Task Manager Web Application

**Team Leads (Software Development Internee)**

**Name : Taha Azhar**

**Intern-Software Development**

**Table Of Content**

Contents

***1.Introduction***  
1.1 Background  
1.2 Objective of the Internship  
1.3 About the Host Organization

***2.Technology Stack***  
2.1 Frontend Technologies  
2.2 Backend Technologies  
2.3 Database and Tools Used

***3.Project Overview***  
3.1 Project Title  
3.2 Problem Statement  
3.3 Project Description  
3.4 Features Implemented

***4.System Architecture***  
4.1 Folder Structure  
4.2 Application Flow  
4.3 Routing and Navigation

***5.Setup and Installation***  
5.1 Backend Setup  
5.2 Frontend Setup  
5.3 Environment Configuration

***6.Usage Guide***  
6.1 Signup and Login  
6.2 Adding Tasks  
6.3 Editing and Deleting Tasks  
6.4 Search and Filtering  
6.5 Progress Tracking  
6.6 Logout Functionalality

***7.Screenshots & UI Previews*  
7.1 Login/Signup Screens  
7.2 Dashboard View  
7.3 Task Management UI**

***8.Challenges Faced & Solutions***

***9.Conclusion*  
9.1 Key Learnings  
9.2 Skills Improved**

***10.References*  
10.1 Tools & Libraries Used  
10.2 Online Resources**

***11.Appendix*  
11.1 GitHub Repository Link  
11.2 Demo Video Link  
11.3 Contact Information**

# **ACKNOWLEDGEMENT**

*I would like to express my sincere gratitude to all those who supported me during my internship and contributed to the successful completion of this project.*

*First and foremost, I am thankful to* ***Faizan Khan****, my internship supervisor at* ***Team Leads (Software Development)****, for their valuable guidance, consistent support, and encouragement throughout the internship period. Their insights and feedback greatly helped me improve both my technical and professional skills.*

*I am also grateful to the entire team at* ***Team Leads (Software Development)*** *for providing a positive learning environment and for being approachable and helpful whenever I needed assistance.*

*A special thanks to my academic mentors and faculty members at* ***Faizan Khan/HR Team****, whose foundational teachings and continuous support enabled me to apply my knowledge effectively during this internship.*

*Lastly, I would like to thank my family and friends for their constant motivation and encouragement during this learning journey.This internship has been a valuable experience, and I am truly thankful for the opportunity to contribute, learn, and grow both personally and professionally.*

# 

# **ABSTRACT**

*This report presents the development of a* ***Task Manager Web Application*** *built using the* ***MERN (MongoDB, Express, React, Node.js)*** *stack as part of my internship project. The goal of the application is to provide a simple, intuitive, and effective tool to manage tasks and monitor progress in real time.*

*The system offers a complete set of features for user authentication (signup/login), task creation, editing, deletion, and search functionality. It also includes a dynamic progress tracker to visually represent task completion, enhancing productivity and clarity for the user.*

*The frontend of the application is developed using* ***React.js****, ensuring a smooth and responsive user experience. The backend is powered by* ***Express.js*** *and connected to a* ***MongoDB Atlas*** *database, enabling seamless data storage and retrieval.*

*This internship project not only demonstrates technical knowledge in full-stack web development but also reflects my ability to build secure, user-friendly applications with real-world usability. Through this project, I have gained hands-on experience with RESTful API development, component-based frontend design, protected routing, and integration of modern web technologies.*

# **Scope of Project:**

## *DESIGN SPECIFICATION*

# *SOLUTION:*

*We will design and build a Web Application named* ***'Task Manager'***

The design of the application follows a **component-based architecture** and a **RESTful API pattern**, ensuring modularity, scalability, and ease of maintenance.

**🔷 *5.1 Frontend (React.js)***

* Built using **React functional components**
* State management with useState and useEffect
* Routes handled by **React Router DOM**
* Forms use basic validation (e.g., required fields, password match)
* Responsive layout with custom CSS and basic design system

***Key Components***

|  |  |
| --- | --- |
| ***Components*** | ***Responsibility*** |
| ***AuthPage*** | ***Handles login and signup forms*** |
| ***Taskform*** | ***Add/Edit tasks inputs*** |
| ***Home*** | ***Displays task list, Progress bar, Search*** |
| ***App.js*** | ***Manages routes and user authentication state*** |

**🔷 *5.2 Backend (Node.js + Express)***

* Built with **Express.js REST API**
* MongoDB used as the database (hosted on **MongoDB Atlas**)
* Mongoose used for schema definition and queries
* Routes defined for CRUD operations on tasks

***API Endpoints Example:***

|  |  |  |
| --- | --- | --- |
| ***Method*** | ***End Point*** | ***Description*** |
| GET | /tasks | Fetch all tasks |
| POST | /tasks | Add new tasks |
| PUT | /task/:id | Update existing task |
| DELETE | /task/:id | Delete a task |

## 

# **Functional Requirements:**

### ***The functional requirements define the core features and behaviors that the system must support to ensure it meets user needs. The following requirements outline the expected functionality of the Task Manager Web Application.***

1. User Authentication
2. Task Management
3. Task Search and Filtering
4. Progress Tracking
5. Navigation and Routing

# **Problem Statements**

***In today's fast-paced digital environment, individuals and teams often struggle to effectively manage and keep track of daily tasks, deadlines, and progress. Many users either rely on manual methods like notepads or spreadsheets, or they find existing task management tools overly complex or cluttered for simple, focused usage.***

***There is a growing need for a lightweight, user-friendly, and accessible task management solution that allows users to:***

* ***Easily create and manage tasks***
* ***Monitor progress with clear visual indicators***
* ***Search and filter tasks efficiently***
* ***Access their data securely with authentication***
* ***Work within a clean, responsive interface***

***This project aims to solve this problem by building a full-stack Task Manager Web Application using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The application provides essential features like user authentication, task CRUD operations, progress tracking, and search functionality — all within a minimal yet functional design. The solution targets individuals or small teams who need a simple, fast, and intuitive way to organize their work.***

***Documentation:***

***Users often struggle with managing tasks due to the lack of simple, effective tools. Many available task managers are either too complex or lack essential features. There is a need for a lightweight, easy-to-use web application that allows users to manage tasks, track progress, and maintain focus in a streamlined digital environment.***

***The documentation of an ideal project will be in the form of a project report comprising of the following documents:***

***Abstract.***

***Table of Contents.***

***Scope of Project.***

***Functional Requirement Specification.***

***Screenshots/Interface.***

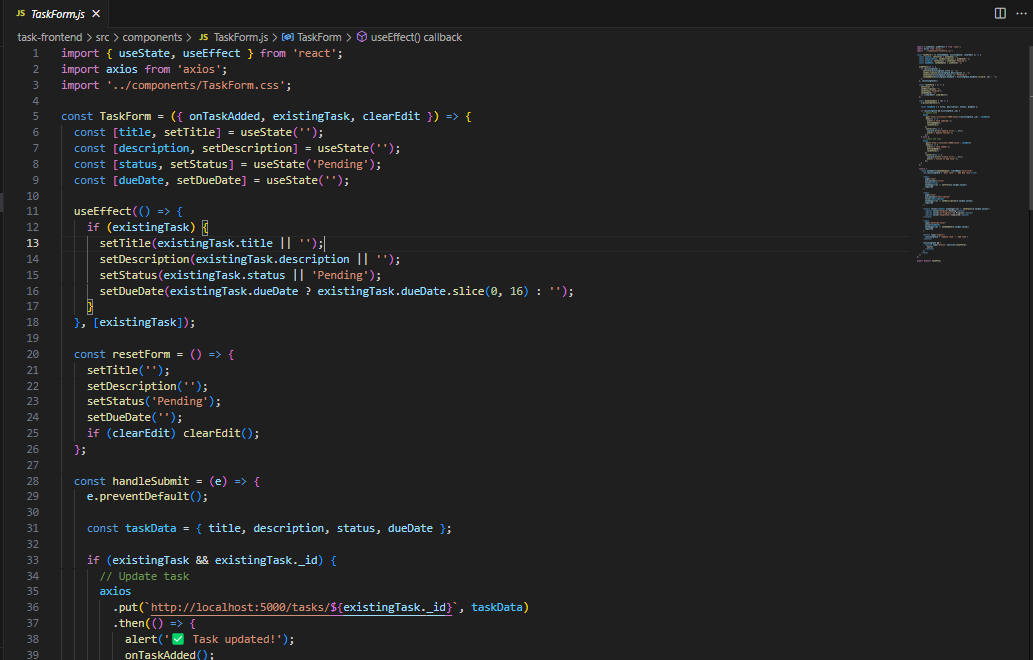
***.***

**Screenshots:**

*Interface:*

***Frontend***

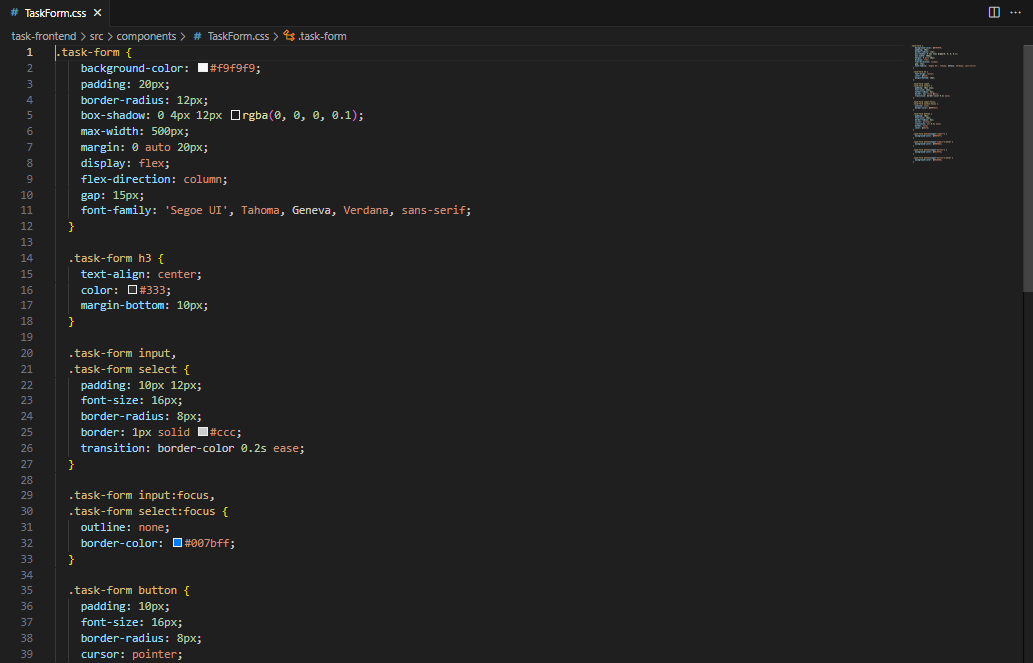
***Component ( TaskForm.js / TaskForm.css )***

