PROJECT REPORT

PROJECT MANAGEMENT APP

Submitted in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE ENGINEERING WITH SPECIALIZATION IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

By
Ashlin James Chakkalakkal

23BAI70722

AIT - Computer Science and Engineering
University Institute of Engineering
Chandigarh University, Gharuan

DECLARATION

This is to declare that this report has been written by me. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I aver that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

ASHLIN JAMES CHAKKALAKKAL

23BAI70722

ABSTRACT

In today's fast-paced digital work environment, efficient task organization and real-time project tracking have become essential for individuals and teams. Despite the availability of various productivity tools, many users struggle to find systems that are simple, visually appealing, and tailored to personalized workflow management. This Project Management System was developed to address this challenge by providing an intuitive, aesthetically designed platform for managing tasks across different stages of progress.

The significance of this project lies in its ability to bridge the gap between usability and functionality by integrating a clean user interface with real-time task updates and secure authentication. The novelty of this system is reflected in its streamlined design, responsive layout, and the use of modern web technologies such as React, Node.js, MongoDB, and Socket.io. The application ensures that users can manage tasks smoothly while experiencing an interface that prioritizes minimalism, clarity, and accessibility.

The development approach adopted focused on creating a hands-on, interactive, and user-centered application. Key functionalities include user authentication through JWT, real-time task synchronization using WebSockets, and persistent storage of tasks in MongoDB. Tasks are categorized into three sections—Pending, Ongoing, and Completed—allowing users to visually track progress and manage workloads effectively. Additional features such as task timers and smooth UI interactions enhance the practicality and usability of the system.

Through this project, users gain a structured and visually appealing workspace that simplifies task tracking and improves productivity. The major outcomes include the ability to add, edit, move, and monitor tasks in real time, ensuring seamless task management across sessions. The project also emphasizes scalability, modularity, and clean UI design, which are essential components for modern web-based productivity tools.

The Project Management System successfully demonstrates how modern full-stack development techniques can be combined with thoughtful design principles to create a highly functional and aesthetically pleasing application. It provides a solid foundation for future enhancements such as team collaboration, analytics dashboards, reminders, and mobile app integration. This project highlights the potential of integrating UI design, cloud-ready architecture, and real-time systems to support efficient and organized workflow management.

Keywords: Project Management, Task Tracking, React, Node.js, MongoDB, WebSockets, Real-time Systems, JWT Authentication, UI Design.

NOMENCLATURE

ABBREVIATION	DESCRIPTION
API	Application Programming Interface; used for communication between frontend and backend.
CRUD	Create, Read, Update, Delete – core operations performed on tasks and user data.
CSS	Cascading Style Sheets; used for designing and styling the user interface.
DOM	Document Object Model; structure used by React to manage UI components.
JWT	JSON Web Token; used for secure user authentication and authorization.
JSX	JavaScript XML; the syntax extension used in React components.
MongoDB	A NoSQL database used to store user and task details.
MERN Stack	MongoDB, Express.js, React.js, Node.js – the full-stack technologies used in this project.
Node.js	JavaScript runtime used to build the backend server.
Express.js	Backend framework used to create RESTful APIs.
NPM	Node Package Manager; used to install dependencies.
React.js	Frontend JavaScript library used to build the user interface.
Socket.io	Library enabling real-time communication between client and server.
UI	User Interface; visual layout and design of the application.
UX	User Experience; overall interaction experience of the user with the system.
HTTP	Hypertext Transfer Protocol; used for client-server communication.
REST API	Representational State Transfer API; defines endpoints for backend operations.
JSON	JavaScript Object Notation; lightweight data format used to exchange information.
Task Board	The visual panel in the app used to display tasks under different statuses.
Pending / Ongoing / Completed	The three workflow stages used to categorize tasks.
Timer	Feature used to measure time spent on each task.
WebSocket	Protocol enabling bi-directional real-time communication.

INDEX

PROJECT TITLE

Project Management App

REFERENCE WEBSITE LINK

1. Trello – Project Management Tool: https://trello.com/
2. React Documentation: https://react.dev/
3. Node.js Documentation: https://nodejs.org/en/docs
4. Express.js Documentation: https://expressjs.com/
5. MongoDB Documentation: https://www.mongodb.com/docs/
6. Socket.io Documentation: https://socket.io/docs/v4/

PROJECT DESCRIPTION

The Project Management System is a full-stack web application designed to help users organize, track, and manage their tasks efficiently through an intuitive and visually appealing interface. Inspired by modern productivity platforms such as Trello, the system provides a structured Kanban-style workflow consisting of three task stages: Pending, Ongoing, and Completed. Each user has a personalized dashboard where they can create, update, move, and monitor tasks in real time.

