Project Report Submission System

Abhijit B.Kadam[1], Somnatha U. Kulkarni[2], Aniket U. Kapale[3], Shweta J.Patil[4],
[1,2,3,4,5] Department of Computer Science and Engineering,
[1,2,3,4,5]Nanasaheb Mahadik College Of Engineering, Peth, India

---***---

Abstract:
The Project Report Submission System is a comprehensive web application developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js). It aims to simplify and secure the process of submitting and managing academic project reports. The platform features distinct access levels for Admins, Teachers, and Students, ensuring role-based control and efficient workflow. With secure JWT-based user authentication, students can upload their reports in PDF format, while faculty can review, approve, and issue certificates. The system includes real-time dashboards, automated email alerts, and dynamic certificate creation using PDFKit. By digitizing the submission process, it enhances transparency, collaboration, and administrative efficiency within academic institutions.

Keywords: MERN Stack, Academic Report System, Role-Based Access Control, JWT Security, Dynamic Certificate Creation, MongoDB Atlas, Real-Time Dashboard, PDF Uploads

1.INTRODUCTION :

With the growing adoption of digital tools in education, it has become essential to modernize traditional administrative tasks to improve efficiency, security, and ease of use. One major area where many institutions still face challenges is in the handling of student project reports. Managing the submission, evaluation, approval, and certification of these reports manually often leads to delays, errors, and communication gaps. The conventional approach—where students submit printed reports, staff manually track approvals, and certificates are issued offline—is outdated and difficult to manage, especially for institutions with large numbers of students.

To overcome these challenges, the Project Report Submission System was developed as an all-in-one web-based solution that digitizes the entire workflow of academic project management. This platform enables seamless interaction between students, teachers, and administrators, making the process of report handling faster, more transparent, and far more reliable. Built using the MERN stack—which includes MongoDB, Express.js, React.js, and Node.js—the system ensures a scalable, high-performance, and flexible infrastructure.

This application uses role-based access, meaning that each user—whether a Student, Teacher, or Admin—has access to features tailored to their responsibilities. Students can upload their reports in PDF format. Teachers can log in to review, approve, or reject submissions, while Admins can monitor the entire process through real-time dashboards and manage user accounts. All uploaded documents are securely stored using MongoDB Atlas, ensuring data safety and accessibility.
Security is a core feature of the system. It uses JSON Web Tokens (JWT) to handle user authentication, ensuring that access to the platform is both safe and restricted to authorized users. Once a project report is approved, the system automatically generates a digital certificate using PDFKit, which is then emailed directly to the student.

The platform also features a live dashboard that gives Admins and Teachers real-time insights into report submissions, approvals, user activity, and more. This allows institutions to make timely decisions, reduce manual follow-ups, and improve coordination between all stakeholders.

In conclusion, the Project Report Submission System transforms the outdated, paper-based report handling method into a fully digital process. It reduces administrative workload, speeds up approvals, improves communication, and ensures that students receive timely feedback and certification. It is a step forward in aligning educational institutions with the demands of the digital age.

2. OBJECTIVES:

The main objectives of the Project Report Submission System are as follows:

To Digitize the Submission Process
Replace the traditional manual method of submitting academic project reports with a secure, web-based digital system.

To Implement Role-Based Access Control
Provide different levels of access and features for Admins, Teachers, and Students based on their responsibilities within the system.

To Ensure Secure Authentication
Use JWT (JSON Web Token) for secure user login and session management, preventing unauthorized access to the platform.

To Enable PDF Report Uploads and Management
Allow students to upload project reports in PDF format and store them securely in MongoDB Atlas for easy access and management.

To Streamline Report Review and Approval

Enable teachers to review submitted reports, approve or reject them, and track the status through an interactive dashboard.

To Automate Certificate Generation
Automatically generate project completion certificates for approved reports using PDFKit, reducing manual effort and errors.

To Send Automated Email Notifications
Notify students via email about their report approval status and deliver certificates directly to their registered email addresses.

To Provide Real-Time Analytics and Dashboards
Equip administrators and faculty with live data and reports on system usage, pending tasks, and submission statistics.

To Improve Academic Workflow and Transparency
Enhance coordination and transparency among students, faculty, and administrators by offering a unified platform for project management.