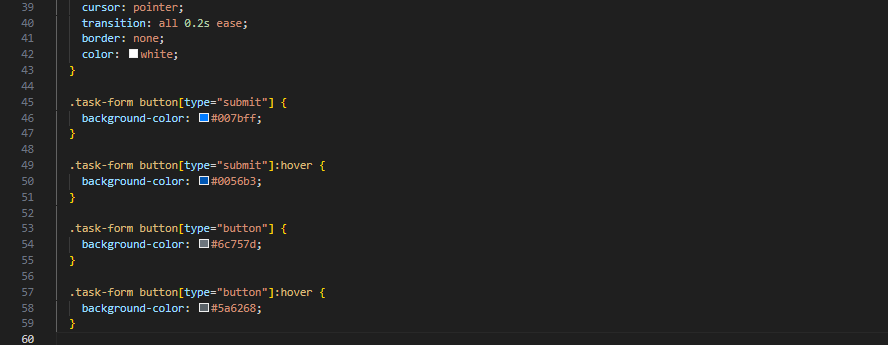
**



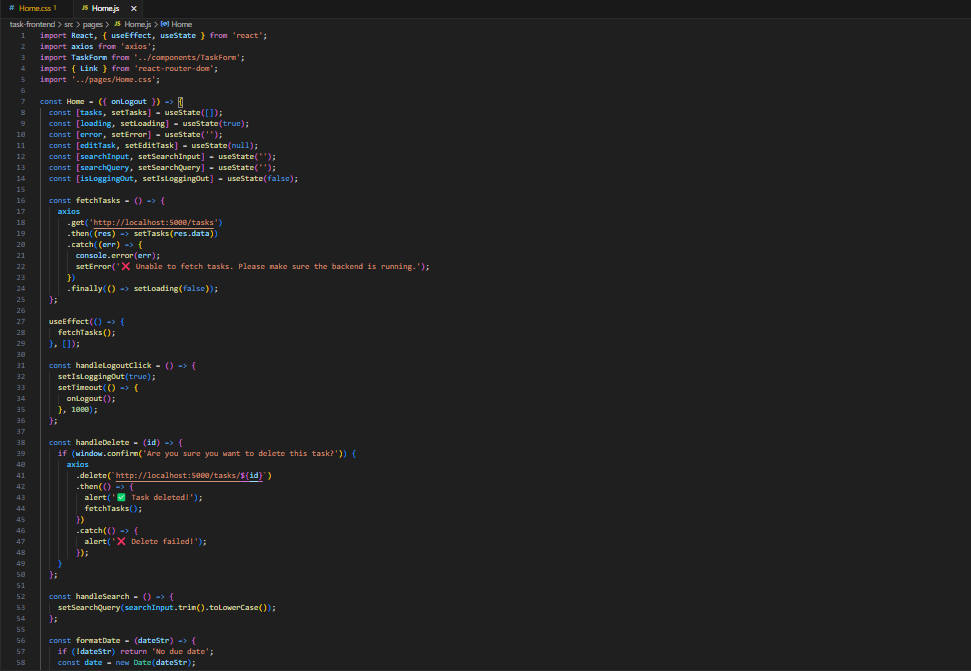
### 

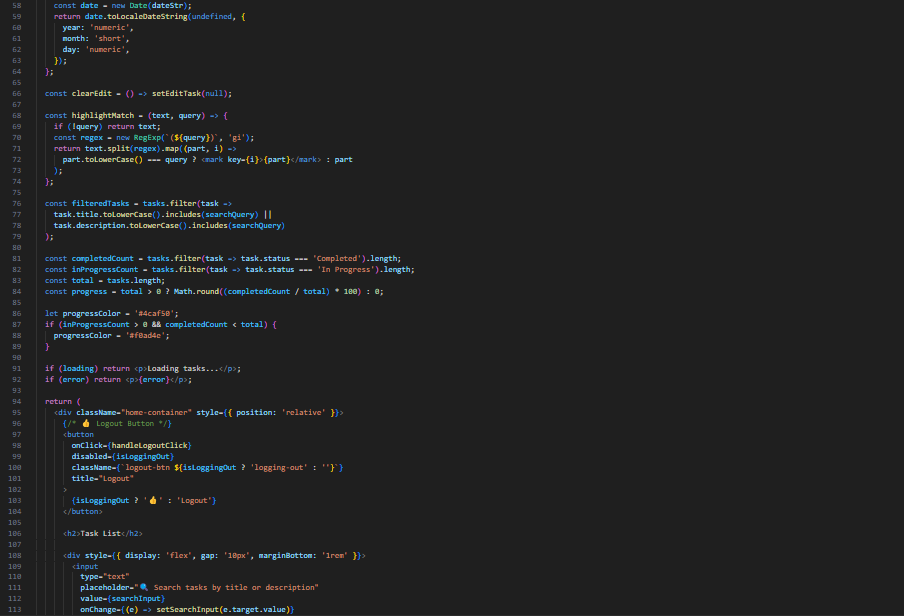
***Task Form.css***

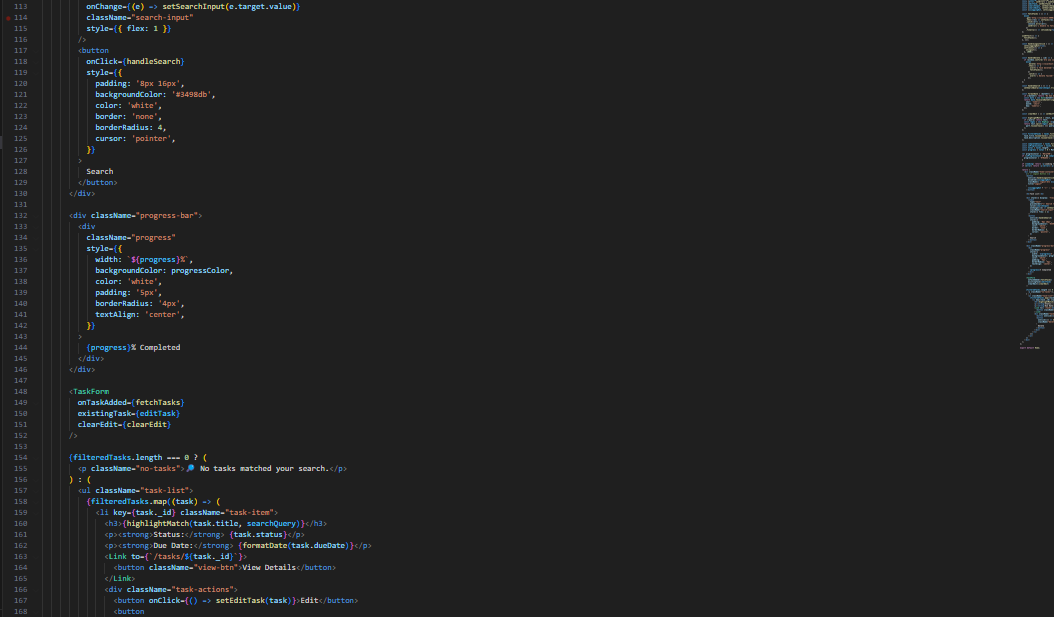


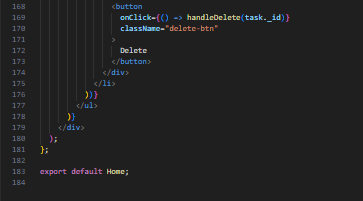


***PAGE ( Home.js / Home.css );***

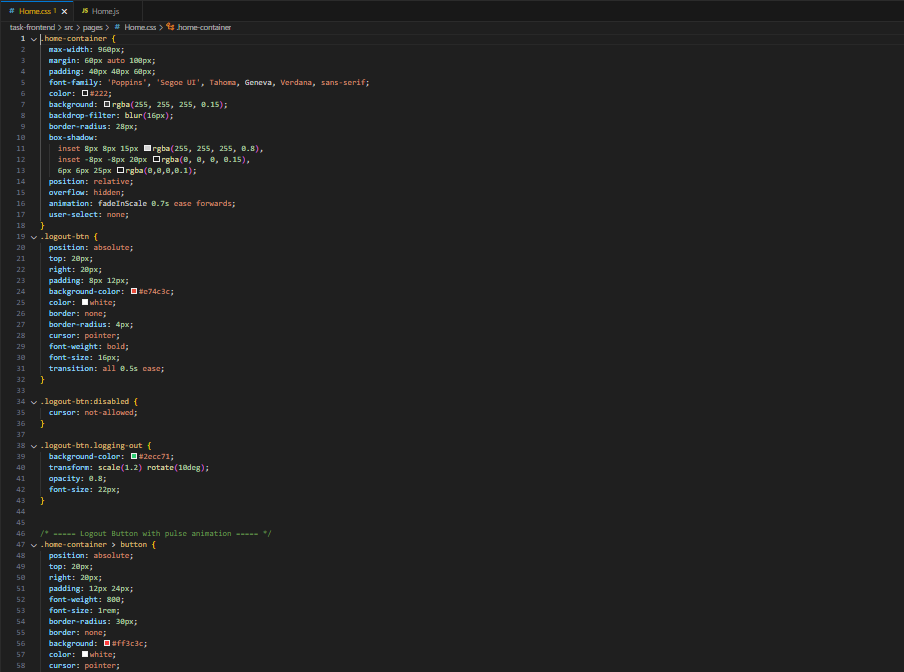
