

TASK-MANAGEMENT SYSTEM

Jyotiprakash Pal, 4th Year, Department of CSE, Gandhi Institute for Technology, BPUT, India

jyoti2021@gift.edu.in

Lalit Kumar Mohanty, 4th Year, Department of CSE, Gandhi Institute for Technology, BPUT, India

Er. Jagannath Ray, Assistant Professor, Department of CSE, Gandhi Institute for Technology, BPUT, India

Abstract-

In today's fast-paced world, efficient task management is critical for personal and professional success. The Task Management System developed as part of this project aims to provide users with a simple yet powerful platform to create, organize, and monitor their tasks effectively. Built using the MERN stack — MongoDB, Express.js, React.js, and Node.js — the system ensures seamless user experience, secure authentication, and real-time task updates.

The platform allows users to register, log in securely, and manage their tasks through an intuitive dashboard. Users can create tasks, assign due dates, prioritize activities, and track the status of tasks (To-Do, In-Progress, Completed). Admin users are given special privileges to monitor and manage all users and system-wide tasks, ensuring effective governance.

Keywords:

MERN STACK

I. INTRODUCTION

In today's fast-paced digital world, managing tasks efficiently is more important than ever for individuals, teams, and organizations. A Task Management System (TMS) facilitates the organization, tracking, and completion of tasks, ensuring deadlines are met, and collaboration is enhanced. This project, "Task Management System," aims to provide users with a robust and intuitive platform to manage their daily tasks. Utilizing the MERN stack—MongoDB, Express.js, React.js, and Node.js—the system offers an end-to-end JavaScript solution, enabling real-time, responsive, and scalable task management.

II. LITERATURE REVIEW

Existing Systems and Studies: Many existing task management applications like Trello, Asana, and Monday.com provide comprehensive solutions for task tracking and team collaboration. However, these solutions often involve costs, lack customizability, or have features not suited for every user.

Studies show that customized task management systems can significantly improve efficiency by aligning directly with user workflows and organizational needs. Moreover, implementing these systems using modern web technologies ensures better performance and scalability.

III. SYSTEM DESIGN:

System Design plays a crucial role in the successful implementation of the Task Management System. It includes various design models such as Data Flow Diagrams (DFD), Entity-Relationship(ER)diagrams,andsystemarchitecture.

The system is divided into multiple modules that interact with each other seamlessly. Detailed diagrams will be used to represent the interaction and data flow among components.

System design focuses on outlining the structure, components, modules, and interactions within the Task Management System. A good design ensures that the system is efficient, secure, scalable, and easy to maintain. This section describes how different layers and components collaborate to deliver the desired functionalities.

IV. IMPLEMENTATION:

The backend of the Task Management System is developed using Node.js and Express.js.

API Endpoints:

- POST/api/register: Register a new user.
- POST/api/login: Authenticate and login user.
- GET/api/tasks: Retrieve all tasks for a user.
- POST/api/tasks: Create a new task.
- PUT/api/tasks/:id: Update a task's details.
- DELETE/api/tasks/:id: Delete a task.

Authentication:

- JSON Web Tokens (JWT) are used for secure user authentication.
- Passwords are hashed using bcrypt before storage.

Error Handling:

- Express middleware is used for centralized error handling.
- Proper HTTP status codes and messages are returned for errors.

The backend of the Task Management System is developed using Node.js with the Express.js framework. It provides a robust, scalable, and efficient environment for handling API requests, managing business logic, authenticating users, and connecting securely with the MongoDB database. The backend is designed using modular architecture to ensure ease of maintenance, testing, and future upgrades.

The frontend is developed using React.js, offering a dynamic and responsive user interface.

Components:

- LoginComponent: Handles user login forms.
- Register Component: Manages user signup forms.
- Task Dashboard Component: Displays user tasks.
- Create Task Component: Form to create new tasks.
- Update Task Component: Edit task details.