To Create a Scalable and Maintainable System
Build a flexible, modular architecture using the MERN stack that can adapt to future academic needs and institutional growth.

3. LITERATURE REVIEW:

Over the years, academic institutions have increasingly adopted digital systems to manage and streamline administrative tasks. Among these, project submission and supervision systems have been of particular interest due to the challenges posed by manual handling of final-year projects.

Bakar et al. (2011) introduced a Final Year Supervision Management System aimed at monitoring student projects within Computer Science departments. Their system emphasized real-time tracking and better communication between supervisors and students, setting the foundation for project monitoring tools in academic settings [1].

Albar (2012) proposed an electronic supervision system architecture for academic environments. His research focused on designing scalable and secure frameworks that facilitate remote project management and collaboration, which laid the groundwork for systems incorporating role-based access and secure communication channels [2].

Ademola et al. (2013) developed a university project management portal that centralized the submission, review, and feedback processes. Their system aimed to reduce administrative workload and support final-year undergraduate project tracking, which aligns with the objectives of modern web-based project submission tools [3].

Ismail et al. (2017) presented the Online Project Evaluation and Supervision System (OPENS), specifically designed to aid in the proposal phase of final year projects. Their system allowed for online evaluations, progress tracking, and seamless student-supervisor interaction, highlighting the importance of an integrated workflow in project supervision [4].

From a technological perspective, the evolution of web development tools and programming languages has contributed significantly to building efficient systems. Resources such as Wikipedia provide foundational information on core technologies used in this project:

Databases play a central role in storing structured data for users, reports, and submission records [5].

PHP, although widely used in earlier systems, has been increasingly replaced by JavaScript-based stacks for more interactive and dynamic web applications [6].

MySQL, a traditional relational database, has been complemented or replaced by NoSQL databases like MongoDB in modern applications for better scalability and flexibility [7].

JavaScript remains a core language for frontend and backend development, especially with the rise of frameworks like React.js and Node.js [8].

CSS (Cascading Style Sheets) continues to be essential for designing responsive and visually appealing user interfaces [9].

HTML (HyperText Markup Language) provides the structural foundation of all web-based systems [10].

While earlier research focused on monolithic or PHP-MySQL-based applications, the current trend moves toward modern full-stack JavaScript frameworks. The Project Report Submission System builds upon this trend by using the MERN stack (MongoDB, Express.js, React.js, Node.js), offering a more interactive and modular design. It incorporates features inspired by previous systems—such as role-based access, progress monitoring, and supervisor feedback—while adding advanced capabilities like JWT authentication, PDF certificate generation, email automation, and real-time dashboards.

In conclusion, the reviewed literature and technological resources support the need for a comprehensive, web-based project submission and supervision platform. By integrating best practices from prior systems and leveraging modern technologies, this project offers a robust solution to the limitations of traditional project handling methods.

4. METHODOLOGY:

The development of the Project Report Submission System followed a structured and practical approach, inspired by the Agile methodology to ensure adaptability, modular development, and timely updates. The process began with understanding the current challenges in manual report submission workflows commonly used in academic institutions. Detailed observations and informal discussions with students, teachers, and administrative staff helped identify the core requirements such as secure authentication, role-based access, report uploads, approval workflows, certificate generation, and email communication.

Once the requirements were clearly defined, the system architecture was designed using the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js.The database schema was structured to manage users, roles, reports, and certificates efficiently. Frontend wireframes and backend API structures were prepared to guide the development process.

During the implementation phase, the project was divided into functional modules. Authentication was handled using JSON Web Tokens (JWT), ensuring that only verified users could access protected features. The system defined specific roles—Student, Teacher, and Admin—each with tailored access. Students were given the ability to upload project reports in PDF format. These reports were stored securely using MongoDB Atlas along with relevant metadata like project title, submission date, and the student's name. Teachers were given access to view, review, and either approve or reject the submissions. Once a report was approved, a project completion certificate was automatically generated using PDFKit and sent to the student's registered email address via Nodemailer.