******

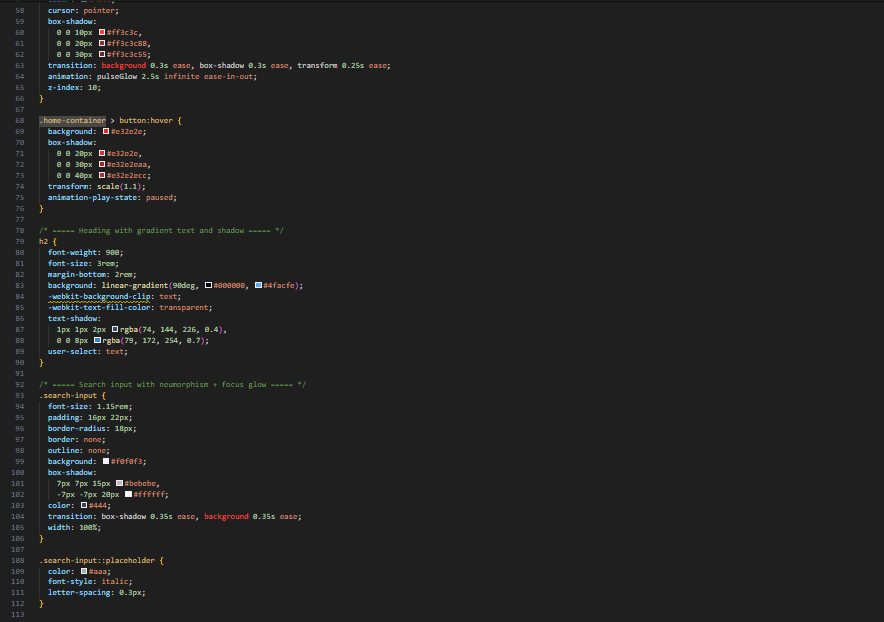


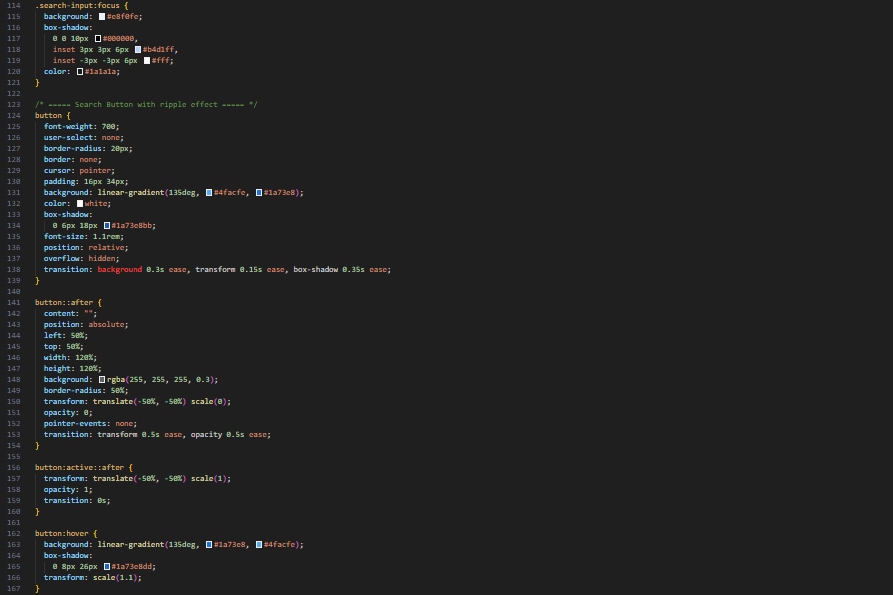


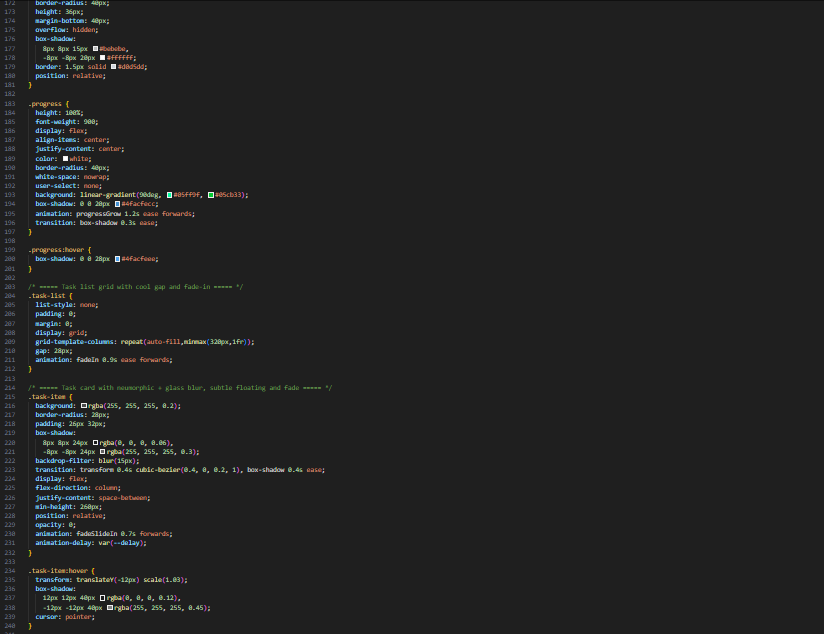


***Home.css***

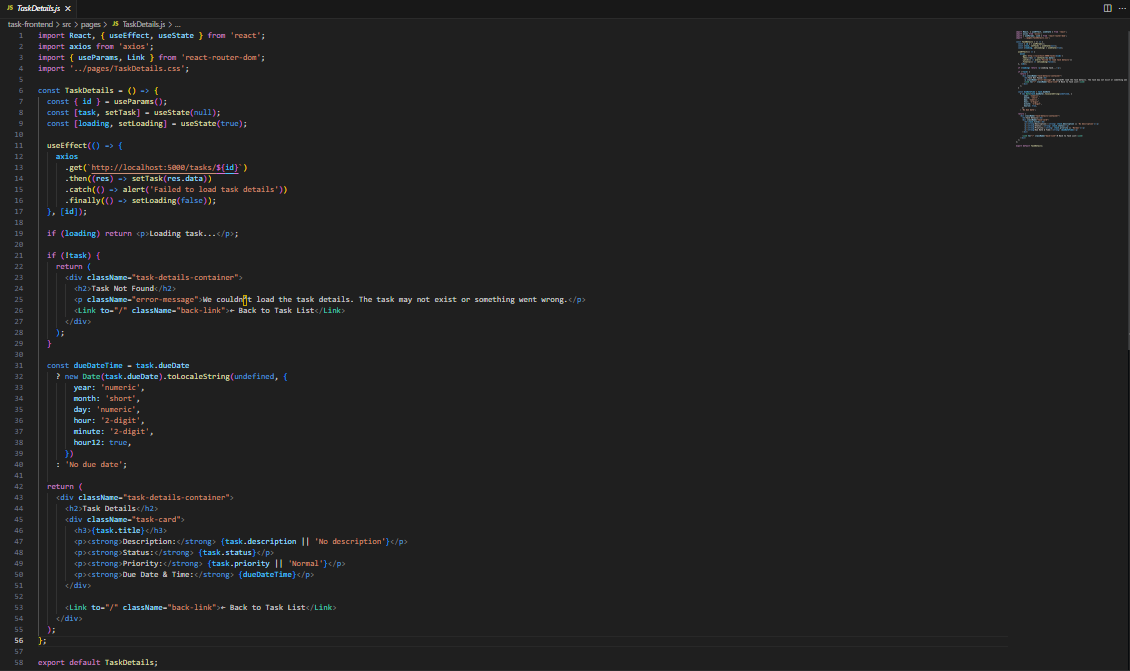
******



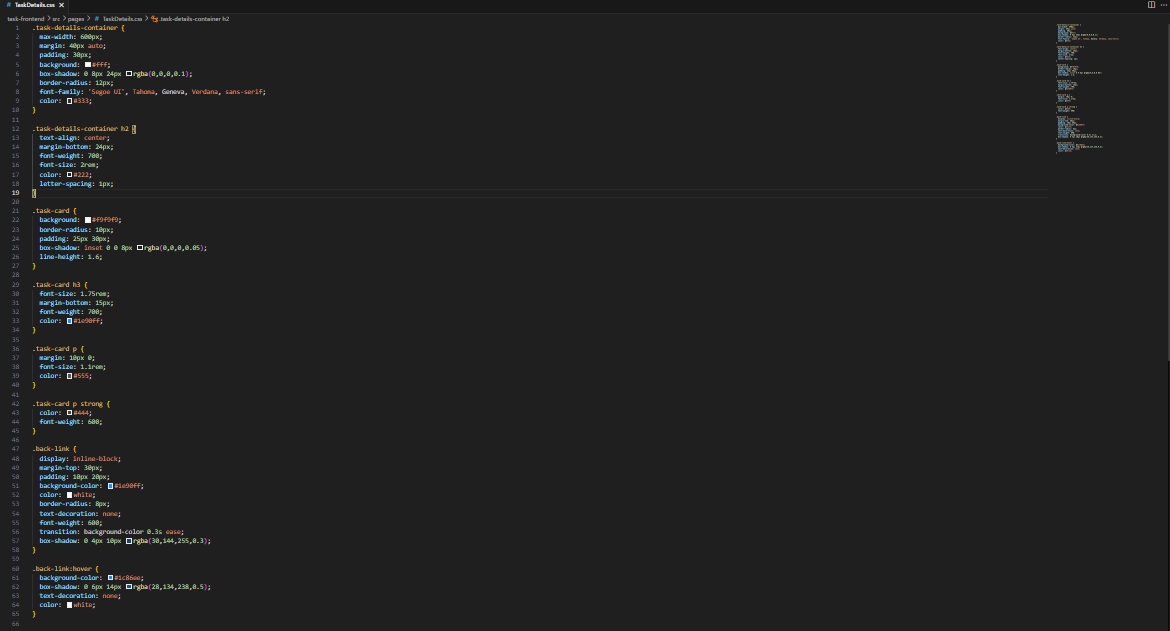




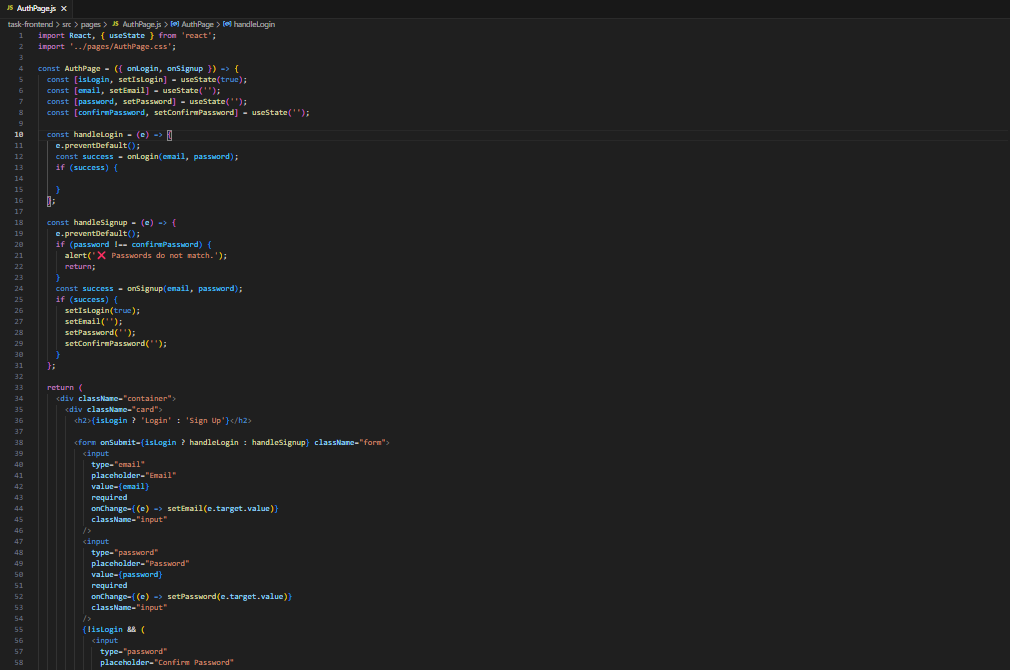
***PAGE ( Taskdetails.js / Task details.css ):***



