Smart Classroom Management Software for Enhanced Learning Environments

[1] Mrs. D. Madhuri, [2] Papana Dwarakesh, [3] Satti Pandu, [4] Tiruveedhula Pradeep

[1]Assistant Professor, Dept of Computer Science and Engineering, St. Ann's College of Engineering and Technology, Chirala-523187, India.

[2,3,4] B. Tech Student, Dept of Computer Science and Engineering, St. Ann's College of Engineering and Technology, Chirala-523187, India.

ABSTRACT- ***The Smart Classroom Management System (SCMS) is a full-stack, role-based web application developed to digitalize and automate the academic and administrative operations of educational institutions. The system serves three distinct user roles — Administrator, Teacher, and Student — each provided with a dedicated dashboard and an access-controlled feature set. Traditional classroom management relies on fragmented tools, manual attendance registers, spreadsheets, and paper-based records, which are time-consuming, error-prone, and offer no centralized visibility for administrators. There is a need for a secure, centralized, and intelligent platform that can automate attendance, scheduling, and communication while eliminating proxy attendance and administrative overhead. The proposed system is built using Node.js and Express.js on the backend with MongoDB as the database, and a React 19 with TypeScript frontend styled using Tailwind CSS. It integrates role-based JWT authentication with an optional browser-based Face Recognition login, an intelligent auto-generation algorithm for conflict-free timetables, real-time announcements and messaging through Socket.IO, and comprehensive digital profile card management for students and teachers.***

KEYWORDS: ***Smart Classroom Management, MERN Stack, Face Recognition, Role-Based Access Control, Automated Timetabling, Real-Time Communication, Web Application.***

INTRODUCTION

Educational institutions manage large volumes of academic and administrative data, including student profiles, attendance records, examination results, fee collections, timetables, assignments, announcements, and communication between faculty and students. Traditional methods that rely on

paper records or standalone spreadsheets are inefficient, time-consuming, and prone to human error. Maintaining records across departments makes retrieval difficult and increases administrative overhead. As institutions adopt digital learning environments, there is a growing need for an integrated platform that streamlines classroom management while improving efficiency, transparency, and accessibility.

This project combines web-based classroom administration with AI-based face recognition to improve attendance accuracy and reduce proxy attendance. The Smart Classroom Management System (SCMS) is designed around three primary user roles — Administrator, Teacher, and Student — each with a dedicated dashboard offering features specific to its responsibilities. The proposed system not only digitalizes routine academic operations but also adds an additional layer of intelligence through automated scheduling and biometric attendance.

LITERATURE SURVEY

Smart classroom and campus management systems have gained significant attention in recent years due to rapid advancements in web technologies and communication frameworks. A smart classroom platform is generally defined as a digital environment that integrates data management, communication, and intelligent automation to enhance the efficiency, transparency, and quality of academic administration. The core concept revolves around centralizing academic operations and enabling real-time, role-based decision-making using data collected from students, teachers, and administrators.

RELATED WORK

Several academic and commercial projects have explored the digitalization of classroom management. Early Learning Management Systems (LMS) such as Moodle and Google Classroom provided content delivery and assignment features but lacked integrated administrative modules such as fee management, timetable generation, or comprehensive digital profile management. Research into smart attendance systems has frequently explored biometric methods, RFID cards, and QR codes, while browser-based face recognition using TensorFlow.js and face-api.js has emerged as a feasible, cost-effective approach that requires only a standard webcam without additional hardware. Constraint-based scheduling algorithms have also been studied extensively for university timetabling, ranging from simple greedy assignment

approaches to backtracking-based solvers capable of handling complex multi-resource constraints.

