Task management Application Using React

Introduction: This project is a Task Management Application developed using React, designed to help users efficiently organize and manage their tasks. The application provides essential features such as:

- Task Management: Add tasks with priority levels, edit existing tasks, and delete tasks effortlessly.
- Search and Filter: Search tasks by name and filter them based on their status (All, Completed, Pending).
- Selection Highlighting: Highlight tasks visually when they are selected for better focus and interaction.

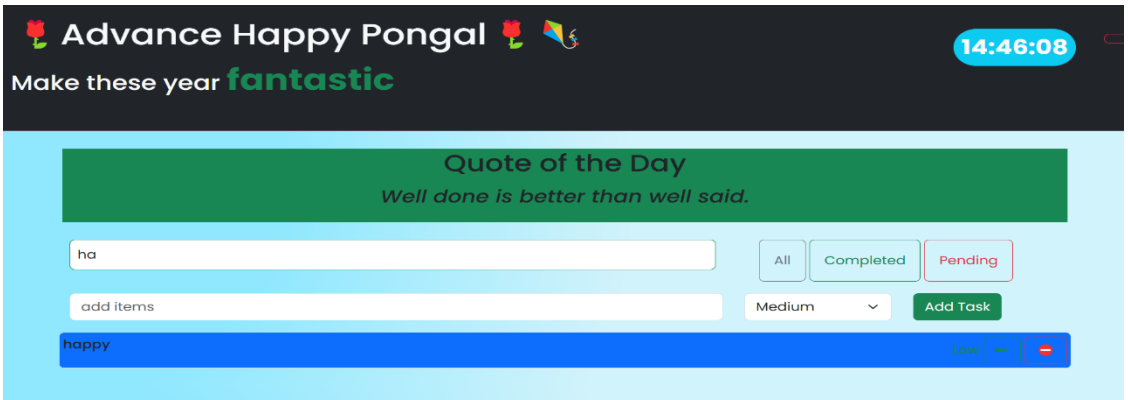
Features:

1)Task Status Management:

- Allow users to update the status of a task (e.g., mark as Pending, In Progress, or Completed).
- Automatically move tasks between status categories for better organization.
- Visually indicate task statuses with distinct styles or icons for quick recognition.

2) Search Functionality:

- Implement real-time search functionality to find tasks based on keywords.



3) **Filter Functionality:**

Filter tasks by:

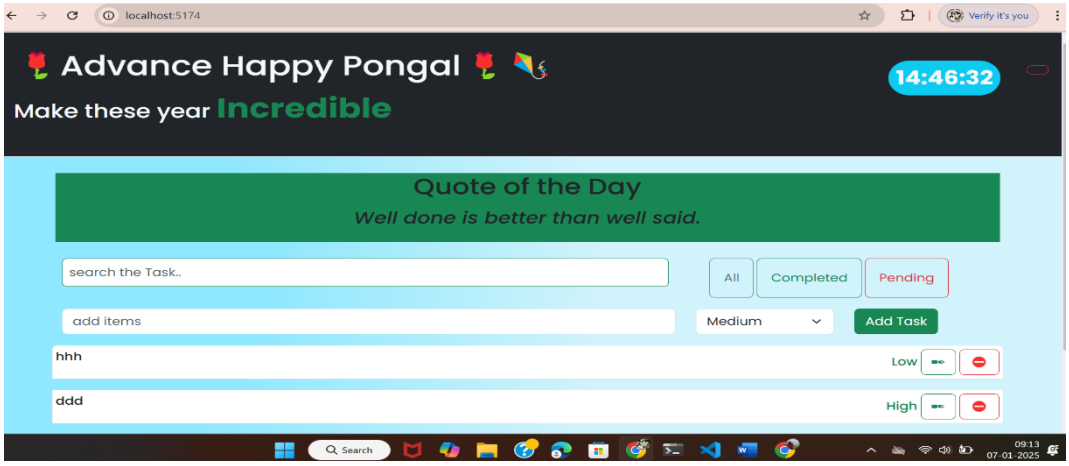
i. **All** tasks.

ii. **Completed** tasks.

iii. **Pending** tasks.