***Taskdetails.css***

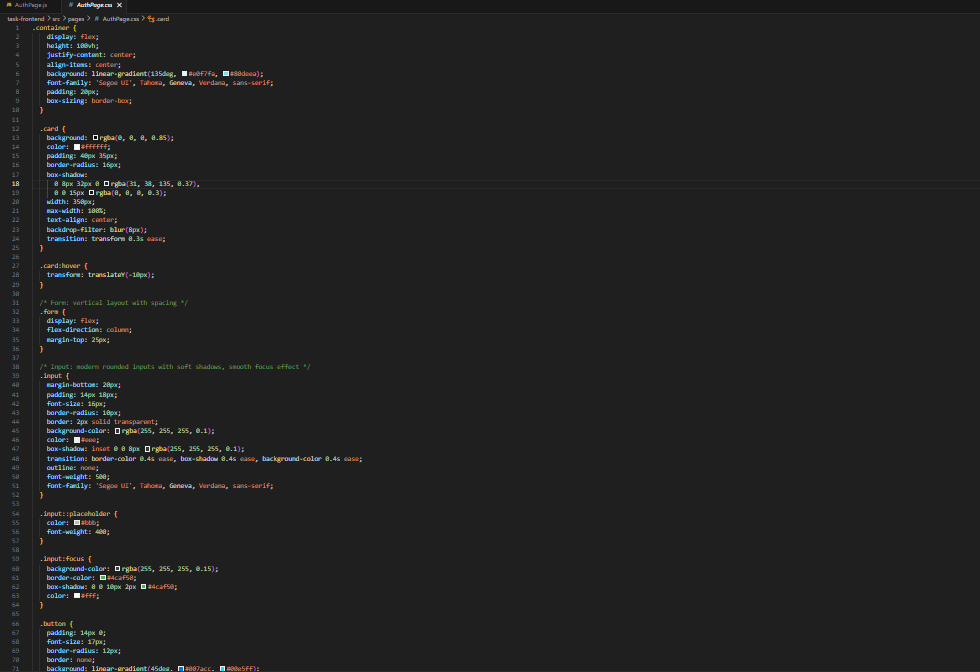
******

***PAGE ( AuthPage.js / AuthPage.css ):***



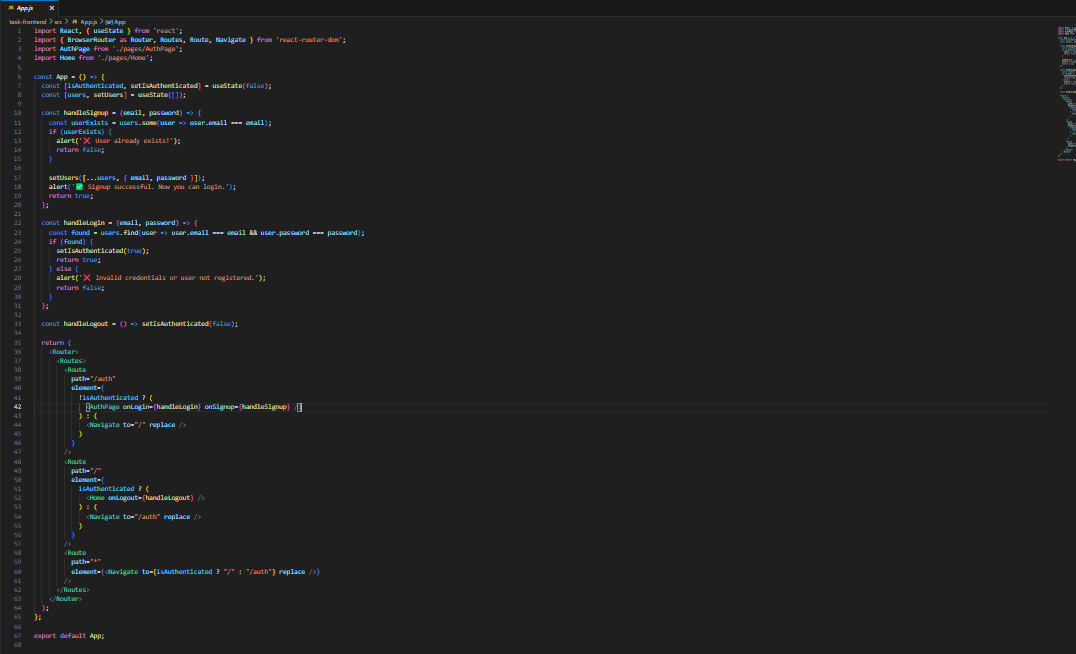


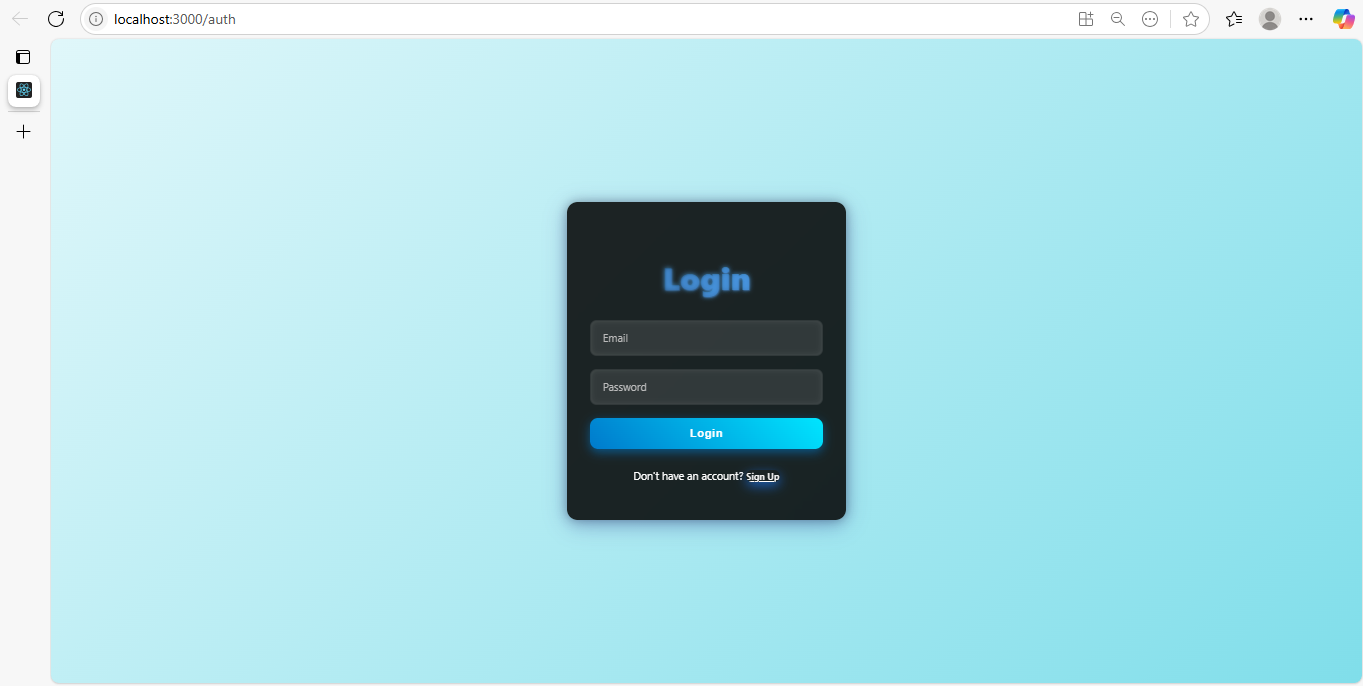
***Auth Page.Css***

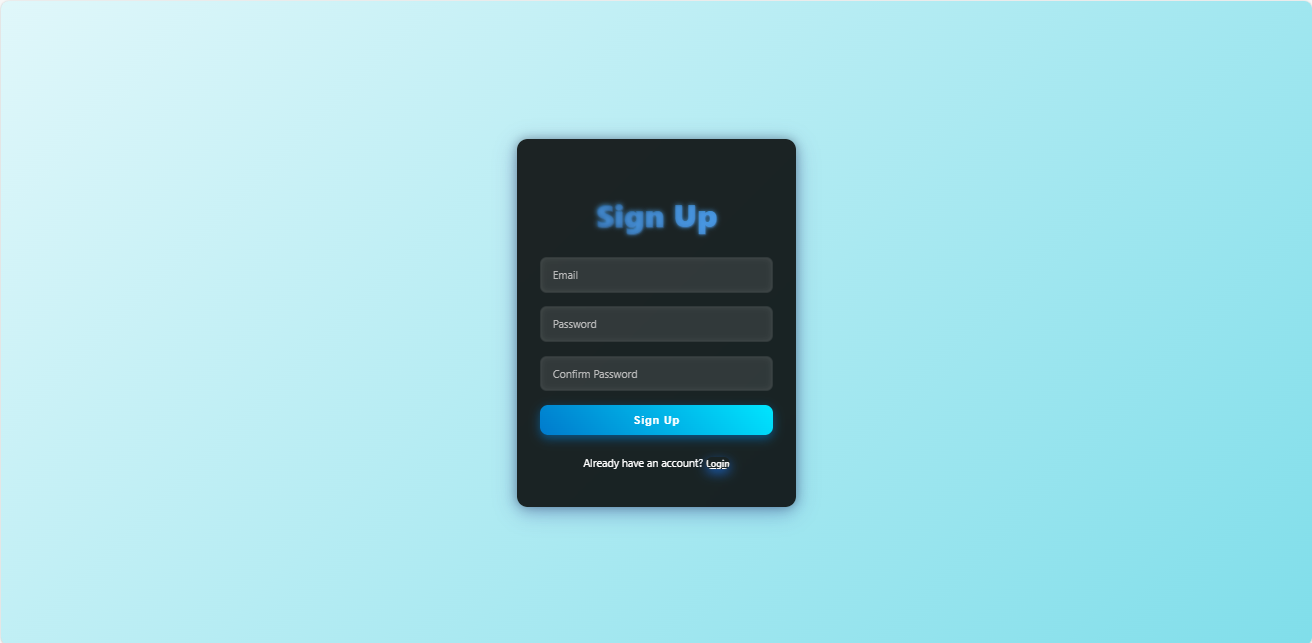




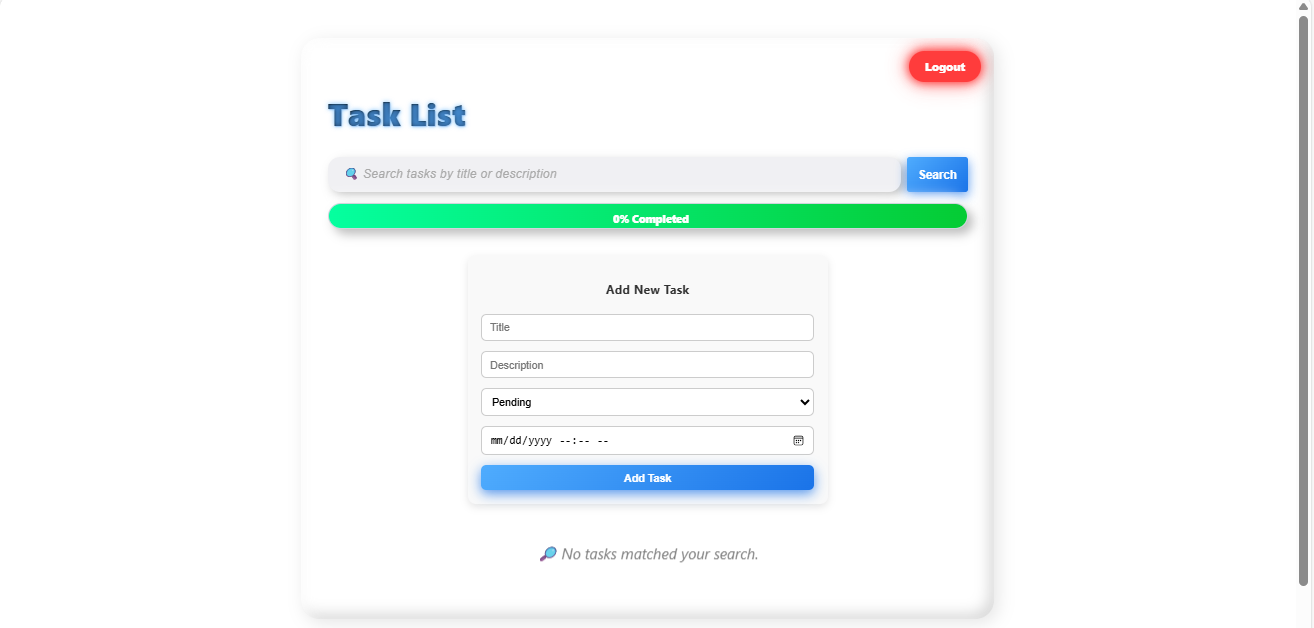
***App.js***

***Auth Page :***

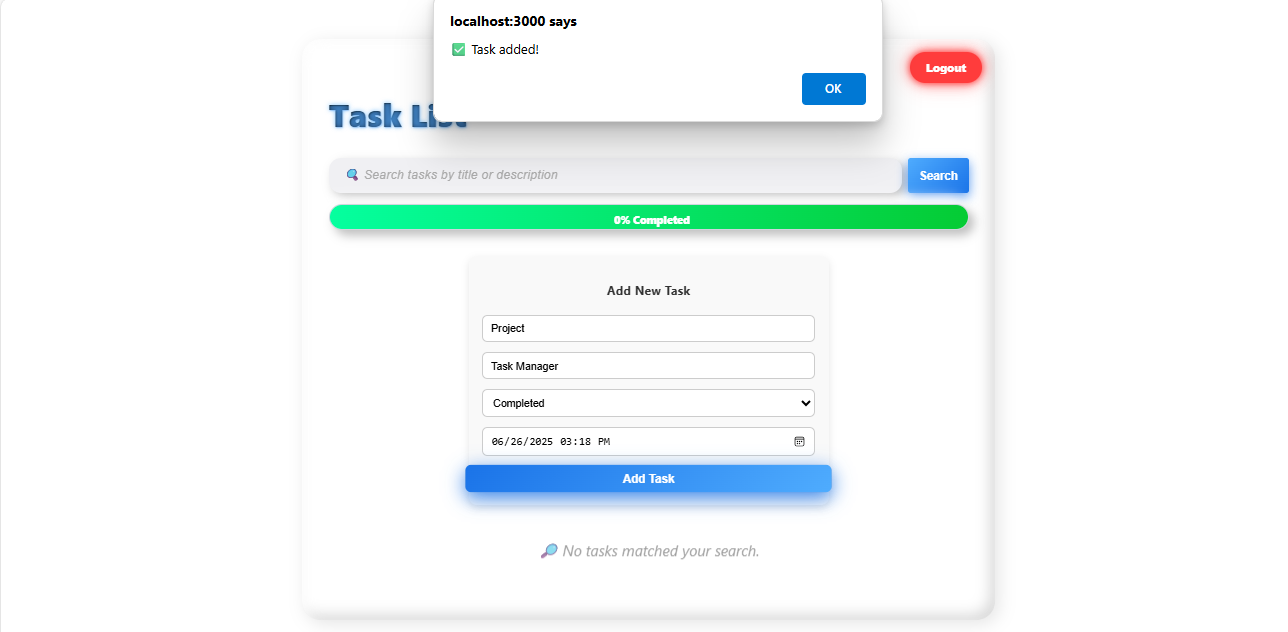


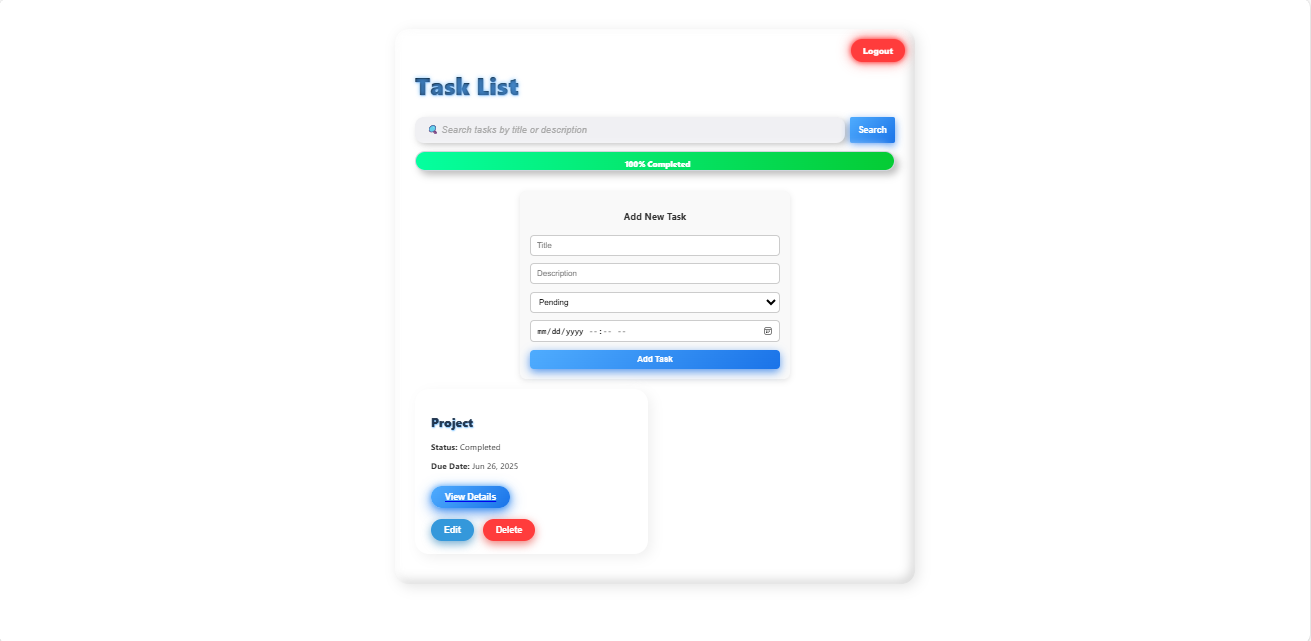


***Home:***

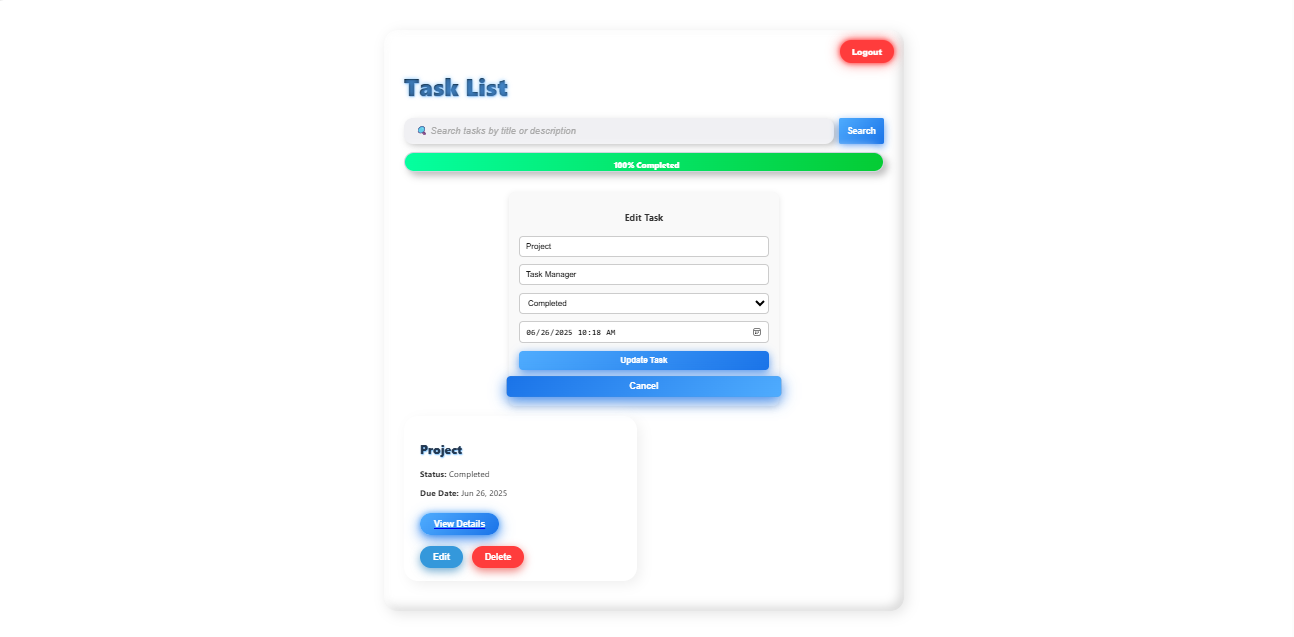


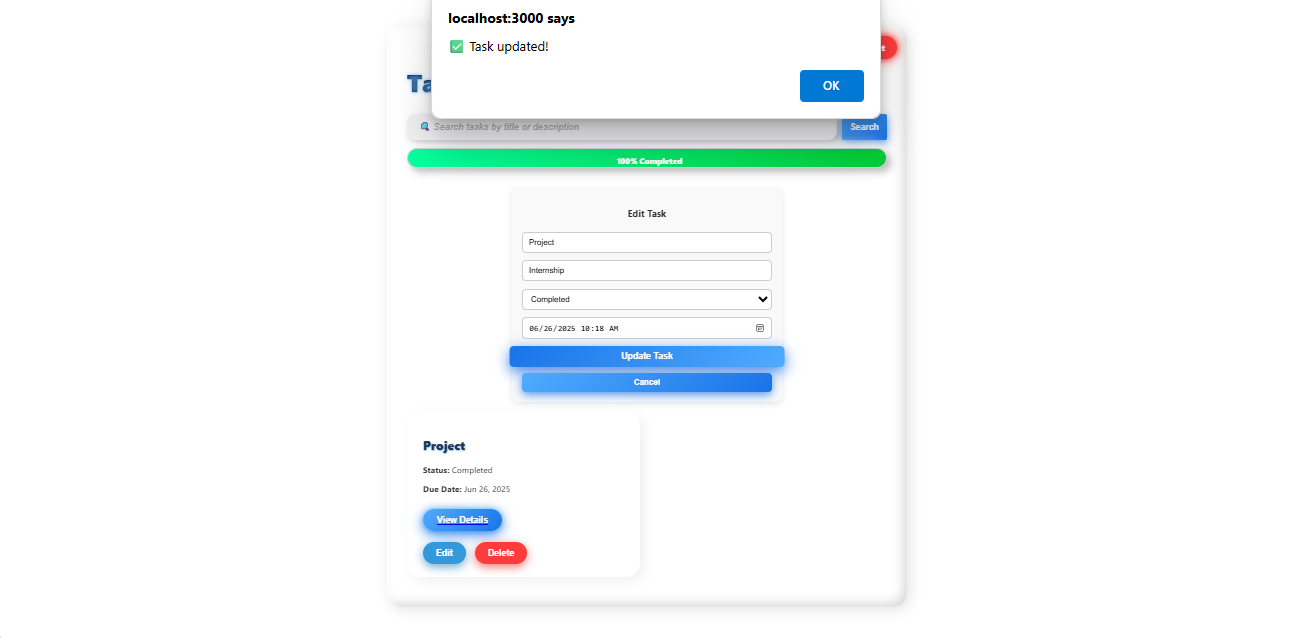