After development, the system was rigorously tested to ensure reliability and functionality. Unit testing was used to verify individual components, while integration testing confirmed smooth communication between the frontend and backend. Real user scenarios were simulated to ensure the system worked as expected from a user perspective**.** Following successful testing, the application was deployed using Vercel for the frontend and Render for the backend, with the database hosted on MongoDB Atlas. All sensitive information, such as API keys and email credentials, was secured using environment variables.

Regular updates are planned to include features such as OTP verification, detailed analytics, downloadable reports, and institutional integrations. This methodology ensured that the final system is not only functional and user-friendly but also scalable and secure for long-term academic use.

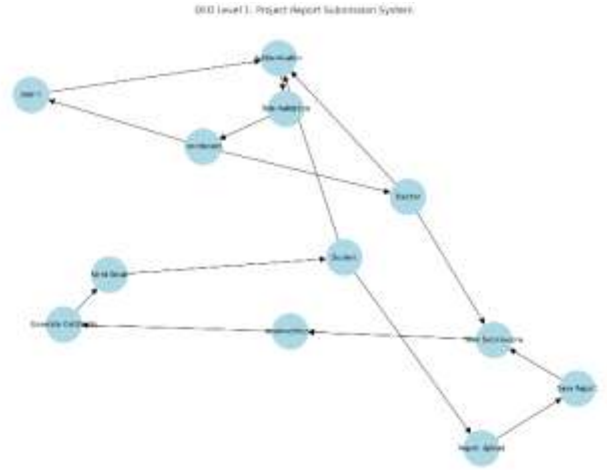


Fig -1: Data flow dig

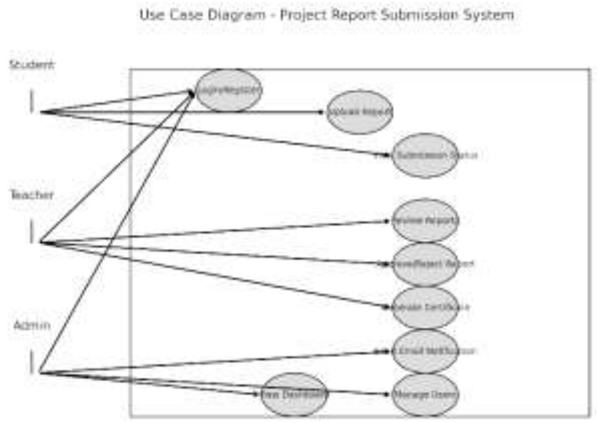
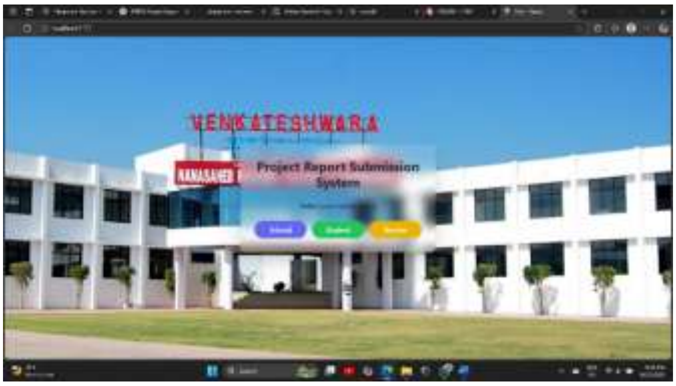
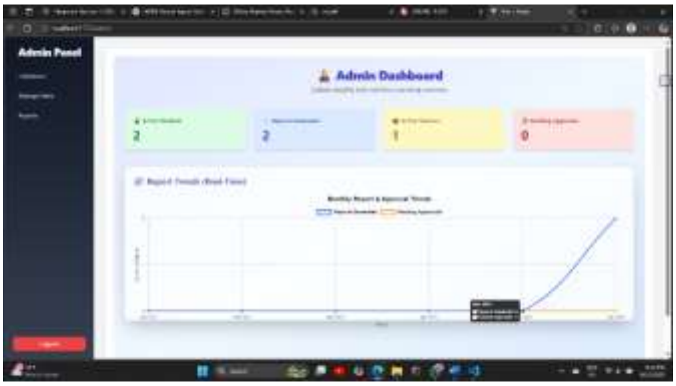