4)UI Highlights:

- Enable users to click on a task to highlight it, making it stand out for better focus



5) Toast Notifications:

- Display real-time notifications or alerts for actions such as successfully adding, editing, or deleting tasks.
- Show error messages with clear explanations if an action fails, ensuring a user-friendly experience.

Technologies Used:

Frontend:

- **React.js**: Core library used to build a dynamic and interactive user interface.
- **React Hooks**: Utilized for managing state (useState) and handling side effects (useEffect) efficiently.

- **Bootstrap:** For creating a modern, responsive, and visually appealing design with pre-built components.
- **React-Toastify**: Enables seamless integration of toast notifications for real-time user feedback.
- **Axios**: Simplifies HTTP requests for interacting with the backend REST API to manage tasks.

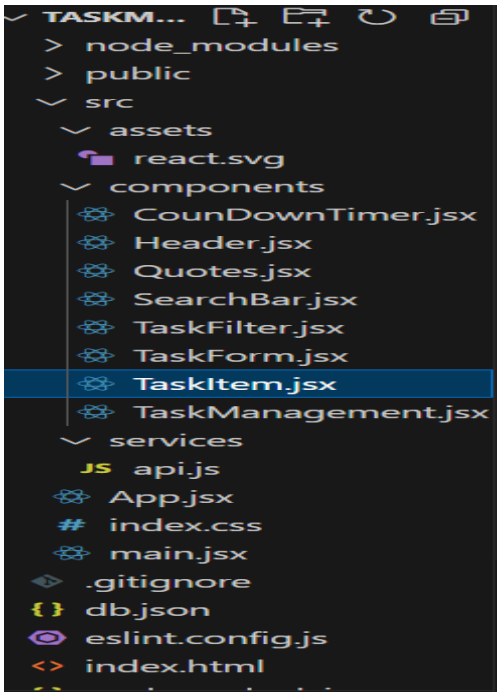
Backend:

A **RESTful API** facilitates seamless task management, providing endpoints to handle CRUD operations:

- **GET**: Retrieve the list of tasks from the server.
- **POST**: Add new tasks to the database.
- **PUT**: Update existing task details, such as description or priority.
- **DELETE**: Remove tasks from the server.

Project Structure:

- The project is structured into well-defined components, ensuring scalability and ease of maintenance.
- Each component is designed to handle a specific functionality (e.g., Task List, Task Form, Search Bar), promoting reusability across the application.
- This approach simplifies debugging, testing, and future enhancements.



Components:

1. **TaskManagement.jsx:**

- The central component that integrates all child components.
- Manages the global state of tasks and application-wide logic.
- Fetches and updates tasks by interacting with the API.
- Coordinates task operations such as adding, editing, deleting, searching, and filtering.

2. TaskItem.jsx:

- Represents each task in the task list.
- Provides functionality for editing, deleting, and highlighting individual tasks.
- Displays task details like description, priority, and status.

3.Quotes.jsx:

- Represents the Quotes in the components.
- It changes the Quote every time when we open the app.

3. TaskForm.jsx:

- Renders the form for adding a new task.
- Allows users to enter task information such as description and priority level.
- Handles form submission and validation.

4. SearchBar.jsx:

- A dedicated search bar component that filters tasks by name in real time.
- Updates the task list dynamically as users type their search query.

5. TaskFilter.jsx:

- Provides filter options to categorize tasks based on their status (All, Completed, Pending).
- Allows users to switch between different task views for better organization.

6. Api.js

- Contains an Axios instance configured for making API requests.
- Centralizes the logic for interacting with the REST API, handling GET, POST, PUT, and DELETE requests.

Conclusion:

In conclusion, a Task Management Application built using React provides a powerful, efficient, and scalable solution for managing tasks. By leveraging React's component-based architecture, the application can easily handle complex interactions like adding, editing, deleting, searching, and filtering tasks in a highly modular and reusable manner. The use of hooks like useState and useEffect enhances state management and ensures smooth UI updates.

Integrating with a RESTful API allows tasks to be stored, retrieved, and manipulated from a backend, ensuring data persistence and synchronization across sessions. React components

like TaskManagement, TaskItem, TaskForm, and others are designed to work cohesively to create a user-friendly and intuitive interface. The integration of real-time features, such as task search and status filtering, further improves the application's usability.

Final Output: