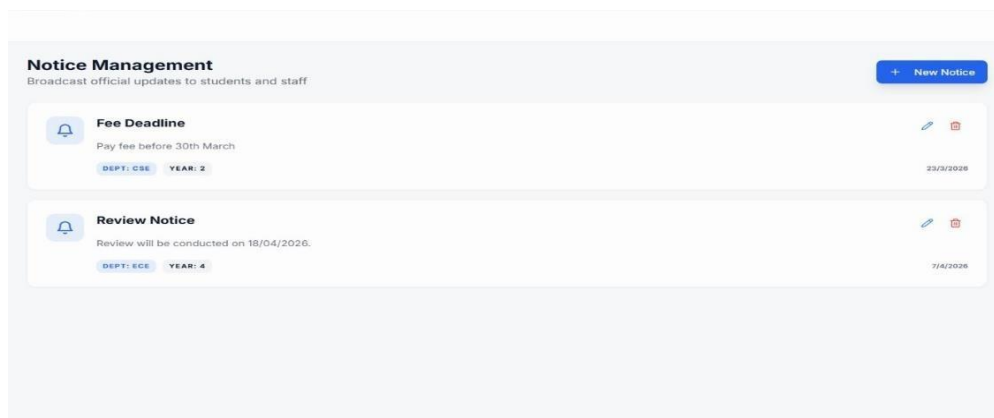
3. Student Access:

The student dashboard shows the status of documents, the total amount owed, recent notices, the student's profile, and live notice alerts. Students can see their academic, fee, document, and placement status. Students can send a direct message to the admin to fix any problems with updates.



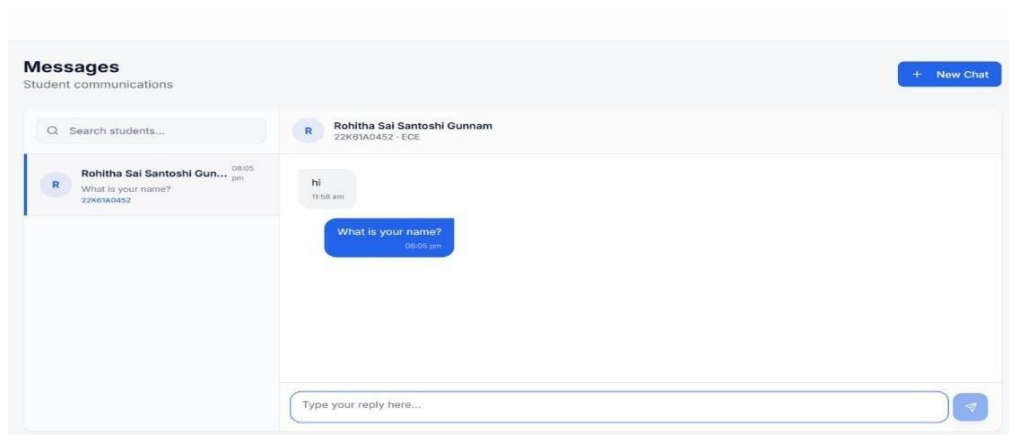
4. Notice Module:

Admin posts the important news, event information, fee details, document submission details, and other alerts.



5. Message Module:

The message module makes it easy for students and the admin to talk to each other. It cuts down on the amount of manual work needed for any kind of updates.



3. Comparison Table:

Parameter	Traditional management System	Proposed System
Data Handling	Manual paper based or scattered excel sheets	Centralized automated database using MongoDB
Time Consumption	High effort for manual verification and also to record updates	Instant updates and real time status viewing
Personalization	Difficult to track individual student progress in real-time	Provides a dedicated dashboard for every student
Placement Logic	Manual checking of eligibility lists	Analysis based on attendance and backlogs
User Experience	Moderate, which requires physical visits for status checks	High; user-friendly web interface with PDF downloads

VII. FUTURE INSIGHTS

- We are also developing an Android and iOS application for students to get quick access to their academic and fee status.
- The AI Placement Predictor predicts the most suitable job roles for students in an easy manner. The predictions of job roles are made through the student's marks and skills.
- Make payment option UPI or Net Banking so students can directly pay fees from the portal. Integration of Payment Gateway.
- In a situation, SMS or Email can be sent to either the student or their parents informing them about any issue regarding low attendance/fees issue.
- The primary benefit of an AI Chatbot assistant is the fact that its operations are completely monitored, can be up-scaled and can be re-modelled to achieve specific goals.
- Event and Exam Alerts keep you informed about our college's upcoming festivals and examinations. It presents city-level events on the digital calendar.

VIII. CONCLUSION

In this research paper, I am talking about the design and implementation of an automated system to manage institutions. This system makes college administration better by letting people track things in time. I used the stack, which includes MongoDB, Express.js, React and Node.js to make the system strong, secure and easy to use. The system has dashboards for students and staff. It also allows different departments to manage their tasks like tracking fees, attendance and old records. This system reduces the need for people to do things manually, removes data and makes sure students can always see their academic and financial information. The automated institutional management system helps close the gap between the college administration and the students, making things more transparent, efficient and based on data, for everyone involved.

REFERENCES

- [1] J. S. Pasaribu and I. S. Argadikusuma, "Design and Testing of a Web-Based Student

- Information Management System," International Journal of Engineering, Science and Information Technology, 2024.
- [2] K. L. Das, P. G. Peeraiah, K. Akash, and U. K. Paswan, "Student Information System for Academic Data Management," International Journal of Engineering Research & Technology (IJERT), 2023.
- [3] M. Saud, A. Khan, and S. Ahmed, "Student Information Management Using Flutter and Firebase," Journal of Software and Cloud Computing, 2023.
- [4] O. Falebita, "Secure Web-Based Student Information Management System," International Journal of Computer Science and Information Security, 2022.
- [5] J. John, R. Mathew, and P. Thomas, "Student Information Management System," International Journal of Engineering Research & Technology, 2022.
- [6] A. Agarwal, S. Verma, and R. Gupta, "Research Paper on Student Information System," International Journal of Advanced Research in Computer Science, 2021.
- [7] A. Al-Hunaiyyan, R. Alhajri, and A. Al-Sharhan, "Student Information System: Investigating User Experience," International Journal of Human Computer Studies, 2021.
- [8] D. Chandima, K. Perera, and S. Fernando, "Web-Based Student Information System for Educational Institutions," International Journal of Information Systems and Technology, 2021.
- [9] I. M. Heryanda, N. K. Sutarsa, and G. P. S. Wijaya, "Implementation of Student Information System to Improve Discipline and Database of Students," International Journal of Information Technology Education, 2019.2
- [10] K. V. Phani Krishna, M. Mani Kumar, and P. S. G. Aruna Sri, "Student Information System and Performance Retrieval Through Dashboard," International Journal of Engineering and Technology, vol. 7, no. 2.7, pp. 682–685, 2018.