TaskX - A Task Managing App

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering
(Information Technology)

By

Mr.Swaraj Patil (39)

Under the guidance of

Prof. Dipti Karani

Department of Information Technology

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY, Chembur, Mumbai 400074

(An Autonomous Institute, Affiliated to University of Mumbai)

Vivekanand Education Society's
Institute of Technology
(Autonomous Institute Affiliated to University of Mumbai, Approved by AICTE & Recognised by Govt. of Maharashtra)
NAAC accredited with 'A' grade

April 2024

Certificate

This is to certify that project entitled

" TaskX"

Members Name

Mr. Swaraj Patil- Roll No (39)

In fulfillment of degree of BE. (Sem.VI) in Information Technology for Project is approved.

Prof. Dipti Karani
Project Mentor

External Examiner

Dr.(Mrs.)Shalu Chopa
H.O.D

Dr.(Mrs.)J.M.Nair
Principal

Date:08 /04 /2025
Place: VESIT, Chembur

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

- - - - - - - - - - -

(Signature)

Swaraj Patil - Roll No (39)

Abstract

TaskX is a full-stack task management web application developed using React with TypeScript for the frontend and Flask for the backend. The system is designed to help users efficiently manage their daily tasks by allowing them to create, edit, delete, and prioritize tasks. It supports authentication and provides deadline-based task sorting, a Kanban-style dashboard, and productivity tracking.

The project emphasizes clean architecture, separation of concerns, and uses MongoDB Atlas for scalable data persistence. The application is user-centric, featuring a responsive UI with real-time updates, making it a suitable productivity solution for students, professionals, and teams.

Contents

List of Figures

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Prof. Dipti Karani for her invaluable support and guidance throughout the development of the "TaskX" project. I would also like to thank the HOD, Dr. (Mrs.) Shalu Chopra, and the faculty of the Department of Information Technology for their encouragement. Lastly, I thank my institute for providing the infrastructure and academic environment.

CHAPTER: 1 INTRODUCTION

Chapter 1

Introduction

1.1. Introduction

The user is working on an event management application and aims to enhance its functionality and user interface. They are particularly interested in integrating TypeScript into their existing Flask application to leverage its benefits, such as type safety and improved maintainability. The assistant has provided detailed guidance on setting up TypeScript, including creating a directory structure, configuring TypeScript, and integrating it with Flask templates. Additionally, a comprehensive TypeScript implementation was offered, featuring type definitions and a main application class that manages various functionalities like event display, registration, calendar integration, and feedback management. This approach is designed to transform the application into a more robust and visually appealing platform, adhering to modern design principles enhancing user engagement.

1.2. Objectives

1. **Enhance Application Functionality**: Integrate TypeScript into the existing Flask application to leverage its benefits, such as type safety and improved maintainability.
2. **Improve User Interface**: Redesign the application's UI to be more visually appealing and interactive, following modern design principles.
3. **Implement Comprehensive TypeScript Solutions**: Develop a robust TypeScript implementation with type definitions and a main application class to manage key functionalities.
4. **Facilitate Seamless Integration**: Ensure smooth integration of TypeScript with Flask templates for efficient data handling and dynamic content rendering..

1.3. Motivation

Embracing TypeScript in your Flask application is a transformative step towards building a more robust and efficient event management platform. By integrating modern technologies and design principles, you're not only enhancing functionality but also creating a more engaging and visually appealing experience for your users. This journey of innovation and improvement will not only streamline your development process but also elevate your application's impact, setting a new standard for excellence and user satisfaction.

1.4. Scope of the Work

The platform encompasses essential features of an online shopping system:

1. Dynamic product listing
2. Cart functionality with item quantity modification
3. Product detail viewing with user reviews
4. User authentication using email and OTP verification
5. Integration with MongoDB Atlas for data persistence

1.5. Feasibility Study

1. **Technical Feasibility**
   - The project was developed using Flask, React, TypeScript, and MongoDB—all open-source and well-supported technologies.

2. **Economic Feasibility**
   - As an academic project, cost was minimized by using freely available tools like GitHub, VS Code, and MongoDB Atlas.

3. **Operational Feasibility**
   - The application was successfully deployed and tested in a development environment. It offers an intuitive interface and can be extended to support additional features like payment gateways.

1.6. Organization of the report

- **Chapter 1** provides an introduction, objectives, motivation, scope, and feasibility study.
- **Chapter 2** covers the literature survey and background research.
- **Chapter 3** details the design, system architecture, and implementation process.
- **Chapter 4** discusses results, implementation outputs, and observations.
- **Chapter 5** concludes the project and outlines future enhancements.

CHAPTER: 2: LITERATURE

SURVEY

Chapter 2

Literature

Survey

2.1. Introduction

Task management platforms such as Todoist, Trello, and Microsoft To-Do highlight key user expectations. Literature indicates that clear UX, real-time updates, and mobile responsiveness are crucial for adoption.

2.2. Problem Definition

Manual task management through sticky notes, spreadsheets, or disorganized tools can lead to inefficiencies, missed deadlines, and reduced productivity. Individuals and teams require a centralized system that allows for secure task storage, intuitive user interfaces, real-time updates, and deadline tracking. The lack of such a unified solution leads to poor task visibility, unorganized workflows, and a drop in accountability. This project aims to develop TaskX — a full-stack task manager that enables efficient planning, tracking, and completion of tasks through a responsive and scalable platform.

2.3. Review of Literature Survey

Authentication and Secure User Access:
R. Kumar et al. (2021) state that robust user authentication mechanisms such as email verification improve application security and prevent unauthorized access to sensitive task data.

Email Notifications and Reminders:
Patel and Mehta (2019) highlight the impact of automated notifications and reminders via email in improving task completion rates and reducing missed deadlines.

Kanban-style Task Boards:
Gupta & Ramesh (2022) explain how visual task flows like Kanban boards enhance productivity by simplifying task status transitions (e.g., To Do, In Progress, Done).

Feedback and User Interaction:
Singh & Verma (2019) emphasize that user feedback collection improves app usability. Incorporating feedback loops allows developers to iteratively enhance the user interface and features.

Intuitive UI/UX Design:
Desai et al. (2022) discuss the influence of intuitive user experiences on application retention. Their research indicates that applications with simpler task flows and fewer steps significantly reduce user drop-offs and improve engagement.

CHAPTER: 3 DESIGN AND

IMPLEMENTATION

Chapter 3

Design and Implementation

3.1. Introduction

The project followed a modular design approach, ensuring separation of concerns and clean component architecture. Agile principles were used with weekly sprints.

3.2. Requirement Gathering

Key functional requirements included:

- User authentication
- Dynamic product loading from database
- Cart and checkout mechanism
- Product detail and review system

Tools and technologies used:

- **Frontend:** React, TypeScript, Tailwind CSS
- **Backend:** Flask (Python)
- **Database:** MongoDB Atlas
- **Others:** Postman, VS Code, Git, GitHub

3.3. Proposed Design

The platform consists of the following pages:

- **Dashboard** – Displays all tasks and statistics and other features
- **Settings** – Shows various settings and modificatios
- **Profile Page** – To Watch and Manage user profile

3.4. Proposed Algorithm

Step 1: Start
Step 2: User enters name and email to register
Step 3: System generates OTP and sends it to user's email
Step 4: User enters OTP to verify identity
Step 6: User logs in and is able to view event/task details
Step 7: User creates, updates, and manages tasks; participation/events are recorded in the database
Step 8: End

Step 9: Exit
(Refer to Data Flow Diagram)