EXISTING SYSTEM

Existing systems used in many educational institutions suffer from critical architectural and operational limitations. Institutions typically deploy separate, disconnected tools for attendance, results, fees, and communication, creating isolated data silos and preventing centralized visibility for administrators. These generic platforms lack strict role-based control, often failing to distinguish between Admin, Teacher, and Student permissions at the API level, which compromises data security. Operational efficiency is further hindered by the absence of real-time features, since legacy spreadsheet-based setups cannot support live announcements, chat modules, or instant attendance feedback. Timetable creation remains largely manual and error-prone, and the lack of digital profile cards means that personal information and documents remain locked in physical files. Overall, existing systems are characterized by time-consuming manual processes, poor data visibility, no browser-based biometric login alternatives, and limited scalability.

PROPOSED SYSTEM

The proposed Smart Classroom Management System (SCMS) is a comprehensive, web-based platform that unifies all academic and administrative operations into a centralized, role-based application, replacing disconnected tools and paper-based record-keeping. The system delivers three distinct user portals. The Admin Portal grants full administrative control over users, classrooms, departments, timetables, fee tracking, grade registries, announcements, and digital profile cards. The Teacher Portal focuses on instructional workflows, enabling educators to manage assigned classes, log attendance, post assignments, enter results, and communicate with students. The Student Portal provides learners with access to personalized timetables, assignments, attendance history, results, fee status, and digital profile information. An optional Face

Recognition login, powered by face-api.js, enables passwordless, contactless authentication and automated attendance marking, eliminating the need for manual registers or proxy attendance.

SYSTEM ARCHITECTURE

The architecture of the SCMS is built on a three-tier client-server framework that cleanly isolates presentation, application logic, and data management. Tier 1, the frontend presentation layer, is a responsive Single Page Application (SPA) built with React 19 and TypeScript, styled using Tailwind CSS, and optimized with the Vite build tool. It communicates with the backend through REST APIs using Axios and maintains persistent real-time connections through Socket.IO, while TanStack Query manages server-state caching and synchronization. Tier 2, the application logic layer, is built on Node.js and Express.js, handling business rules, JWT-based authentication, role authorization, and a structured REST API. Security is reinforced using Helmet for secure HTTP headers, CORS restrictions, and bcryptjs for password hashing. Tier 3, the data layer, uses MongoDB accessed through the Mongoose ODM, with dedicated collections for users, classrooms, departments, attendance, results, fees, and messages. Facial images are captured through a webcam, converted into face descriptors in the browser, and compared with stored descriptors to authenticate users and mark attendance automatically.

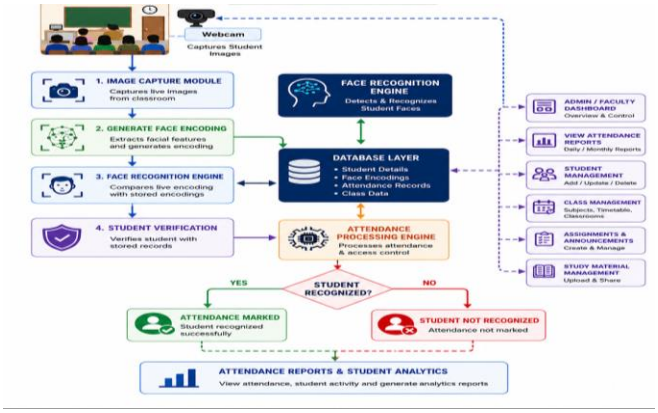

Fig 1: System Architecture

METHODOLOGY DESCRIPTION

The methodology of the SCMS is designed to provide intelligent automation, monitoring, and control of classroom activities using a modern JavaScript technology stack. The system follows a structured approach that includes role-based authentication, data acquisition through dedicated modules, backend processing via REST controllers, decision-making through validation and scheduling algorithms, and real-time actuation through Socket.IO events. Core modules include User Authentication (with optional Face ID login), Role-Based Dashboards, User and Admin Management,

Department and Subject Management, Classroom Management, Timetable Management with an auto-generation algorithm, Attendance Management, Assignment Management, Results Management, Fee Management, Announcements, Real-time Messaging, and Digital Profile Card management. Each module is scoped to the relevant classroom or user and enforces role-specific access at both the frontend routing level and the backend API level.