***Task Added with Details:***

******

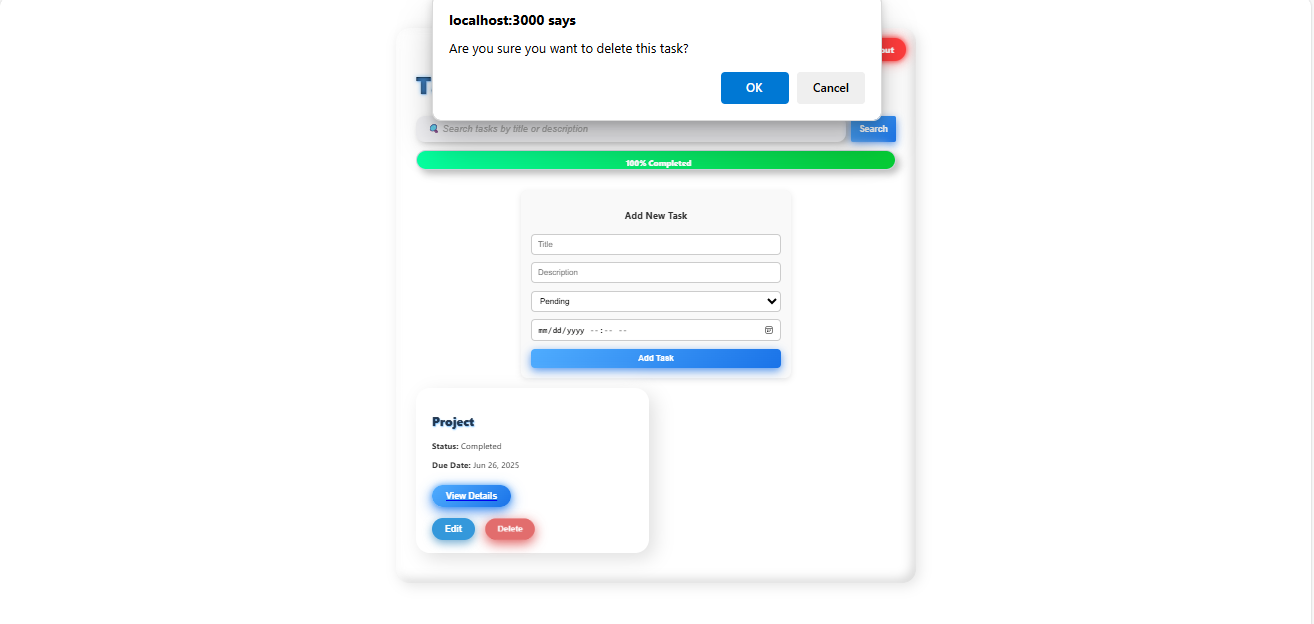


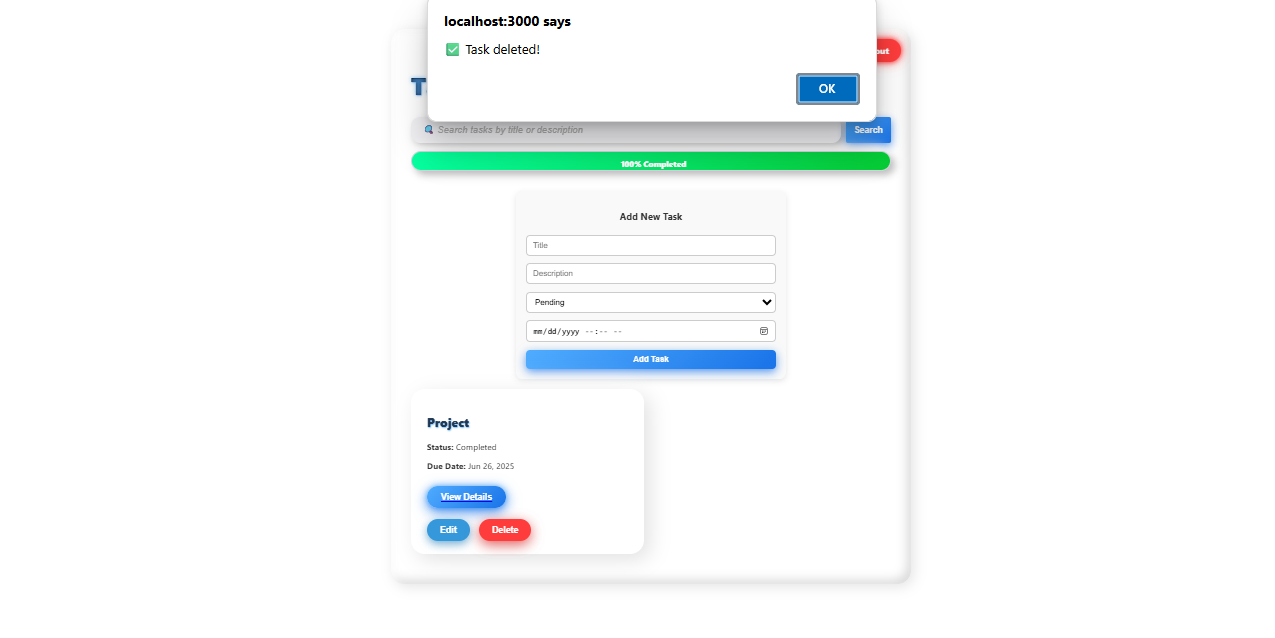
***Then Edit Task:***

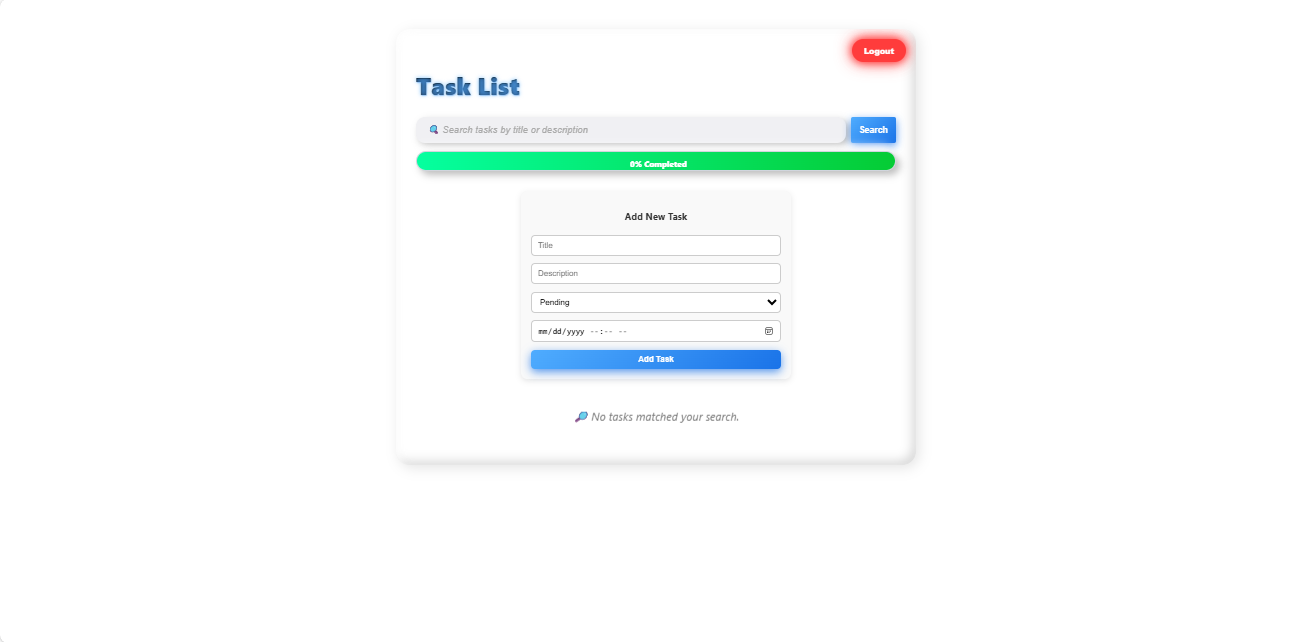




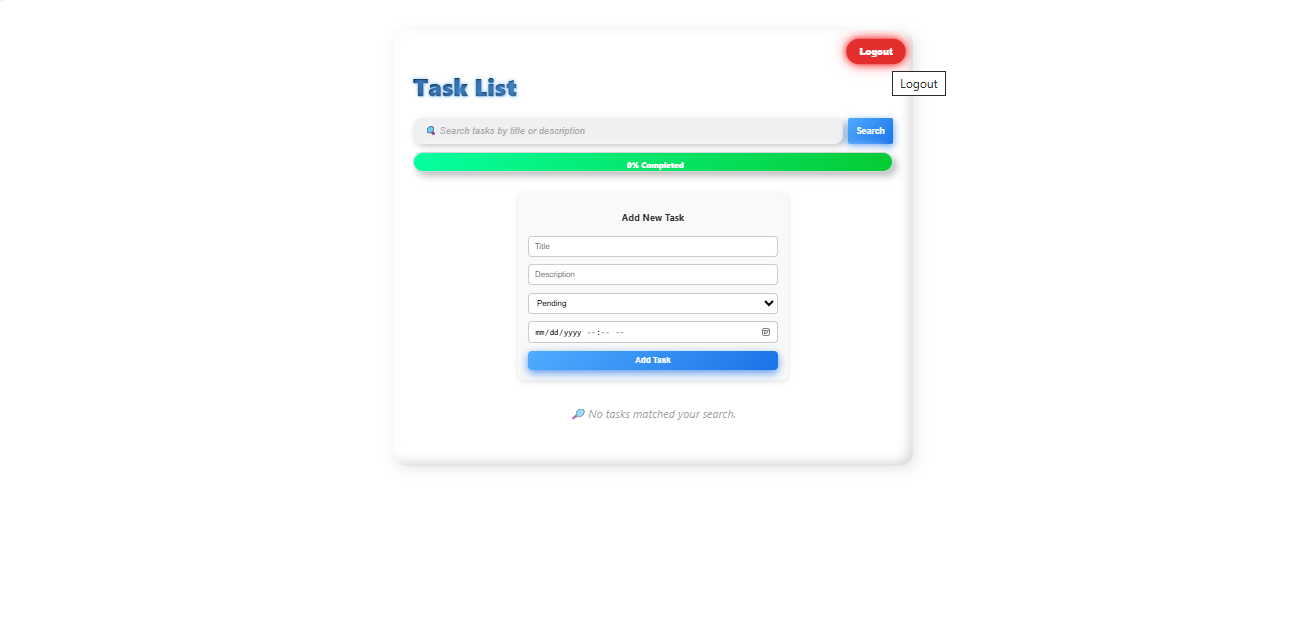
***Delete Task:***

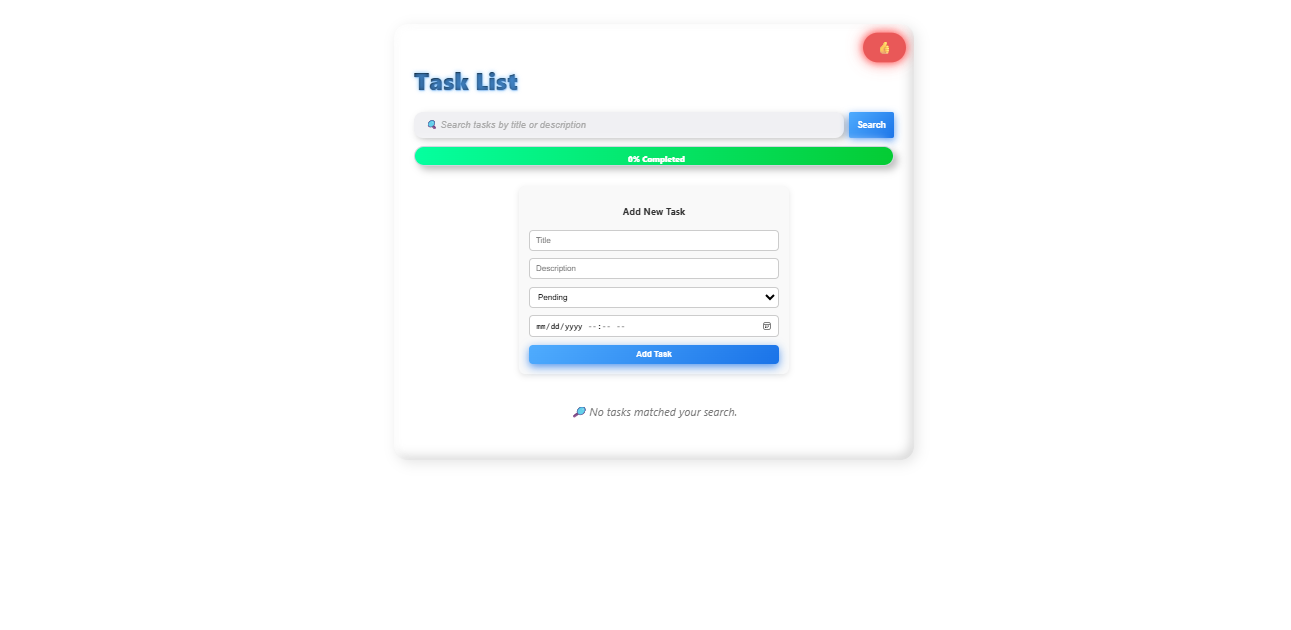




