iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** III **Month of publication:** March 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.78949

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue III Mar 2026- Available at www.ijraset.com*

# Smart Student Productivity & Task Management System

ThotiGiri[1], Talari Naveen[2], Talari Nithin[3], Tallapalli Yeswanth Kumar[4], Tamisetty Pallavi[5], Mr. George Sebatian[6]
*[1, 2, 3, 4, 5]CSE (Data Science), [6]Assistant Professor, Sri Venkateswara College of Engineering and Technology (SVCET), Chittoor*

***Abstract:*** ***In modern educational environments, students often struggle to manage multiple academic tasks, assignments, and deadlines effectively. The " Smart Student Productivity & Task Management System (StudiIn) " is designed to provide a centralized platform that helps students organize tasks, track progress, and improve productivity while enabling administrators to assign tasks and monitor student performance efficiently. The system is developed using the MERN Stack (MongoDB, Express.js, React.js, Node.js) and includes features such as task management, progress tracking, productivity analytics dashboards, and role-based access control.***

***The platform also integrates an AI-powered chatbot assistant to guide users and enhance the overall user experience. In addition, the system provides a secure password recovery mechanism using OTP-based email verification implemented through the Resend API. By combining task management, analytics, and intelligent assistance, the proposed system improves productivity and provides an effective solution for academic task monitoring.***

## I. INTRODUCTION

With the increasing complexity of academic activities, students often face challenges in organizing their tasks, assignments, and deadlines. Managing multiple academic responsibilities such as coursework, projects, and assessments requires proper planning and monitoring. Traditional task management methods such as manual notes, to-do lists, or simple reminders are often insufficient for managing multiple academic responsibilities effectively.

In addition, educational institutions and administrators require efficient tools to track student task completion and monitor productivity levels. Without a centralized system, monitoring student progress becomes difficult and time-consuming, which can lead to missed deadlines and reduced productivity among students. Moreover, the absence of proper tracking systems makes it difficult for students to evaluate their performance and identify areas where improvement is required.

To address these challenges, the Smart Student Productivity & Task Management System is developed as a web-based platform that enables students and administrators to efficiently manage academic tasks and monitor progress. The system includes features such as task creation and management, task progress tracking, productivity analytics dashboards, role-based access control, AI-powered chatbot guidance, and secure password recovery using OTP-based email verification. These features provide a structured environment that improves task organization and helps users manage their academic activities more efficiently.

The system also allows administrators to assign tasks to students, monitor task completion, and analyze student performance through analytical dashboards. Additionally, the platform improves communication and coordination between students and administrators by providing a centralized interface for task management. The integration of data visualization, task tracking, and intelligent assistance enhances the overall learning management experience and helps students maintain better productivity and organization in their academic activities.

## II. LITERATURE REVIEW

There have been several studies and research works related to task management systems, productivity tracking platforms, and learning management systems designed to improve student performance and academic organization. With the rapid development of web technologies and cloud-based platforms, many researchers have focused on developing digital tools that help students manage academic activities, track task progress, and improve productivity. Various studies highlight the importance of centralized platforms that allow users to organize tasks, monitor progress, and analyze productivity trends.

### *A. Task Management Systems*

Many existing task management systems provide users with the ability to create, organize, and track tasks efficiently. Applications such as Trello, Asana, and Todoist have demonstrated the importance of digital task tracking tools in improving productivity.

©IJRASET: All Rights are Reserved 

These systems allow users to set deadlines, organize tasks, and monitor task completion. However, most of these tools are designed for general productivity and are not specifically tailored for academic environments where students need to manage assignments, coursework, and project deadlines.

### *B. Student Productivity Monitoring Systems*

Several studies have emphasized the importance of productivity monitoring in educational environments. Productivity monitoring systems help students track their work progress, identify pending tasks, and maintain better academic discipline. Research shows that when students can visualize their productivity using dashboards and reports, they are more likely to complete tasks on time and improve their performance.

