# EduCloud Enterprise: A MERN-Stack Based Cloud ERP System for Real-Time Institutional Management

**Dabhi Het Bharatbhai**

*Department of Computer Science Engineering*

Guide: Vanshika Tiwari | Industry Mentor: Dhaval Shah

***Abstract***

*Managing daily tasks in educational institutes using manual paperwork takes a lot of time and creates many mistakes. Tasks like tracking student fees, taking daily attendance, and sharing study materials need a better digital solution. This research paper explains the development of "EduCloud Enterprise," a cloud-based Enterprise Resource Planning (ERP) system built entirely using the MERN stack (MongoDB, Express.js, React.js, and Node.js). The main goal of this project is to provide a fast, secure, and budget-friendly software alternative to heavy commercial ERP systems. The system includes several automated modules such as a Smart Fee Ledger, a Digital Resource Library, Real-Time Attendance Tracking, and Dynamic Academic Scheduling. By using a Single Page Application (SPA) design, the website loads almost instantly without refreshing. Security is handled using JSON Web Tokens (JWT) to make sure students only see their own private data. The results of this project show that building a custom ERP with modern web technologies greatly improves the daily management of coaching institutes and saves hours of manual administrative work.*

***Keywords*** *— MERN Stack, ERP System, Web Development, Educational Management, React.js, Cloud Computing, Database Management.*

## I. INTRODUCTION

In today's digital world, schools, colleges, and private coaching centers need fast computer systems to handle their daily work. Large universities easily buy expensive Enterprise Resource Planning (ERP) software, but mid-sized coaching centers find these big systems too costly and very hard to understand. Because of this, many small institutes still use old Excel sheets to calculate fees and WhatsApp groups to share study notes.

When data is scattered across different apps, it becomes very difficult for the admin to track which student has paid the fees and who is absent. EduCloud Enterprise was built to solve this exact problem. It provides a single, centralized web platform where admins, teachers, and students can interact easily.

Instead of using older web languages like PHP where the page reloads on every click, EduCloud uses modern JavaScript technologies. This research focuses on how the MERN stack was used to create a fast, secure, and highly interactive system. The project automates daily tasks, meaning the admin does not need to calculate pending fees manually; the system does it automatically.

## II. LITERATURE REVIEW

Before starting this project, many existing educational tools were studied. Applications like Google Classroom are excellent for submitting homework and video calling, but they completely lack financial tools like fee ledgers and payment tracking. On the other hand, full-scale ERP systems have hundreds of features like transport management and hostel tracking, which a normal coaching institute does not need.

Recent studies show that using JavaScript for both the frontend website and the backend server makes development much faster. Also, moving from traditional session-based logins to token-based security (JWT) provides much better protection against hackers. This project uses these modern development methods to create a lightweight system that perfectly matches the needs of a modern coaching institute without any extra useless features.

## III. METHODOLOGY

The development of EduCloud Enterprise was done step-by-step using the Agile methodology. This means the project was built module by module, starting from the database design to the final user interface.

### *A. Application Architecture*

The system is built using a 3-Tier Client-Server architecture:

- **Frontend (User Interface):** Built with React.js. It controls what the user sees on their screen. Tailwind CSS was used to make the website responsive, meaning it works perfectly on mobile phones, tablets, and computer screens.
- **Backend (Server Logic):** Built with Node.js and the Express.js framework. This part handles all the calculations, like subtracting paid fees from the total amount and verifying passwords.
- **Database (Storage):** MongoDB was used to store all the data. It is a NoSQL database, which makes it very fast to fetch student details and attendance records.

## IV. CORE SYSTEM MODULES

The system is divided into four main modules. Each module targets a specific problem faced by the coaching institute.

### *A. User Authentication and Dashboard*

Security is the most important part of the system. The application uses the bcrypt library to hide user passwords in the database. When a person logs in, the backend creates a secure JSON Web Token (JWT). The system reads this token to check if the user is an Admin, a Teacher, or a Student. If a student tries to open the admin panel, the system immediately blocks them.