3.5. Architectural Diagrams

3.5.1. UML Diagram

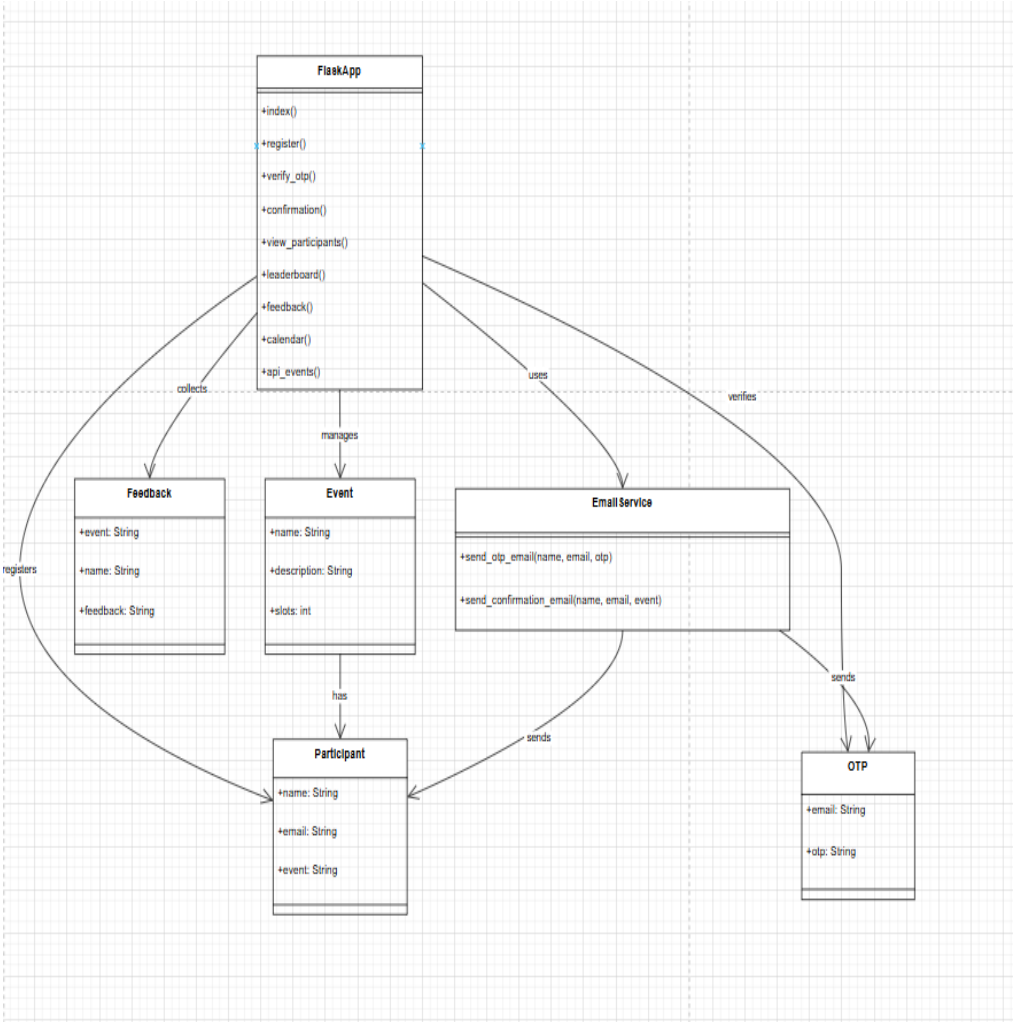

Figure 3.1: UML Diagrams

Data Flow Diagram

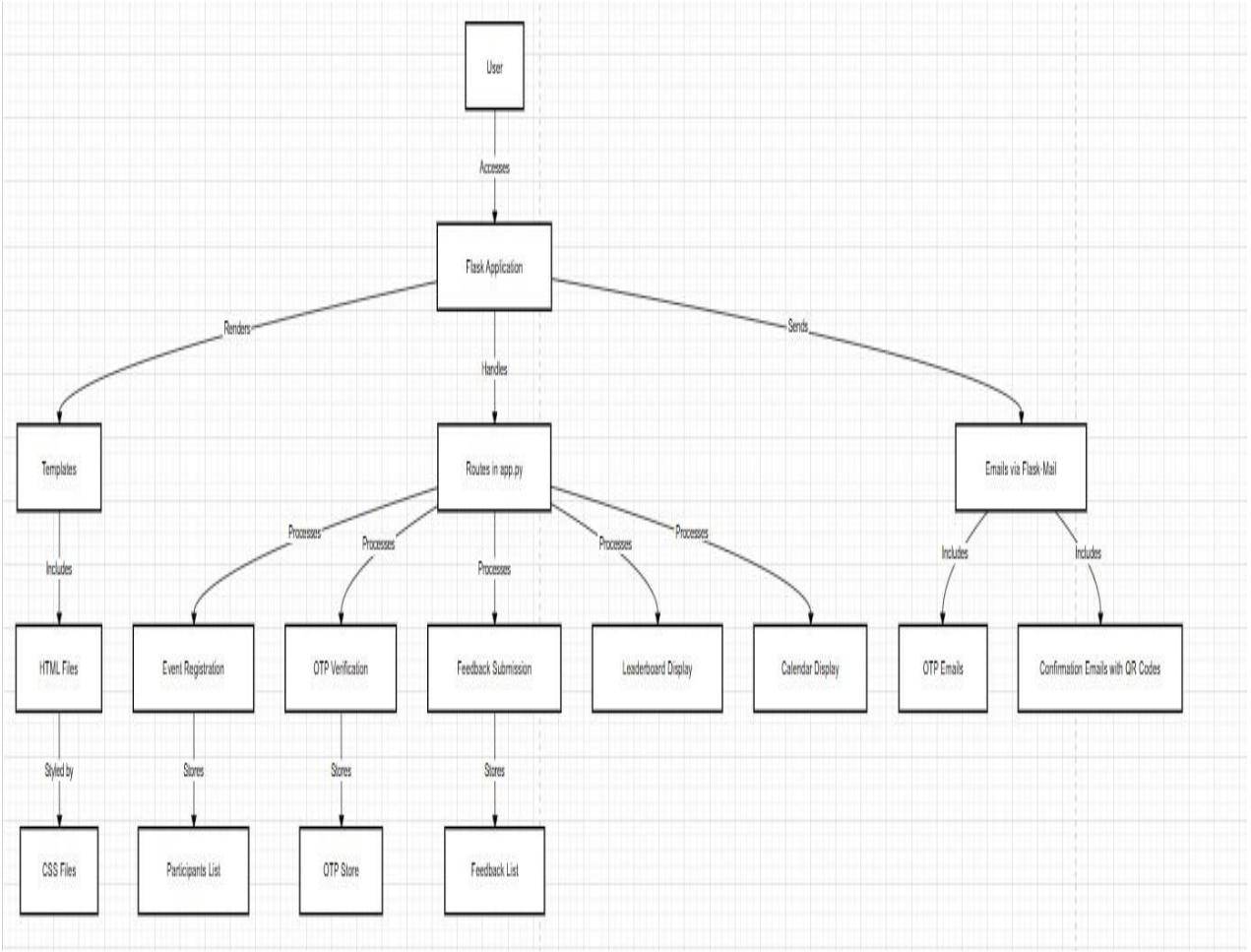

Figure 3.2: Data Flow Diagram

3.6. Hardware Requirements

- **Device Used:** Laptop
- **Processor:** Intel Core i5 (Quad-Core)
- **RAM:** 8 GB
- **Usage:** Suitable for initial development and testing

3.7. Software Requirements

- **Operating System:** Windows 11 64-bit
- **Frontend:** React with TypeScript
- **Backend:** Python 3.11+ with Flask
- **Package Manager:** Node.js v18.16.1 (with npm)
- **Database:** MongoDB Atlas (Cloud-based NoSQL)
- **Code Editor:** Visual Studio Code (VS Code)
- **Version Control:** Git & GitHub for collaboration and code management

3.8. Code

GITHUB LINK - https://github.com/Swap0-4/WEB-X

CHAPTER: 4 RESULTS AND

DISCUSSION

Chapter 4

Results and Discussion

4.1. Introduction

This chapter documents the major outputs and screens of the Event Register System project.