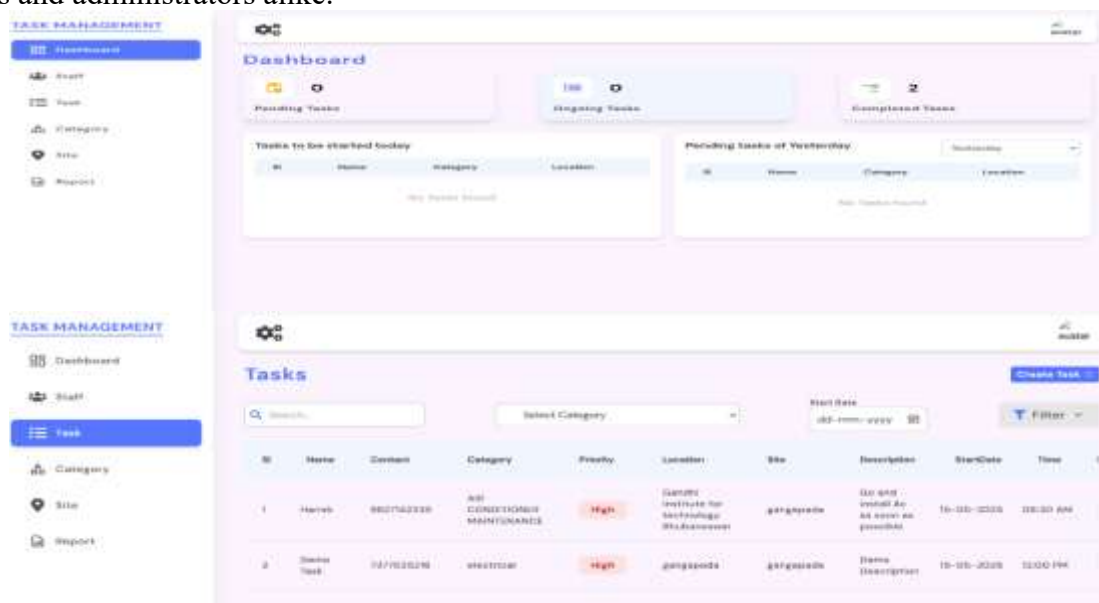
State Management:

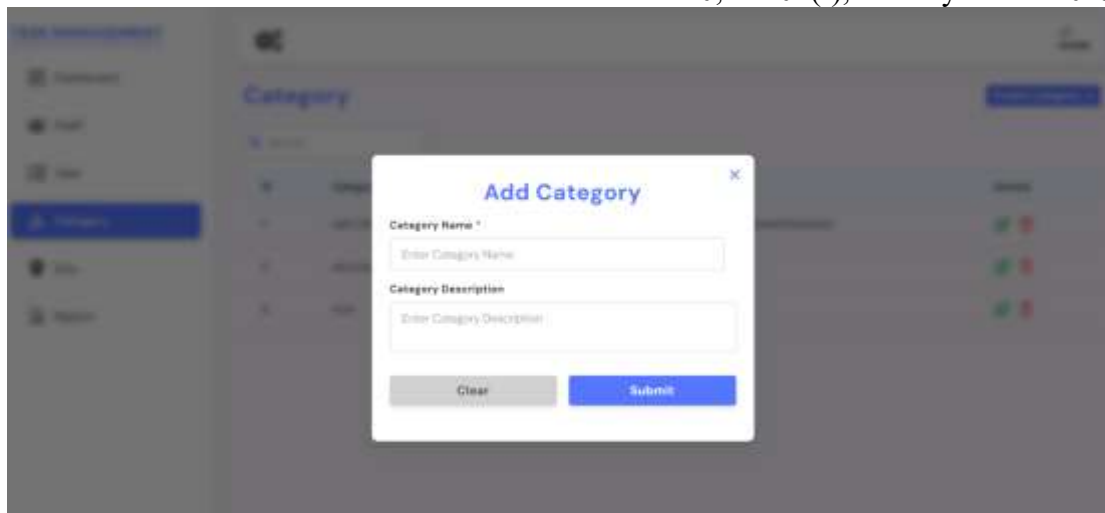
- React hooks (useState, useEffect) are used for managing component states.
- Context API is used for authentication context.

Routing:

- React Router is used for navigating between pages (login, dashboard, tasks, etc.).
- Protected routes are implemented to prevent unauthorized access.

The frontend of the Task Management System is built using React.js, a powerful JavaScript library known for its efficient rendering and dynamic user interfaces. The design philosophy of the frontend focuses on modularity, responsiveness, security, and an intuitive user experience to meet the demands of users and administrators alike.





V. RESULT

The Task Management System successfully meets the initial project objectives by providing users with a platform to efficiently manage their tasks.

Key achievements include:

- Secure user authentication system using JWT.
- Efficient task creation, assignment, and status tracking.
- Responsive frontend accessible from multiple devices.
- Scalable backend architecture using Node.js and Express.js.
- Real-time updates on the dashboard upon task creation and updates.

Performance testing showed that API response times remained below 300ms for standard operations. User testing revealed a high level of satisfaction with the system's usability and interface design.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my project guide and faculty members of the CSE department at GIFT Bhubaneswar for their valuable support and guidance throughout this project.

I extend my heartfelt thanks to our Head of Department for providing the resources and encouragement necessary for the successful completion of this Task Management System.

This project has helped me enhance my knowledge in web development and project management.

I deeply appreciate the learning environment provided by GIFT Bhubaneswar.

I acknowledge the importance of teamwork, planning, and timely execution gained through this project.

REFERENCE

- <https://www.mongodb.com/what-is-mongodb>
- <https://expressjs.com/>
- <https://reactjs.org/>
- <https://nodejs.org/>
- <https://jwt.io/introduction>
- <https://www.npmjs.com/package/bcrypt>
- <https://restfulapi.net/>
- <https://www.postman.com/>
- <https://tailwindcss.com/>
- <https://mui.com/>