Fig -2: Use case dig

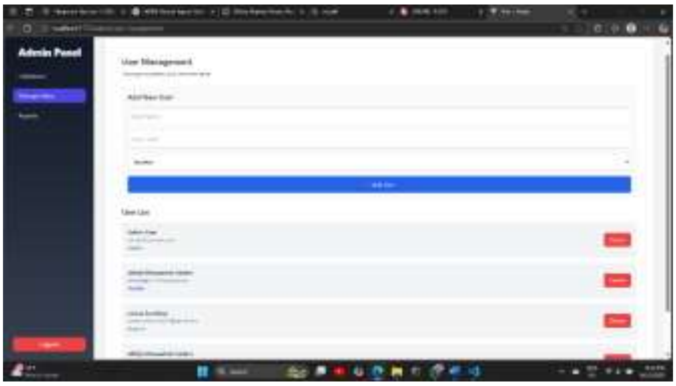
5. RESULTS:



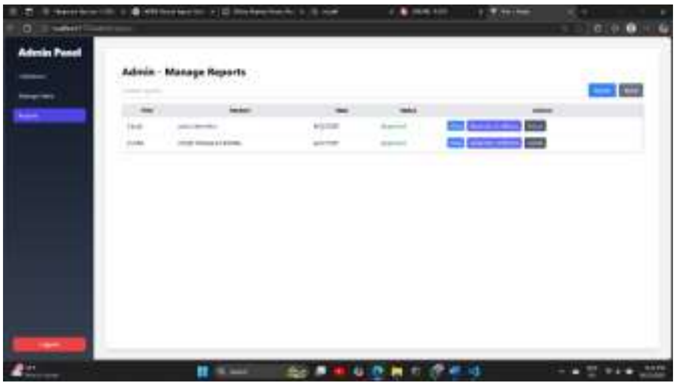
1.1 Login Page



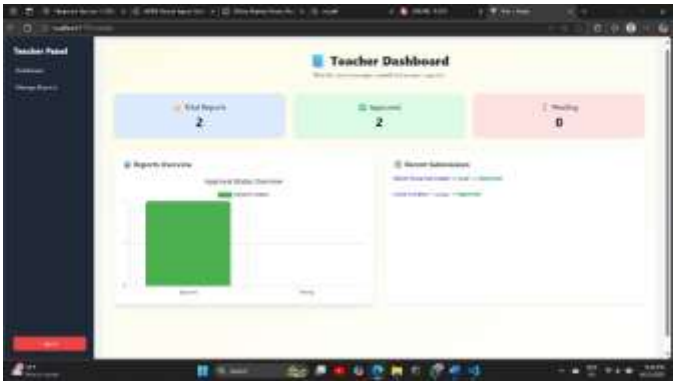
1.2 Admin Dashboard Page



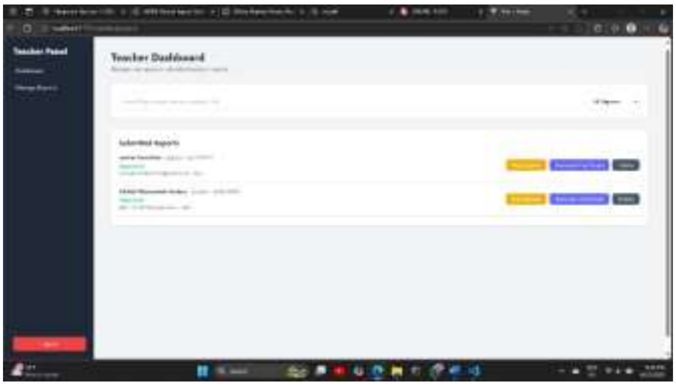
1.3 User Management Page



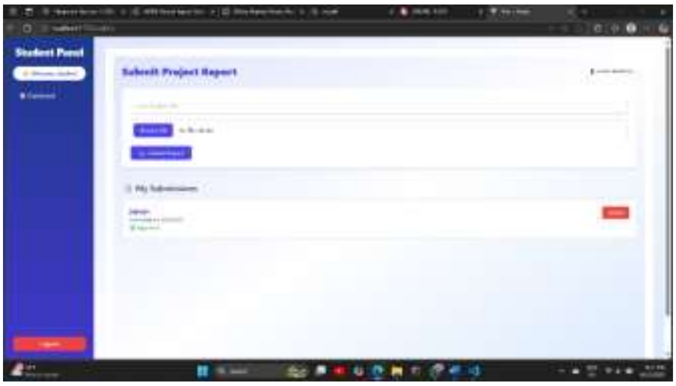
1.4 Admin’s Report Management Page



1.5 Teacher Dashboard Page



1.6 Teacher’s Report Management Page



1.6 Student Dashboard Page

6. CONCLUSIONS:

The Project Report Submission System was designed and implemented to address the inefficiencies and limitations of traditional, manual report submission processes in academic institutions. By leveraging modern web technologies through the MERN stack—MongoDB, Express.js, React.js, and Node.js—the system provides a scalable, secure, and user-friendly platform that simplifies project management for students, teachers, and administrators alike.

The system successfully automates key operations including report uploads, status tracking, approvals, and certificate generation. It ensures secure access using JWT-based authentication and enhances communication through automated email notifications. Role-based access control allows each user to perform specific functions based on their responsibilities, maintaining both organization and data privacy.

With the integration of real-time dashboards, PDFK it for dynamic certificate generation, and cloud-based data storage using MongoDB Atlas, the platform not only improves transparency and workflow but also reduces the administrative burden on faculty and staff. Furthermore, the system is built to be extendable, making it adaptable to future requirements such as plagiarism detection, downloadable analytics, and integration with institutional portals.

In conclusion, the Project Report Submission System offers a comprehensive, reliable, and modern solution to the challenges faced in managing academic project reports. It enhances efficiency, ensures data security, and fosters better collaboration within educational environments, making it a valuable asset for academic institutions aiming to embrace digital transformation.

ACKNOWLEDGEMENT:

I would like to express my sincere gratitude to all those who supported and guided me throughout the development of the Project Report Submission System.

First and foremost, I would like to thank my project guide and faculty members for their valuable insights, constant encouragement, and expert guidance. Their constructive feedback at every stage of the project helped me improve and complete this work successfully.

I am also thankful to my institution for providing the necessary resources and an environment conducive to learning and innovation. The support from the academic and administrative staff made the process smooth and organized.

