Task Management Application

Introduction

The Task Management Application is designed to allow users to manage their tasks efficiently. Users can create, read, update, and delete tasks through an intuitive user interface. The application is built using modern web technologies, ensuring responsiveness and usability across different devices.

Technology Stack

- Front-end: HTML, CSS , JavaScript
- Back-end: Node.js, Express.js, MongoDB
- Deployment Platforms: GitHub

Features

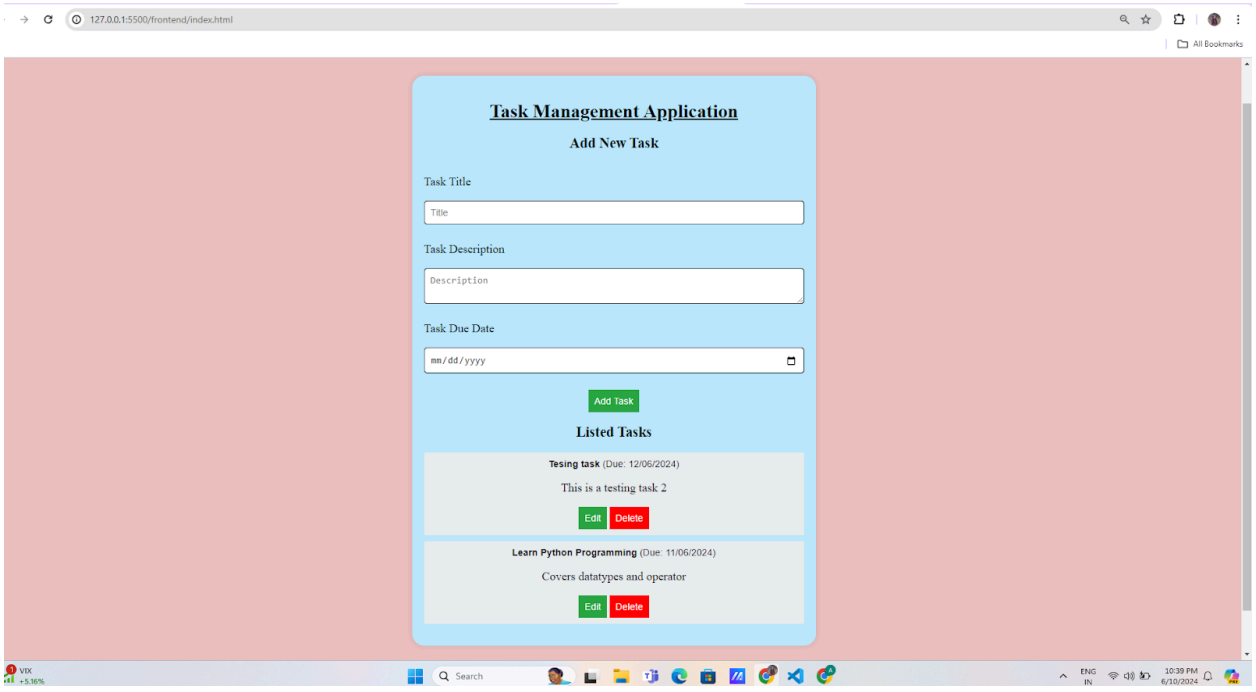
Front-end:

1. Landing Page: Displays a list of tasks.
2. Add New Task: Users can add tasks with a title, description, and due date.
3. View Task Details: Users can view detailed information about each task.
4. Edit Task: Users can edit existing tasks.
5. Delete Task: Users can delete tasks.
6. Responsive Design: The application is designed to be responsive and usable on both desktop and mobile devices.

Back-end:

1. RESTful API: Handles CRUD operations for tasks.
2. Endpoints:
    - GET /api/tasks: Retrieve all tasks.
    - POST /api/tasks: Create a new task.
    - GET /api/tasks/:id: Retrieve a task by its ID.
    - PUT /api/tasks/:id: Update an existing task.
    - DELETE /api/tasks/:id: Delete a task.

Output



Challenges Faced

- **State Management**: Ensuring the front-end UI stays in sync with the back-end data was a key challenge.
- **Responsive Design**: Making the UI look good on both desktop and mobile devices required careful use of CSS and Bootstrap classes.
- **Error Handling**: Implementing comprehensive error handling for both the front-end and back-end to ensure a smooth user experience.

Future Improvements

- **User Authentication**: Add user authentication to allow multiple users to manage their own tasks.
- **Task Categories**: Implement categories for tasks to help users organize their tasks better.
- **Priority Levels**: Allow users to set priority levels for tasks.
- **Notifications**: Add email or SMS notifications for due dates.