4.2. Results of Implementation

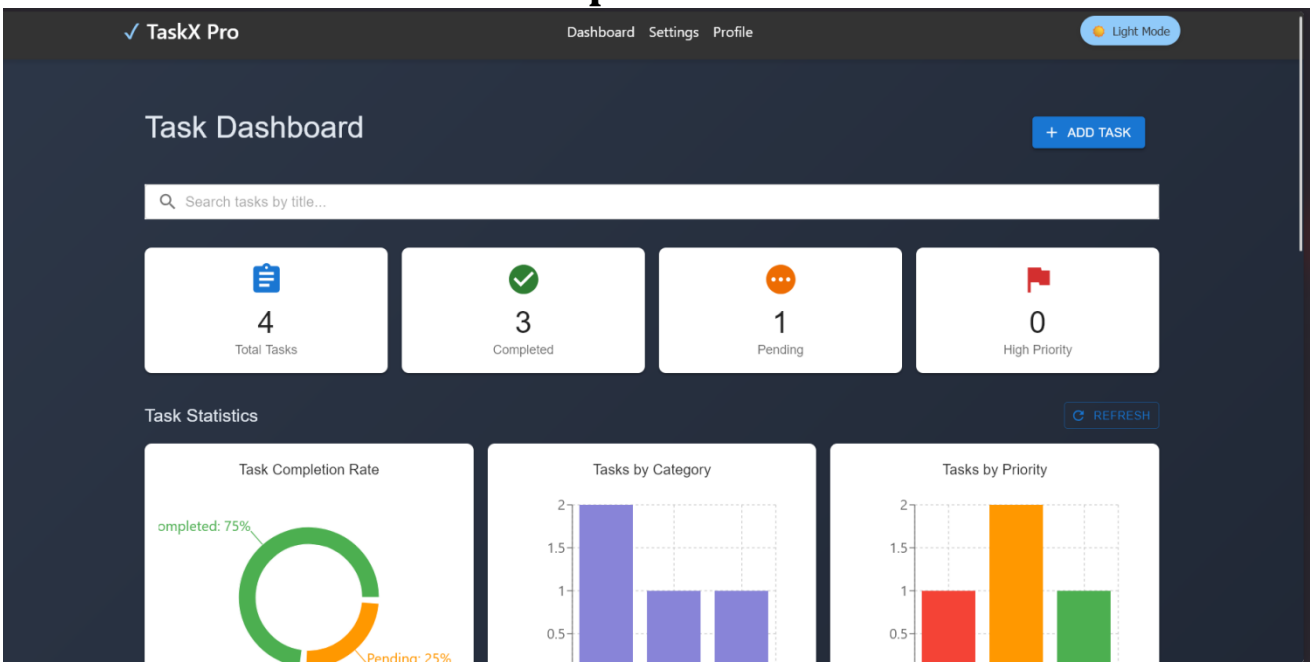

Figure 4.1: Dashboard Page

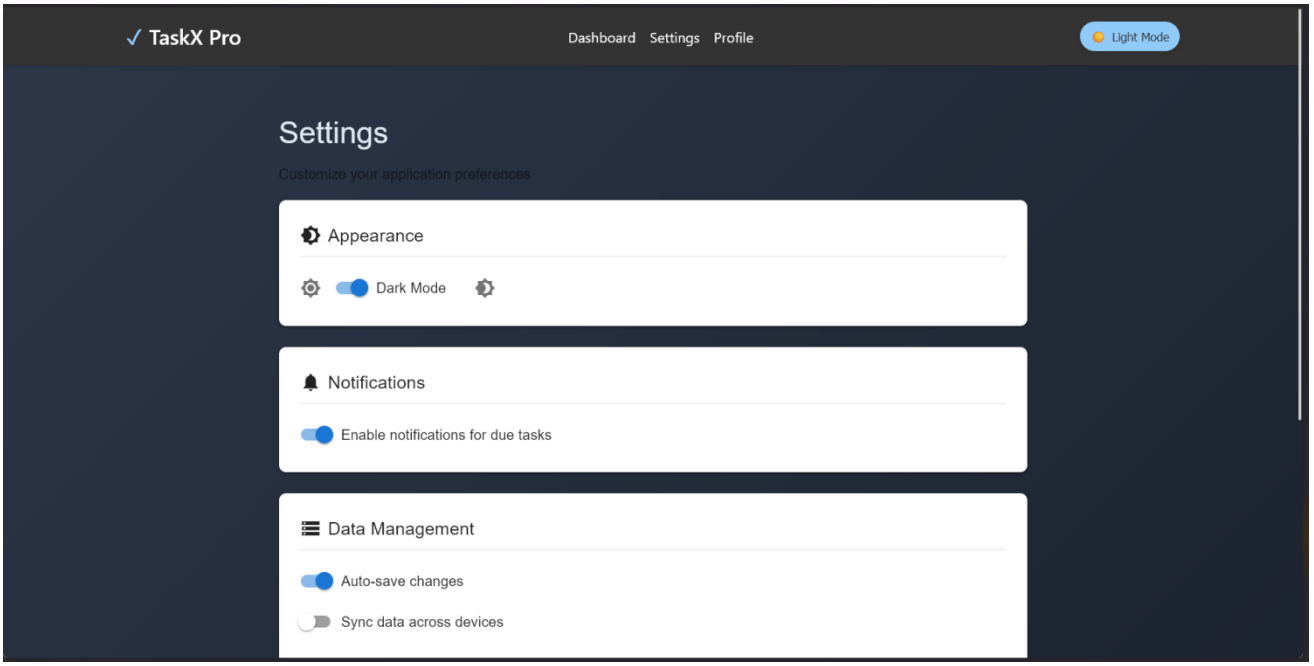

Figure 4.2.Settings Page

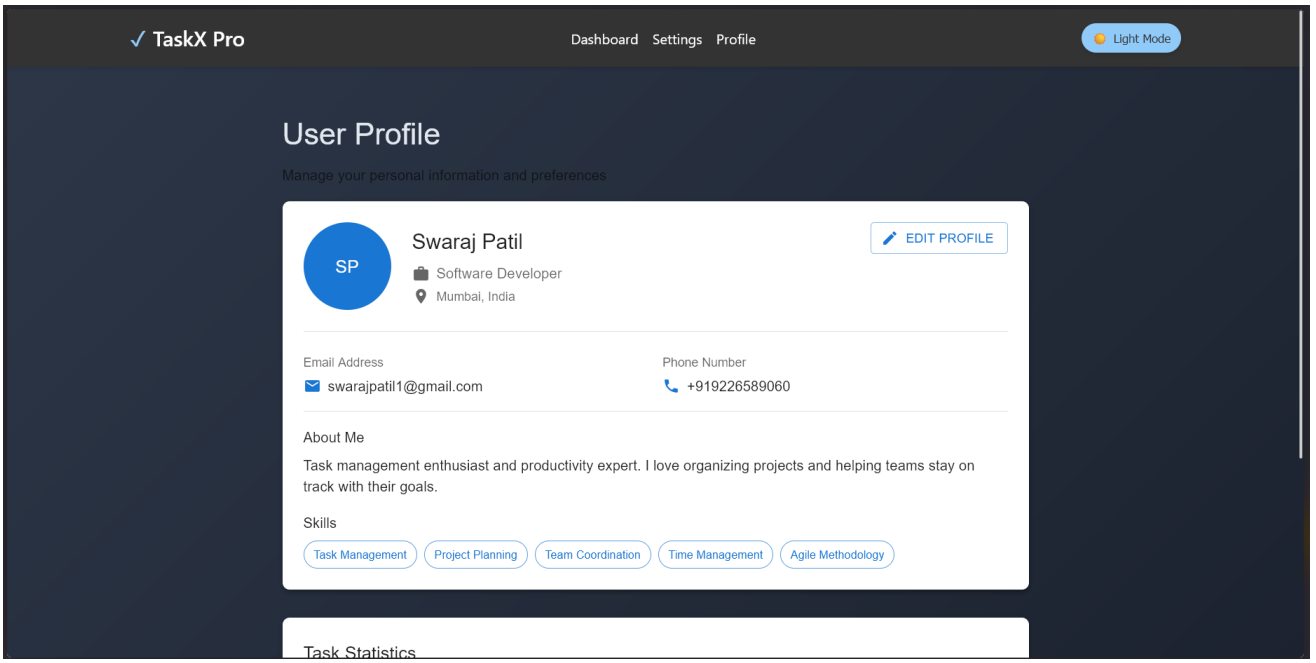

Figure 4.3: User Profile Page

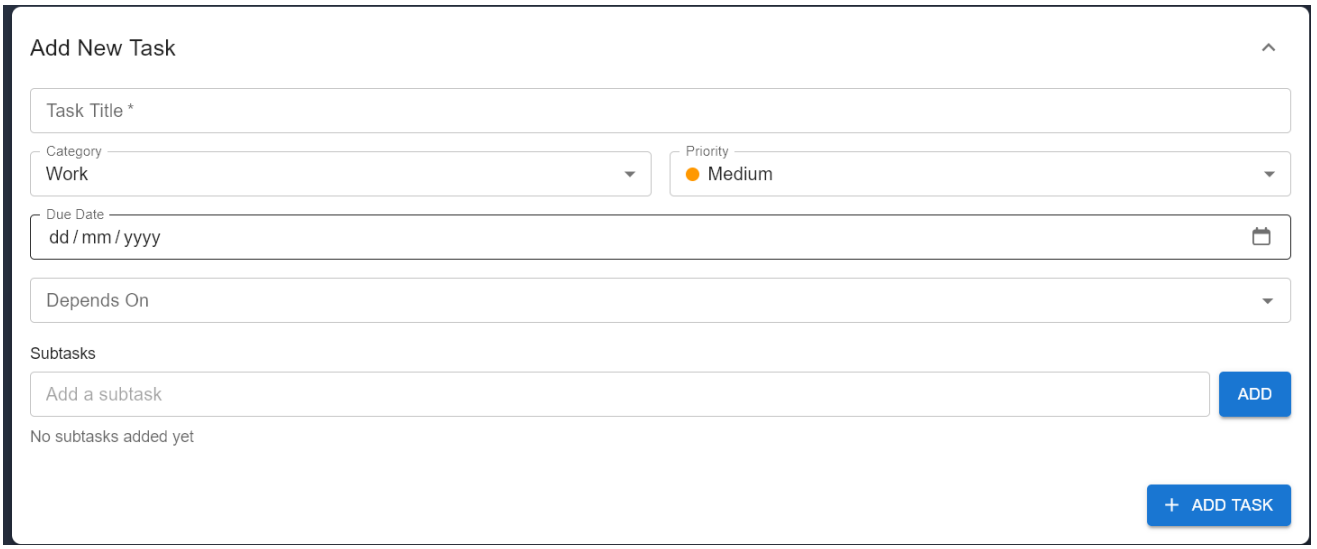

Figure 4.4: Task Section

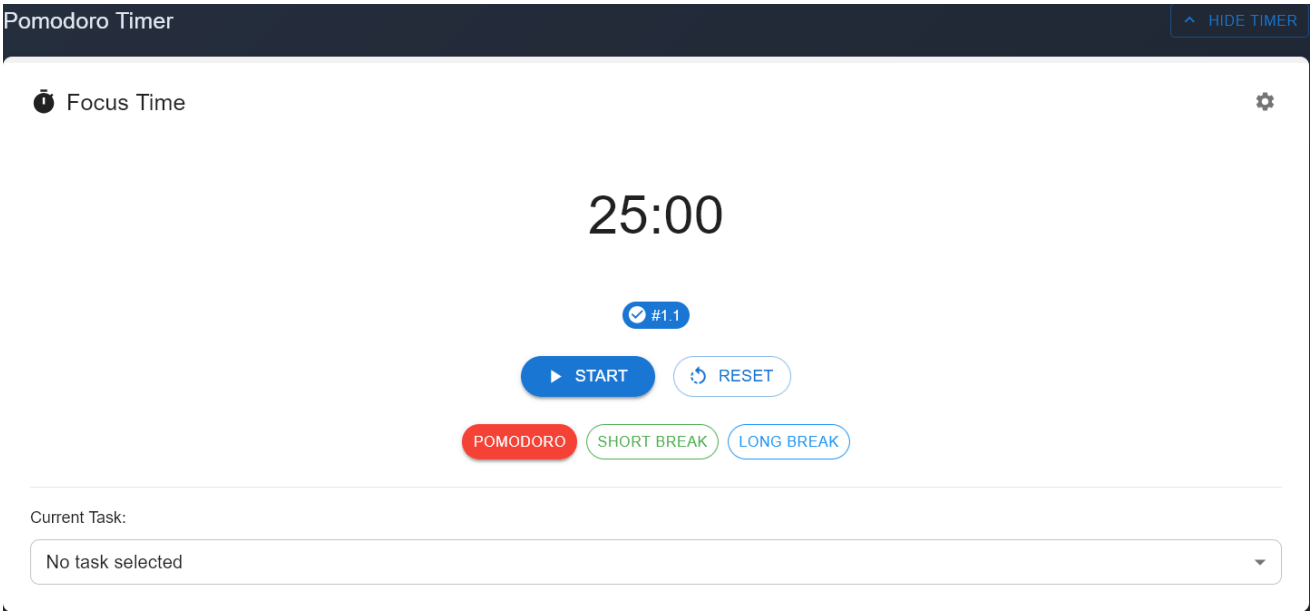

Figure 4.5: Pomodoro Timer Window

Figure 4.6: Task Management Page

4.3. Observation/Remarks

The **TaskX Task Manager System** showcases the practical application of a robust full-stack web application tailored for task and event management. The project leverages a modular and scalable architecture to ensure maintainability and adaptability in real-world use. It integrates **Flask** for efficient backend API routing and business logic, **React with TypeScript** for building a highly interactive and type-safe frontend, and **MongoDB Atlas** for secure, cloud-based data storage.

CHAPTER: 5 CONCLUSION

Chapter 5

Conclusion

5.1 Conclusion