### *C. Web-Based Learning and Task Management Platforms*

With the growth of web-based technologies, many educational institutions have adopted online platforms for managing academic activities. Learning Management Systems (LMS) such as Moodle and Google Classroom provide platforms where instructors can assign tasks and monitor student progress. However, these systems primarily focus on course delivery and assignment submission rather than personal productivity management and detailed task tracking.

### *D. Analytics and Data Visualization in Productivity Systems*

Recent studies highlight the importance of data visualization in productivity and task management systems. Visualization tools such as charts, graphs, and dashboards help users understand task distribution, productivity trends, and completion rates. Interactive dashboards allow users to analyze their progress and make better decisions regarding time management and task prioritization.

### *E. AI-Based Chatbot Assistance*

Artificial Intelligence-based chatbot systems have been widely used in modern web applications to assist users and improve user interaction. Chatbots can provide automated responses to user queries, guide users through system functionalities, and reduce the need for manual assistance. Research studies show that chatbot assistants enhance user engagement and simplify system navigation.

### *F. Secure Authentication and Password Recovery Systems*

Security is an essential aspect of modern web applications. Many studies emphasize the importance of secure authentication systems and password recovery mechanisms. OTP-based email verification systems are widely used to ensure secure password reset functionality. These systems help users recover their accounts securely while protecting sensitive data from unauthorized access.

### *G. Research Gap Identification*

Although many task management and productivity systems exist, most focus only on individual functionalities such as task tracking or analytics. There is a lack of integrated platforms that combine student task management, productivity analytics, administrative monitoring, chatbot assistance, and secure authentication. The proposed Smart Student Productivity & Task Management System (StudiIn) addresses this gap by providing a comprehensive web-based platform that improves student productivity and academic task management.

## III. PROPOSED METHODOLOGY

The proposed methodology presents a structured approach for developing a web-based system that helps students manage academic tasks and improve productivity. The Smart Student Productivity and Task Management System is designed as a fullstack application using the MERN stack. The system integrates several components including user authentication, task creation, progress tracking, analytics dashboards, chatbot assistance, and secure password recovery. The proposed system provides an end-to-end solution that allows students to organize their academic tasks efficiently while enabling administrators to monitor student performance through analytical dashboards.

### *A. System Architecture and Platform Design*

The proposed system follows a full-stack architecture consisting of a frontend interface, backend server, application logic, and database layer. The frontend of the system is developed using React.js, which provides an interactive and responsive user interface for students and administrators.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The backend is implemented using Node.js and Express.js to handle API requests, manage system logic, and process user operations such as authentication and task management. MongoDB is used as the database to store user information, task details, and progress data. This layered architecture ensures smooth communication between the frontend and backend through REST APIs and provides scalability, flexibility, and maintainability for the overall system.

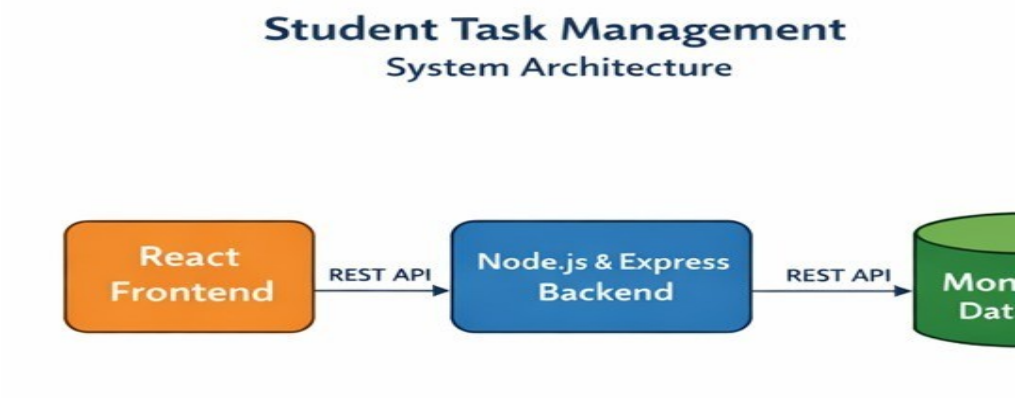



Fig: System Architecture of the proposed system

*B. User Authentication and Account Management*

The first stage of the system involves user authentication and account management. Users are able to register and log in to the system using their credentials. The system implements secure authentication using JSON Web Tokens to maintain user sessions and protect sensitive information. In addition, the system provides a password recovery mechanism that allows users to reset their passwords in case they forget them. This mechanism uses OTP-based email verification implemented through the Resend API. When a user requests a password reset, a verification code is sent to the registered email address, and after successful verification, the user is allowed to update the password securely.