RESULTS AND DISCUSSION

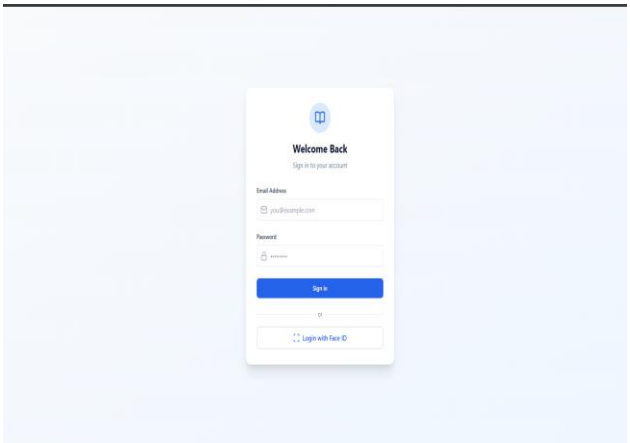

Fig 2: Login Page

The Login Page allows administrators, faculty members, and students to securely access the Smart Classroom Management Software using either credential-based login or the optional Face Recognition login.

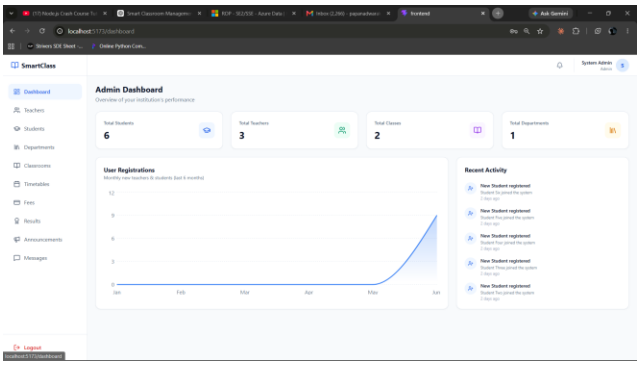

Fig 3: Admin Dashboard

The Admin Dashboard provides an overview of the institution, including student, teacher, class, and department statistics, along with recent activity summaries.

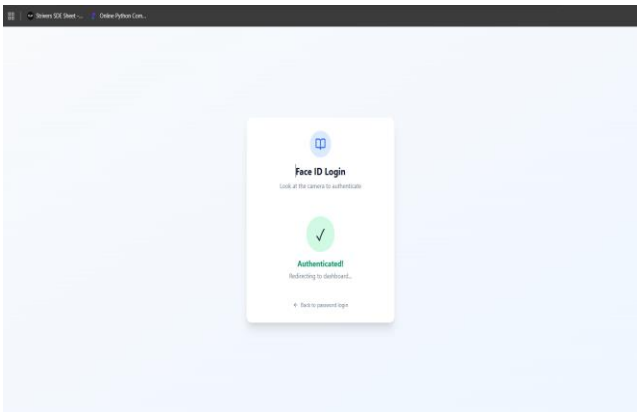

Fig 4: Face Recognition Login

The Face Recognition Login page authenticates users using browser-based facial recognition powered by face-api.js. Upon successful verification, the user is automatically redirected to the appropriate role-based dashboard and, where applicable, attendance is marked automatically.

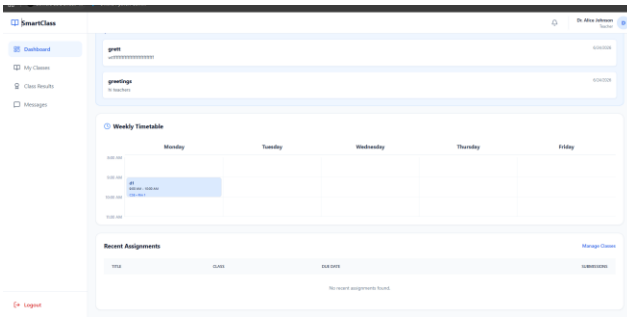

Fig 5: Teacher Dashboard

The Teacher Dashboard provides access to class schedules, assignments, announcements, and classroom activities, enabling teachers to manage academic tasks efficiently.