Fig. 1. Secure login portal and main administrative dashboard.

### *B. Smart Fee Ledger*

Keeping track of fee installments in a notebook causes many calculation mistakes. In this module, the admin enters the total course fee when a student joins. Whenever the student pays an installment, the admin records it in the system. The backend automatically subtracts the paid amount and updates the "Pending Due" instantly.

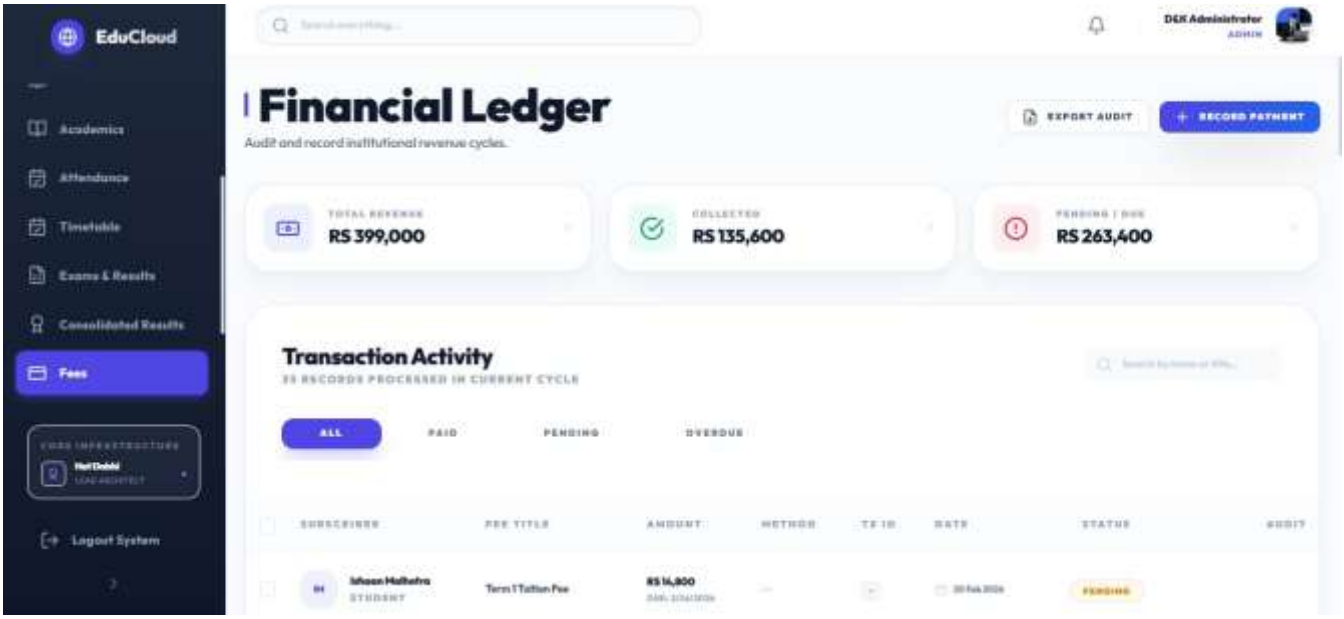


Fig. 2. Automated smart fee ledger calculating pending student dues.

### *C. Digital Resource Library*

Sharing study notes in messaging apps is very confusing because old files get lost. EduCloud includes a highly organized digital library. Teachers can upload PDF notes and assign them to a specific class batch. When a student logs in, the backend filters the data so the student only sees the notes meant for their exact batch.