A special thanks to my classmates and friends who offered their suggestions, participated in testing, and provided moral support during the project development phase.

This project would not have been possible without the combined efforts and support of everyone involved. I am truly grateful to all.

REFERENCES:

Bakar, M. A., Jailani, N., Shukur, Z., & Yatim, N. F. M. (2011). A system was proposed for supervising and tracking Computer Science final-year projects, focusing on improving the quality of project monitoring within academic institutions. Procedia - Social and Behavioral Sciences, 18, 273–281.

Albar, A. M. (2012). Introduced a conceptual framework for electronic supervision systems in educational settings, aimed at enhancing collaboration between supervisors and students. European Journal of Business and Management, 4(8), 140–148.

Ademola, A., Adewale, A., & Ike, D. U. (2013). Designed a university portal to manage undergraduate final-year projects digitally, providing centralized control and access for students and faculty. International Journal of Engineering and Computer Science, 2(10).

Ismail, S. I., Abdullah, R., Kar, S. A. C., Fadzal, N., Husni, H., & Omar, H. M. (2017, December). Developed an online tool (OPENS) to assist with the creation and supervision of final-year project proposals**,** enhancing the early stages of project work. Presented at the 2017 IEEE 15th Student Conference on Research and Development (SCOReD), 210–214. IEEE.

Wikipedia contributors. (n.d.). Overview of databases as structured collections used for data storage, retrieval, and management. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Database

Wikipedia contributors. (n.d.). General information about PHP as a server-side scripting language used primarily for web development. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/PHP

Wikipedia contributors. (n.d.). Introduction to MySQL, a relational database system widely used in web applications for data handling. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/MySQL

Wikipedia contributors. (n.d.). Description of JavaScript as a client-side and server-side programming language for dynamic web development. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/JavaScript

Wikipedia contributors. (n.d.). Explanation of Cascading Style Sheets (CSS) used for styling web documents. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Cascading_Style_Sheets

Wikipedia contributors. (n.d.). Overview of HTML as the standard markup language used to structure content on the web. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/HTML