The system is built using the MERN stack: MongoDB, Express.js, React.js, and Node.js, ensuring a scalable, responsive, and interactive user experience. The frontend is developed with React, offering a clean, modern UI enhanced by a custom-designed color palette and typography for improved readability and usability. The backend is implemented using Node.js and Express.js, with JWT-based authentication to ensure secure login and user access control.

A notable feature of the application is its real-time task synchronization using Socket.io, allowing updates to reflect instantly across user sessions. Each task also includes a built-in timer functionality, enabling users to track the time spent on individual tasks, contributing to improved productivity and work analysis. MongoDB is used for reliable storage of user accounts and tasks, supporting fast retrieval and robust data management.

The system emphasizes simplicity, efficiency, and user-centered design. It provides an organized space where users can focus on their work, improve their workflow, and monitor progress effortlessly. This Project Management System demonstrates the integration of modern web technologies to create a functional, aesthetically refined, and productivity-driven application suitable for personal and professional use.

PROBLEM STATEMENT

In modern work and academic environments, managing multiple tasks and tracking their progress has become increasingly challenging. Existing project management tools are often overly complex, require paid subscriptions, or lack features tailored for individual productivity needs. Many users struggle to organize their tasks effectively due to the absence of a simple, intuitive, and visually appealing system that supports real-time updates and seamless workflow management.

The primary problem is the lack of an accessible and user-friendly platform where individuals can categorize their tasks, monitor progress across different stages, and efficiently manage their workload. Additionally, there is a need for secure user authentication and persistent task storage to ensure that users can maintain their personal task boards across sessions.

To address these issues, a streamlined Project Management System is required, one that provides essential task management features such as task creation, categorization, time tracking, real-time updates, and a clean, organized interface that enhances productivity without unnecessary complexity.

HIGH LEVEL DESIGN

The High-Level Design of the Project Management System outlines the system architecture, major components, data flow, and interactions between the user interface, backend server, and database. The application is developed using the MERN stack, ensuring a scalable, modular, and maintainable architecture.

1. System Architecture Overview

The system follows a **three-tier architecture**, consisting of:

a. Presentation Layer (Frontend – React.js)

- Provides the user interface for login, registration, and task management.
- Displays task categories (Pending, Ongoing, Completed) using a clean, responsive UI.
- Manages timer events and user interactions.
- Communicates with backend through REST APIs and WebSockets.

b. Application Layer (Backend – Node.js & Express.js)

- Handles all business logic and API processing.
- Manages user authentication using JWT tokens.
- Contains APIs for creating, retrieving, updating, and deleting tasks.
- Manages real-time communication between clients using Socket.io.

c. Data Layer (Database – MongoDB)

- Stores user details (name, email, password hash).
- Stores task details (title, status, timestamps, timer value, userId).
- Ensures fast queries and flexible schema using a document-based structure.

2. Major System Components

a. Authentication Module

- Provides secure user login and registration.
- Uses JWT for authorization and protected API routes.
- Ensures each user has an isolated task board.

b. Task Management Module

- Allows users to create, edit, update status, delete, and track tasks.
- Categorizes tasks into three workflow stages.
- Provides timer functionality for tracking task duration.

c. Real-Time Update Module (Socket.io)

- Sends instant task updates across sessions.
- Ensures the task board reflects changes without page refresh.
- Enhances user experience by making interaction seamless.

d. User Interface Module

- Includes an aesthetically designed dashboard.
- Uses responsive layouts for column-wise task display.
- Implements smooth animations and clean typography for better usability.

3. Data Flow Description

1. The user logs in or registers through the frontend.
2. Credentials are sent to the backend, which verifies them and returns a JWT token.
3. The frontend stores this token and uses it for all authenticated requests.

4. When the user performs a task action (add, update, move, delete):

    - The frontend sends an API request to the backend.
    - The backend updates the MongoDB database.
    - A Socket.io event is triggered to broadcast the update.
    - The frontend receives the real-time update and refreshes the UI accordingly.

LAYER	TECHNOLOGY USED
Frontend	React.js, HTML, CSS
Backend	Node.js, Express.js
Database	MongoDB
Authentication	JWT
Real-time updates	Socket.io

FUTURE SCOPE

- Team Collaboration: Add shared project boards where multiple users can work together on tasks.
- Role-Based Access: Introduce user roles such as Admin, Manager, and Member for better access control.
- Advanced Task Options: Include subtasks, file attachments, comments, reminders, and priority levels.
- Analytics Dashboard: Provide visual charts showing productivity, task progress, and time spent.
- Calendar Integration: Sync deadlines and reminders with external calendars like Google Calendar.
- Mobile Application: Develop an Android/iOS app for easier access and real-time updates.