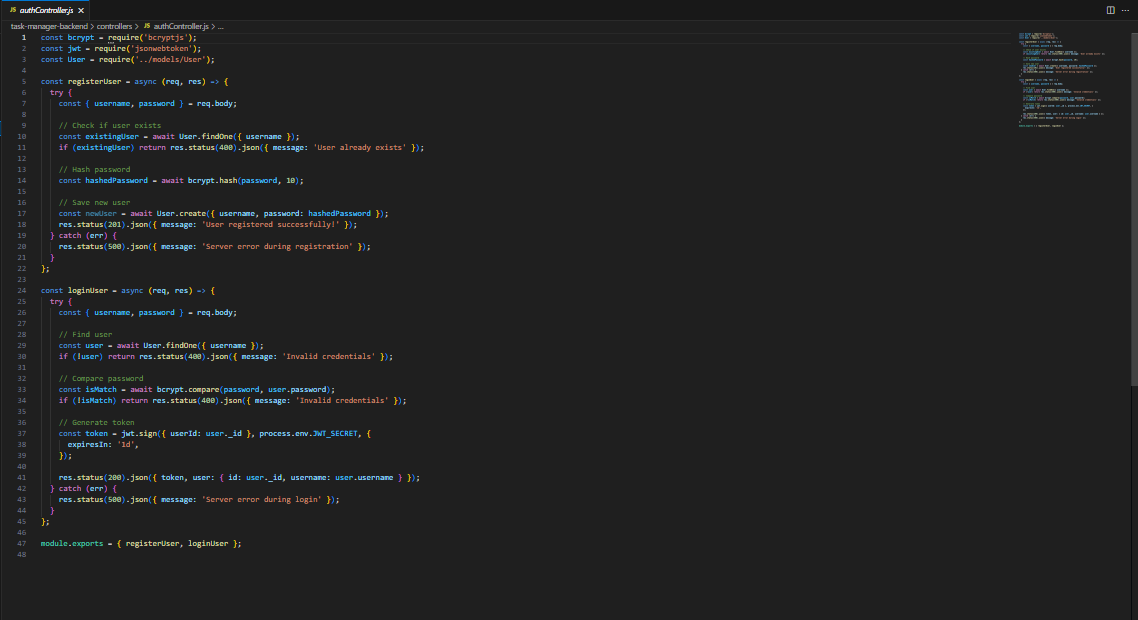


***Then Logout:***

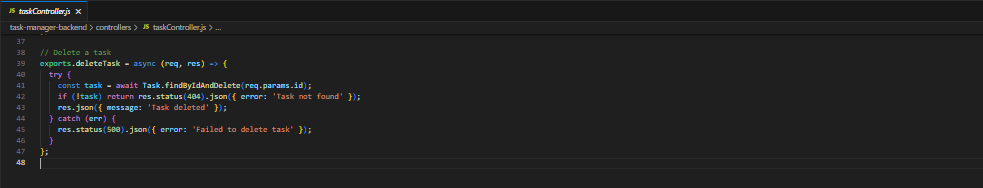




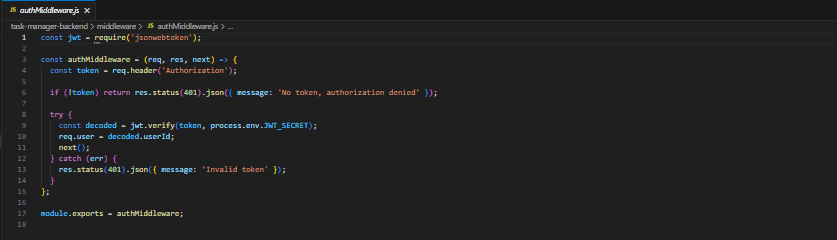
***BACKEND Screenshots***

***AuthController.js***

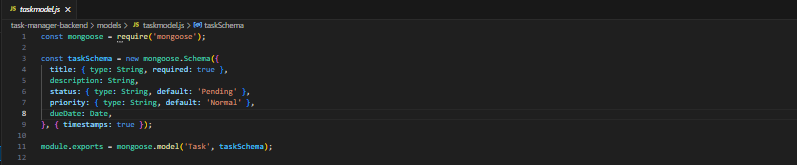