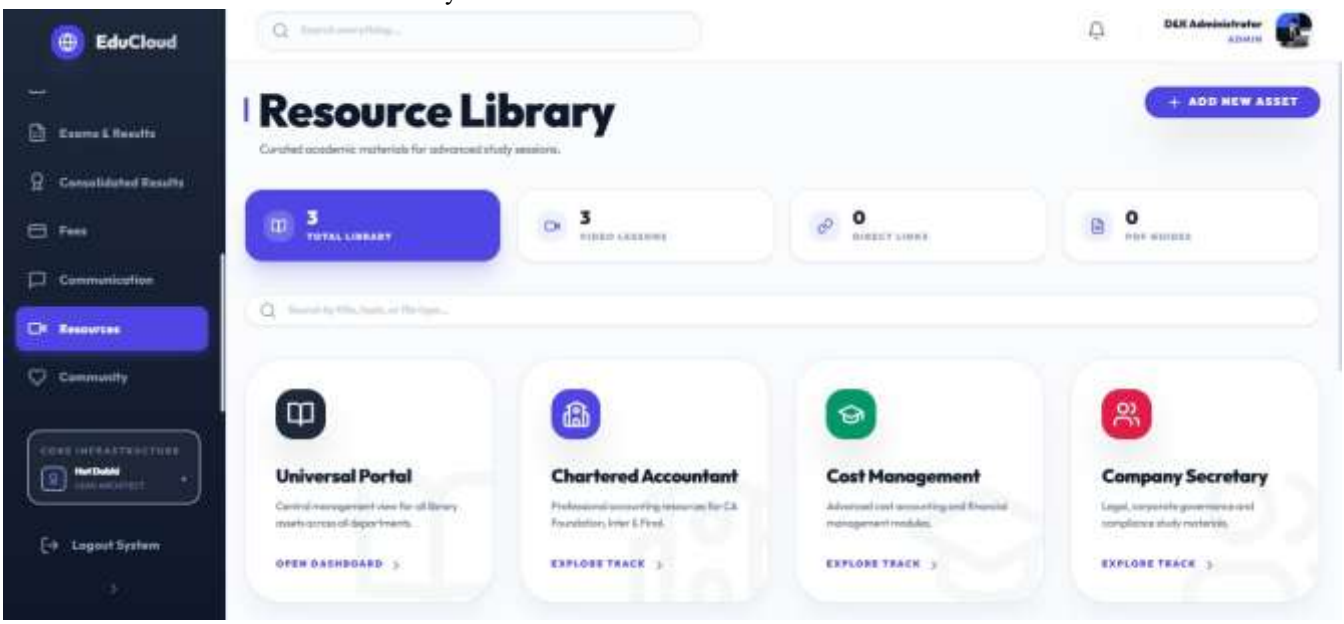


Fig. 3. Secure digital resource library for batch-wise study materials.

### *D. Real-Time Attendance Management*

Taking attendance by calling out names takes 10 to 15 minutes of class time. With the EduCloud attendance module, teachers get a digital list of students on their screen. Because it is a React Single Page Application (SPA), the teacher can click "Present" or "Absent" for 50 students in just a few seconds without the webpage reloading. The data is saved instantly to MongoDB.