*C. Task Creation and Task Management*

Once users are authenticated, students can create and manage their academic tasks through the task management module. Each task contains important details such as task title, task description, start date, and deadline. The system supports both personal tasks created by students and tasks assigned by administrators. All task information is stored in the MongoDB database and can be accessed whenever required. This module enables students to organize their assignments, projects, and study-related activities in a structured manner, thereby improving academic organization.

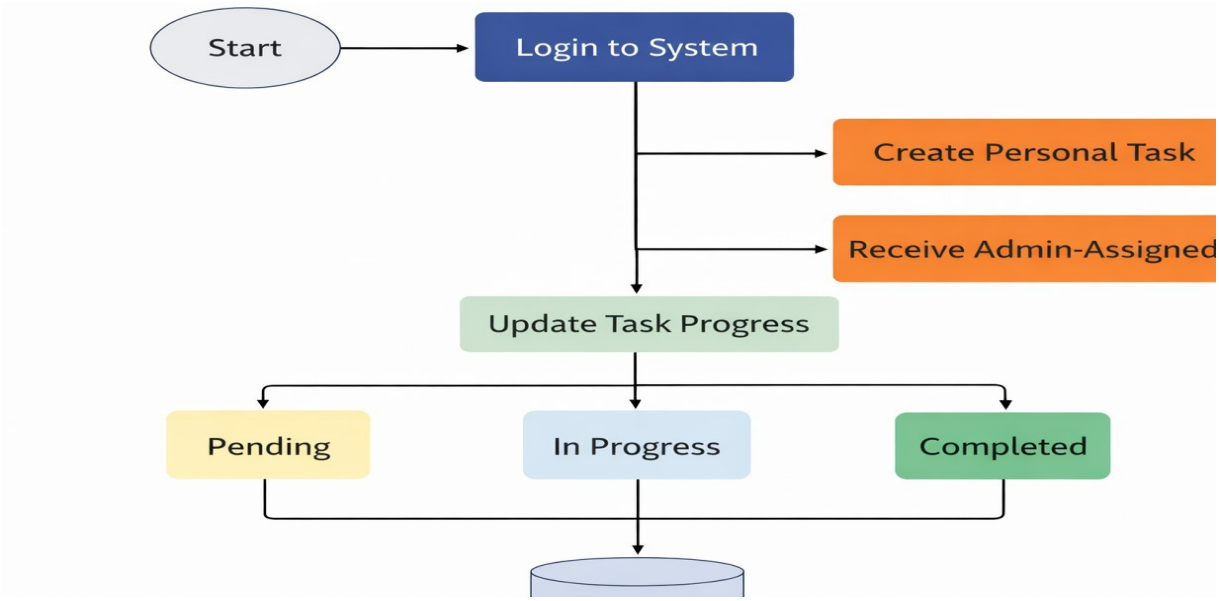



Fig: Task management workflow diagram

*D. Task Progress Tracking*

The system provides functionality to monitor and update the progress of each task. Students can update the completion status of their tasks as they progress with their work. Tasks are categorized into different states such as pending, in progress, and completed. The system continuously updates the progress information in the database and reflects the changes in the user dashboard. This feature allows both students and administrators to track the completion status of tasks effectively and ensures that academic activities are completed within the specified deadlines.

*E. Productivity Analytics and Dashboard Visualization*

The proposed system includes an analytics dashboard that helps users analyze their productivity and task completion performance. The dashboard displays important information such as the total number of tasks, completed tasks, pending tasks, and tasks currently in progress. These statistics are represented through graphical visualizations such as charts and progress indicators. The use of data visualization helps users understand productivity patterns and identify areas that require improvement. Administrators can also analyze student performance using these dashboards to monitor academic progress.