***Taskcontroller.js***

******

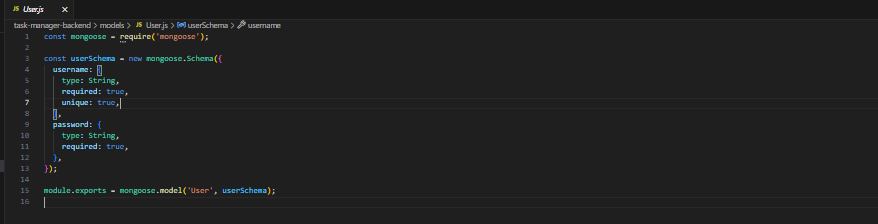
***authMiddleware.js***



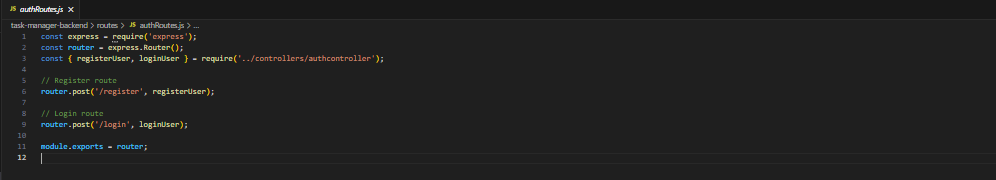
***taskModel.js***



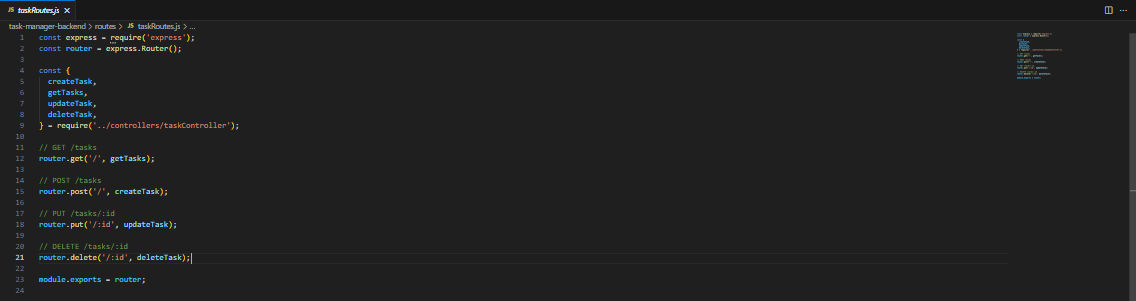
***UserModel.js***



***AuthRoutes.js***



***taskRoutes.js***

******

## 

Thankyou.