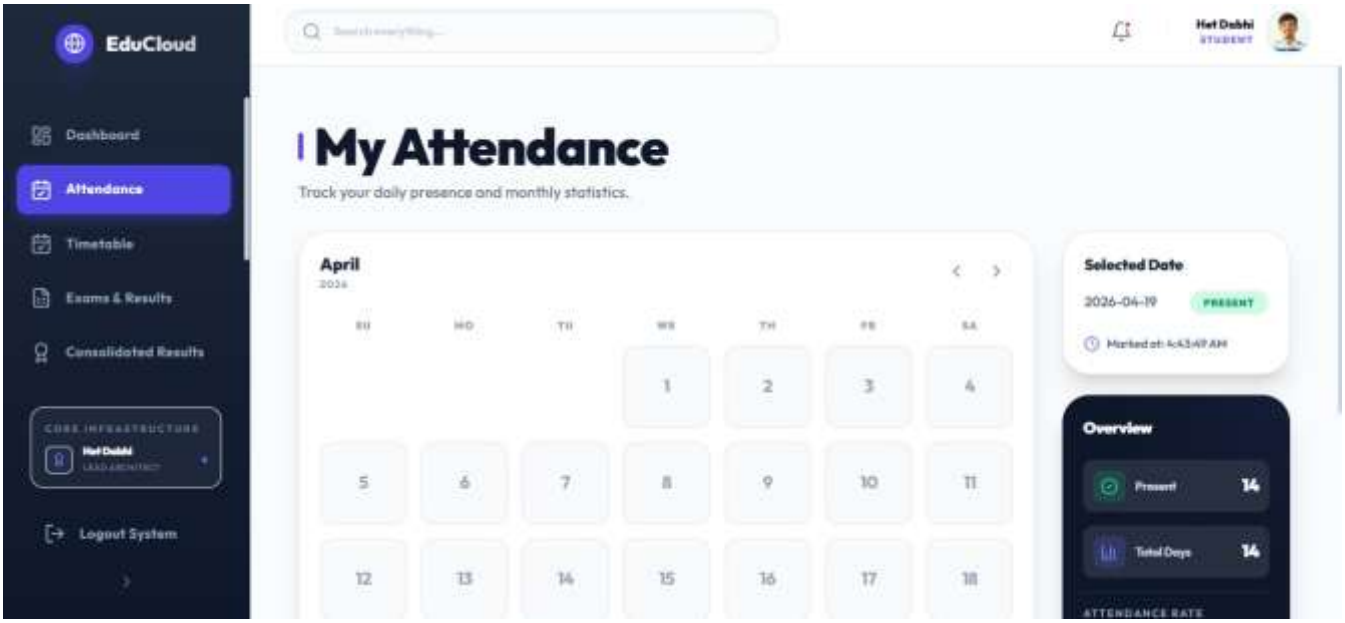


Fig. 4. Real-time digital attendance tracking module.

## V. SYSTEM PERFORMANCE AND TESTING

Testing was done continuously during the development of the project to ensure there were no bugs.

- ***API Testing:*** All backend routes were tested using Postman. The connection between the Node.js server and the MongoDB database proved to be very fast. Fetching a full list of student records takes less than 300 milliseconds.
- ***UI/UX Testing:*** The user interface was tested on different screen sizes. All buttons, tables, and navigation menus adjust perfectly whether the user is on a laptop or a mobile phone.
- ***Security Testing:*** Tests were conducted to ensure that one student cannot access another student's fee details or attendance records, proving the JWT security is working correctly.

## VI. CONCLUSION

The EduCloud Enterprise project successfully demonstrates how modern web development tools can solve real-world management problems. By building the system entirely on the MERN stack, the project achieved its goal of providing a fast, secure, and easy-to-use platform. Features like the smart fee ledger and digital library replace slow manual paperwork with instant digital actions. The system is lightweight, highly scalable, and provides a much better experience for both students and institute staff compared to traditional methods.

## VII. FUTURE SCOPE

The project is currently fully functional, but it can be upgraded with more features in the future to make it even better:

- ***Online Payment Gateway:*** Adding integrations like Razorpay or Stripe so students can pay their pending fees directly from their dashboard using UPI or debit cards.
- ***Automated Notifications:*** Connecting the backend to an SMS or Email API so the system automatically sends a message to students when their fee due date is near.
- ***Mobile Application:*** Using React Native to convert the current website code into a dedicated mobile app for Android and iOS devices.

.

## REFERENCES

[1] Meta Platforms Inc., "React – A JavaScript library for building user interfaces," React Official Documentation, 2026.

[2] OpenJS Foundation, "Node.js API Documentation," Node.js Official, 2026.

[3] MongoDB Inc., "MongoDB Architecture and Manual," MongoDB Documentation, 2026.

[4] Auth0 Inc., "JSON Web Tokens (JWT) Structure and Security," JWT.io, 2026.

[5] R. S. Pressman, Software Engineering: A Practitioner's Approach, 8th ed., McGraw-Hill, 2014.

[6] Tailwind Labs, "Tailwind CSS - Rapidly build modern websites," Tailwind Documentation, 2026.