*F. AI Chatbot Assistance*

The system integrates an AI-powered chatbot assistant that helps users interact with the platform more efficiently. The chatbot provides guidance on system features and helps users understand how to perform various operations such as creating tasks, updating progress, and viewing analytics. The chatbot improves the user experience by providing quick responses to common queries and assisting users in navigating the system without requiring external help.

Summary

The proposed methodology provides a comprehensive framework for implementing the Smart Student Productivity and Task Management System. The system integrates task management, productivity monitoring, analytics visualization, chatbot assistance, and secure authentication within a single platform. By utilizing modern web technologies such as React.js, Node.js, Express.js, and MongoDB, the system offers a scalable and interactive environment for managing academic tasks. The platform improves student productivity by providing organized task management tools while also enabling administrators to monitor and evaluate student performance effectively.

## IV. RESULTS AND DISCUSSION

*A. System Testing and Implementation Results*

The Smart Student Productivity and Task Management System was successfully implemented using the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. The system was tested using multiple user accounts representing both student and administrator roles. Various functionalities such as task creation, task assignment, progress tracking, analytics dashboards, chatbot assistance, and password recovery were tested to evaluate the performance of the system. The testing results confirm that the system operates efficiently and provides a reliable platform for managing academic tasks.

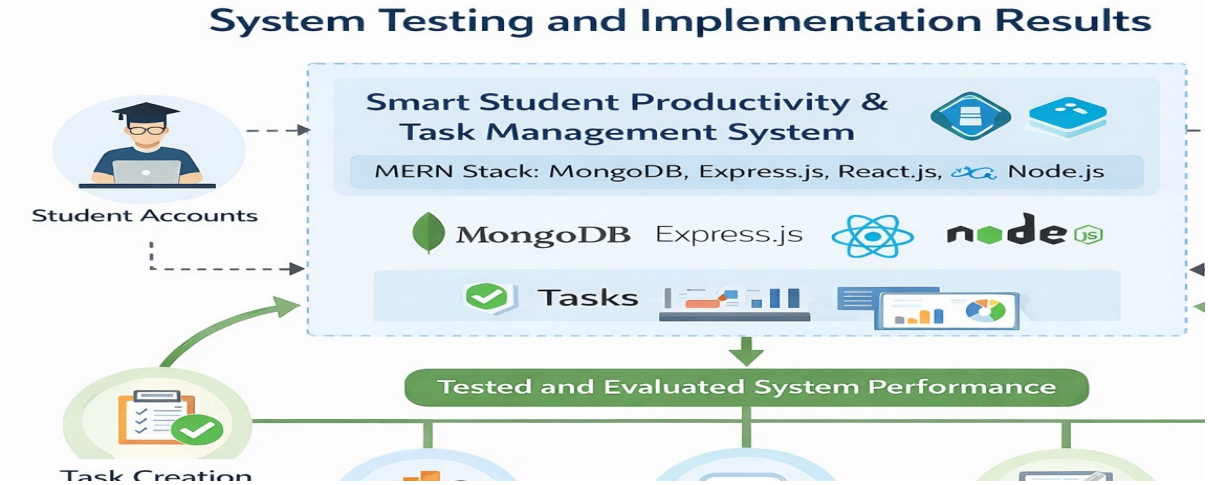



Fig: System testing flowchart for smart student

### B. *Task Management System Results*

The implemented system allows students to create personal tasks and administrators to assign tasks to students. The testing results show that tasks can be successfully created, updated, and stored in the MongoDB database. The system organizes tasks efficiently and displays them in the task dashboard, allowing users to easily manage their academic activities.

### C. *Task Progress Monitoring Results*

The task progress tracking feature allows students to update the status of tasks based on their completion level. Tasks are categorized into three main states: Pending, In Progress, and Completed. During system testing, the task status was updated correctly and reflected immediately in the dashboard. This feature helps students monitor unfinished tasks and ensures better task completion management.