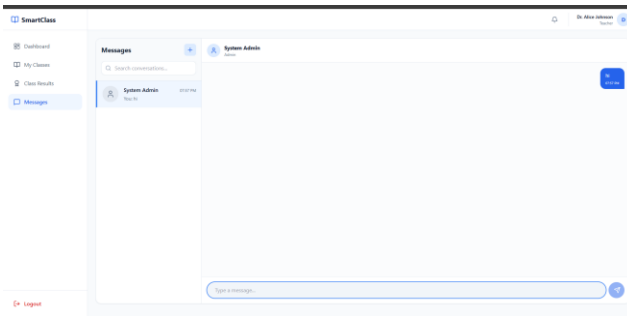

Fig 6: Real-Time Messaging Module

The Messaging Module enables real-time communication between administrators, faculty members, and students through an integrated, Socket.IO-powered chat interface.

CONCLUSION

The Smart Classroom Management System successfully addresses the operational challenges faced by educational institutions through a modern, full-stack web application. By integrating role-based access control, real-time communication, intelligent scheduling, and comprehensive profile management into a single platform, the system eliminates the fragmentation and inefficiency of traditional paper-based and multi-tool approaches. The constraint-based timetable generation algorithm enables administrators to produce conflict-free weekly schedules within seconds, while the optional Face ID login demonstrates the practical application of browser-based machine learning for passwordless, contactless attendance without requiring additional hardware.

FUTURE SCOPE

The proposed Smart Classroom Management System can be further enhanced by developing a dedicated mobile application for push notifications and on-the-go attendance access. Migrating file storage to cloud-based services and deploying the backend timetable generation as a centralized API would improve scalability across large institutions. Integrating AI-powered analytics could help identify at-risk students and generate academic performance reports, while a secure online examination module, a dedicated parent portal, QR-code-based attendance, multi-factor authentication, and

integration with standard LMS technologies such as SCORM and xAPI would further extend the platform's academic lifecycle coverage.

REFERENCES

[1] P. Harini, "Sms based wireless e-notice board," *International Journal of Emerging Technology and Advanced Engineering,* vol. 3, no. 3, pp. 181-185, 2013.

[2] P. Harini, "Enhanced packet delivery techniques using crypto-logic riddle on jamming attacks for wireless communication medium," *Int. J. Latest Trends Eng. Technol,* vol. 2, no. 4, pp. 469-478, 2013.

[3] P. Harini, "[Retracted] Machine Learning Algorithms Are Applied in Nanomaterial Properties for Nanosecurity," *Journal of Nanomaterials,* vol. 2022, no. 1, p. 5450826, 2022.

[4] MongoDB, Inc. (2024). MongoDB Manual: Document-Oriented NoSQL Database. Retrieved from https://www.mongodb.com/docs/manual/

[5] Meta Open Source. (2024). React: A JavaScript Library for Building User Interfaces (Version 19). Retrieved from https://react.dev/

[6] Microsoft Corporation. (2024). TypeScript Handbook: Static Type Checking for JavaScript. Retrieved from https://www.typescriptlang.org/docs/

[7] TanStack. (2024). TanStack Query v5: Powerful Asynchronous State Management for Web Applications. Retrieved from https://tanstack.com/query/latest

[8] Socket.IO. (2024). Socket.IO Documentation: Real-Time Bidirectional Event-Based Communication (Version 4). Retrieved from https://socket.io/docs/v4/

[9] Vincent, J. (2020). face-api.js: JavaScript API for Face Detection and Face Recognition in the Browser with TensorFlow.js (Version 0.22). GitHub Repository.

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

[11] Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.

[12] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[13] Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems (7th ed.). Pearson Education.

[14] Stallings, W. (2017). Cryptography and Network Security: Principles and Practice (7th ed.). Pearson Education.

[15] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A

literature survey. ACM Computing Surveys,

35(4), 399-458.