The **TaskX Task Manager System** successfully demonstrates the use of modern full-stack development practices in building a responsive, user-centric, and scalable task management platform. By leveraging technologies such as **Flask** for backend logic, **React with TypeScript** for a structured and component-based frontend, and **MongoDB Atlas** for secure and flexible cloud-based data storage, the system delivers a seamless experience for user registration with **OTP-based email verification**, real-time **task and leaderboard tracking**, and **feedback collection**. This project has enhanced understanding of frontend-backend integration, RESTful API design, secure authentication workflows, and emphasized the critical role of usability, modularity, and real-time data interaction in building modern web applications.

5.2 Future Scope

The future development of the **TaskX Task Manager Web App** presents numerous opportunities for enhancement, aimed at improving user productivity, scalability, and automation.

- **Integration with QR Code Check-ins:**
  Generating QR codes for individual tasks or project boards will allow users to easily share tasks with collaborators, enabling faster onboarding and coordination.
- **Real-Time Leaderboard Updates:**
  Incorporating **WebSocket-based** updates will allow real-time synchronization of task status and leaderboard rankings (if gamification elements are included), enhancing interactivity and engagement.

- **Admin Dashboard and Analytics:**Introducing a role-based admin dashboard will allow organizers to manage Tasks, monitor registrations, view feedback, and download participation analytics with ease.

- **Automated Report and Certificate Generation:**
  Automatically generating **completion reports or certificates** for team members based on task milestones can incentivize users and streamline end-of-project formalities.

- **Integration with Third-Party Email Services:**
  Using tools like SendGrid or Mailgun for high-volume email delivery can improve reliability and ensure timely delivery of OTPs and confirmation emails.

Bibliography

[1] R. Kumar and A. Singh, "A Comparative Study of Modern Web Frameworks for Full-Stack Development", International Journal of Computer Applications, vol. 183, no. 22, pp. 18-24, 2021.

[2] A. Ramesh and L. Joshi, "OTP-Based User Authentication for Secure Login in Web Applications", International Journal of Cybersecurity, vol. 9, no. 1, pp. 45–51, 2023.

[3] M. Desai and S. Thakkar, "Responsive Web Design Techniques in Modern UI Frameworks", International Journal of Web & Semantic Technology (IJWesT), vol. 13, no. 3, pp. 32–39, 2022.

[4] P. Sharma and R. Patel, "Gamification in Web Applications: Impact of Leaderboards on User Engagement", Journal of Interactive Media, vol. 6, no. 2, pp. 28–35, 2020.

[5] T. S. Verma and D. Singh, "Feedback Systems in Web Applications for Continuous Improvement", International Journal of Human-Computer Studies, vol. 11, no. 4, pp. 64–71, 2021.