### D. *Analytics Dashboard Results*

The analytics dashboard provides visual insights into task performance and productivity trends. The system generates summary statistics such as total tasks, pending tasks, in-progress tasks, and completed tasks. These results are represented using graphical charts and visual indicators, helping users understand productivity patterns and task completion performance.

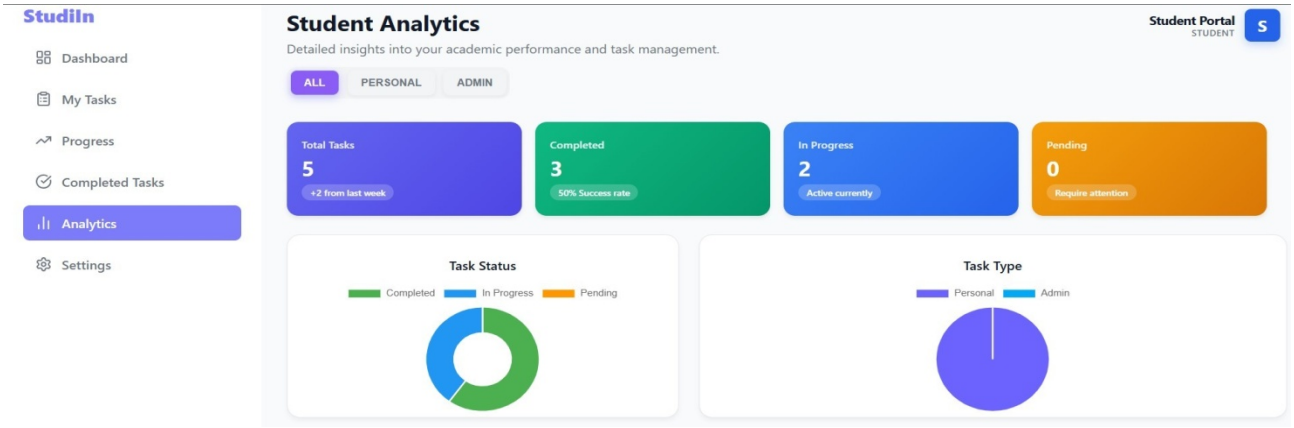



Fig: Analytics Dashboard Showing Task Performance and Productivity Trends

### E. *Chatbot Assistance Results*

The system includes an AI-powered chatbot that assists users in navigating the platform and understanding different features of the system. During testing, the chatbot successfully provided responses to user queries related to task creation, progress updates, and dashboard features. The chatbot improves user interaction and provides quick guidance when users need help.

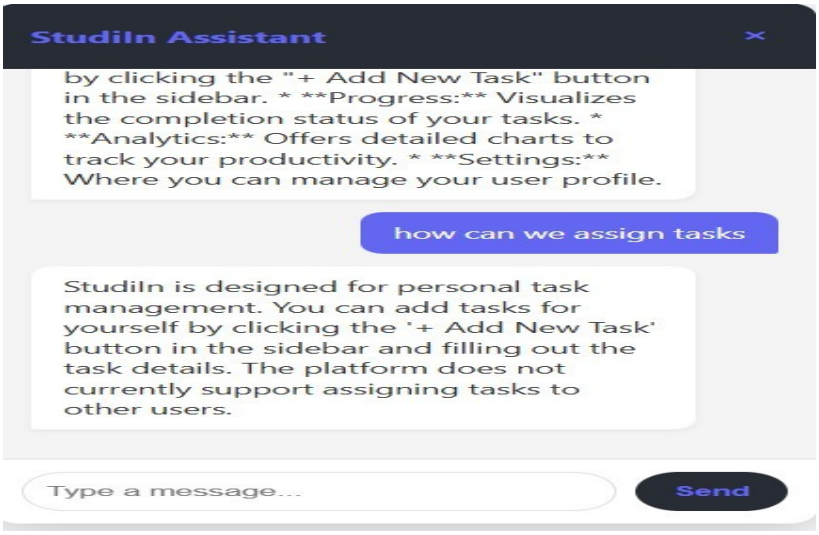



Fig: AI Chatbot Interaction in the Proposed System

*F. Security and Password Recovery Results*

The system implements secure authentication using JSON Web Tokens to protect user sessions. In addition, the password recovery mechanism uses OTP-based email verification through the Resend API. During testing, verification codes were successfully delivered to registered email addresses, allowing users to securely reset their passwords.

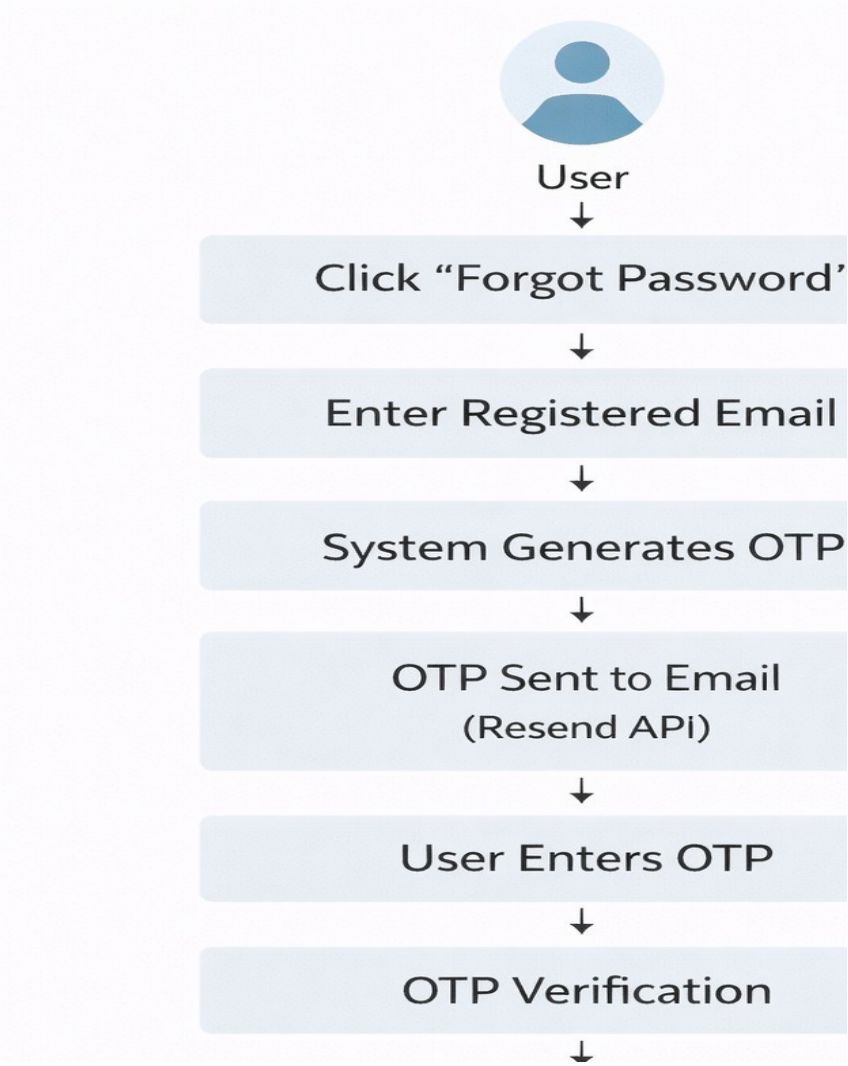



Fig: OTP-based password recovery flowchart

*G. Discussion and Observations*

The experimental results demonstrate that the proposed system successfully provides an efficient platform for managing academic tasks and monitoring productivity. The integration of task management, analytics dashboards, chatbot assistance, and secure authentication improves the usability and effectiveness of the system. The platform helps students organize academic responsibilities while allowing administrators to monitor student performance through analytical insights.

## V. CONCLUSIONS

This paper presents the Smart Student Productivity and Task Management System (StudiIn), a web-based platform designed to help students manage academic tasks efficiently and improve productivity. The proposed system integrates important functionalities such as task creation, task assignment, progress tracking, analytics dashboards, and chatbot assistance within a single platform. The system is developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js), providing a
scalable and interactive solution for managing academic activities.

The platform allows students to organize personal tasks, update task progress, and monitor productivity using visual analytics dashboards. Administrators can assign tasks to students and monitor their completion status, which helps in tracking student performance effectively. The integration of an AI-powered chatbot further enhances user experience by guiding users in navigating the platform and understanding different system features.

In addition, the system implements secure authentication and password recovery using OTP-based email verification through the Resend API, ensuring reliable and secure access for users. Experimental testing results demonstrate that the proposed system improves task organization and productivity monitoring. Future enhancements may include mobile application support, intelligent task recommendation systems, and real-time notifications to further improve the functionality of the platform.

## VI. ACKNOWLEDGMENT

I would like to express my sincere gratitude to everyone who supported and guided us in completing the project titled “Smart Student Productivity and Task Management System (StudiIn)”. First, we would like to thank our college management for providing the necessary infrastructure and resources required for the successful completion of this project.

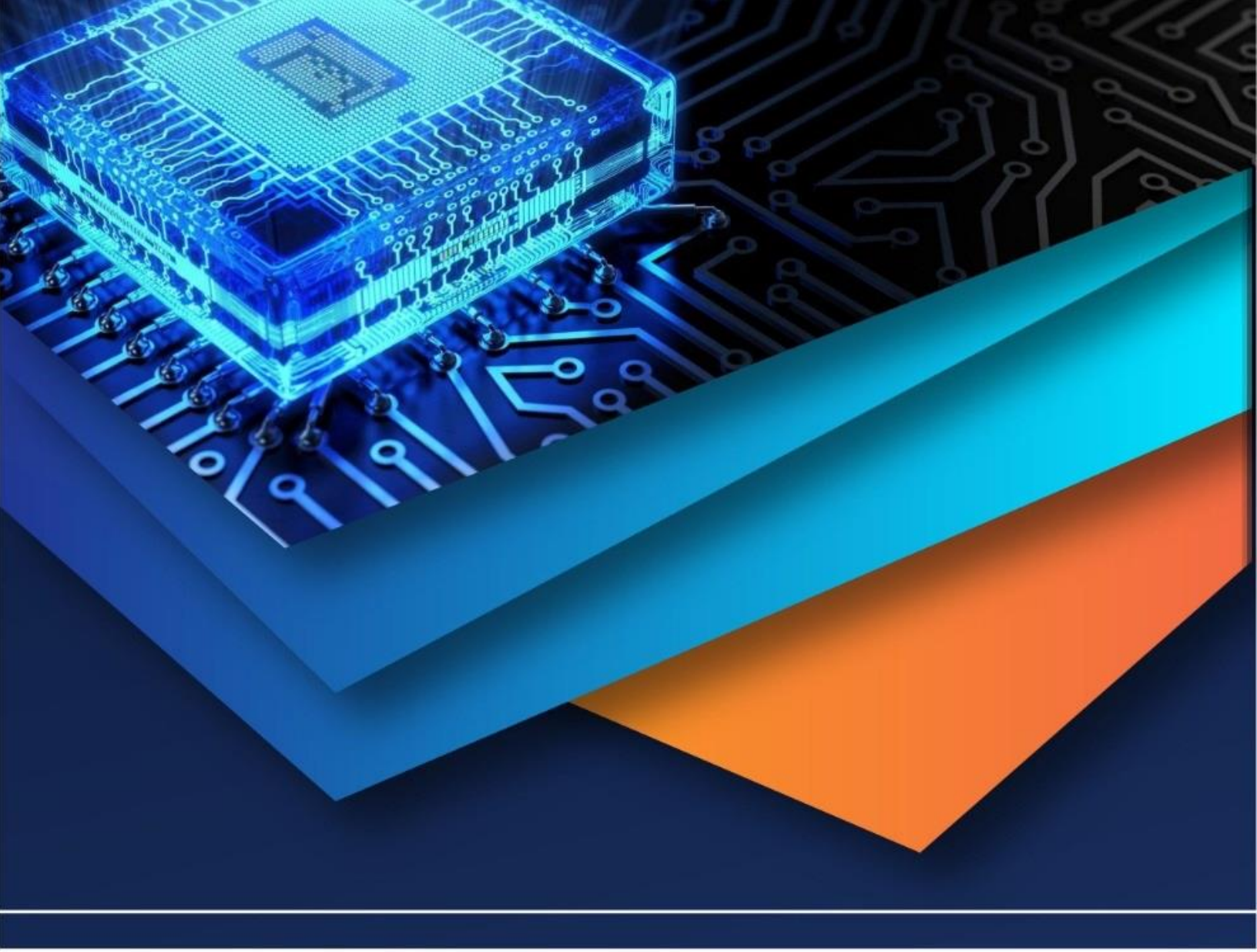
We are especially grateful to our project guide and faculty members for their valuable guidance, continuous encouragement, and constructive suggestions throughout the development of the project. Their support helped us in understanding the technical concepts and successfully implementing the system. We also thank the Department of Computer Science and Engineering for their cooperation and encouragement during the project work.

Finally, we would like to thank our friends and family members for their constant motivation and support. This project provided us with valuable practical knowledge in full-stack web development using the MERN stack, task management systems, and web application development, and we are thankful to everyone who directly or indirectly contributed to the successful completion of this project.

## REFERENCES

[1] M. Abadi et al., “React: A JavaScript Library for Building User Interfaces,” Facebook Open Source Documentation, 2023. Describes the React.js framework used for developing dynamic and responsive web user interfaces.
[2] R. Fielding, “Architectural Styles and the Design of Network-Based Software Architectures,” Doctoral Dissertation, University of California, Irvine, 2000. Introduces REST architectural principles used for communication between frontend and backend systems.
[3] D. Flanagan, JavaScript: The Definitive Guide, 7th ed., O’Reilly Media, 2020. Provides comprehensive concepts of JavaScript used for developing interactive web applications.
[4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994. Explains software design principles applicable to scalable system architecture.
[5] S. Tilkov and S. Vinoski, “Node.js: Using JavaScript to Build High-Performance Network Programs,” IEEE Internet Computing, vol. 14, no. 6, pp. 80–83, 2010. Discusses Node.js architecture for building scalable server-side applications.
[6] A. Holowaychuk, Express.js Guide, StrongLoop Inc., 2018. Explains the Express.js framework used for building RESTful APIs and backend services.
[7] K. Chodorow, MongoDB: The Definitive Guide, 3rd ed., O’Reilly Media, 2019. Provides detailed concepts of MongoDB NoSQL database used for storing application data.
[8] T. Bray, “JSON Web Token (JWT) Authentication,” Internet Engineering Task Force (IETF), RFC 7519, 2015. Describes JWT-based authentication used for secure user session management.
[9] D. Crockford, JavaScript: The Good Parts, O’Reilly Media, 2008. Presents best practices for JavaScript programming used in modern web applications.
[10] Chart.js Documentation, “Simple yet Flexible JavaScript Charting Library,” Chart.js Open Source Project, 2023. Describes the charting library used for implementing analytics dashboards and graphical data visualization.
[11] Resend Inc., “Resend Email API Documentation,” Resend Developer Platform, 2024. Provides documentation for sending OTP-based verification emails used for password recovery.
[12] J. Sommerville, Software Engineering, 10th ed., Pearson Education, 2016. Discusses system design, software architecture, and development methodologies.
[13] M. Richards and N. Ford, Fundamentals of Software Architecture, O’Reilly Media, 2020. Explains architectural patterns used in modern web application development.
[14] S. Newman, Building Microservices, O’Reilly Media, 2015. Discusses scalable backend design approaches relevant to modern web platforms.
[15] M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, 2002. Presents design principles for developing maintainable enterprise web applications.
[16] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed., Pearson Education, 2021. Explains AI concepts that support chatbot-based assistance systems.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)