

TaskManagement App Using React.js

Introduction

Our Task Management App is a simple and effective tool to help you organize and stay on top of your daily tasks. It's built using React, which makes the app fast, reliable, and interactive.

Features

- **Task Creation:** Easily add and edit tasks with custom details.
- **Filtering & Searching:** Quickly locate tasks using advanced filters and a responsive search bar.
- **Real-Time Feedback:** Instant notifications for task updates using Toast notifications.
- **Interactive UI:** A clean, responsive interface designed with React's component architecture for seamless navigation.
- **Dynamic State Management:** Components like TaskManager, TaskItem, and TaskFilter dynamically handle user actions with React's state and props.
- **Dark & Light Mode:** Built-in theme switcher for enhanced accessibility and user preference.
- **Flexibility:** Easily scalable and maintainable code, designed for future enhancements.

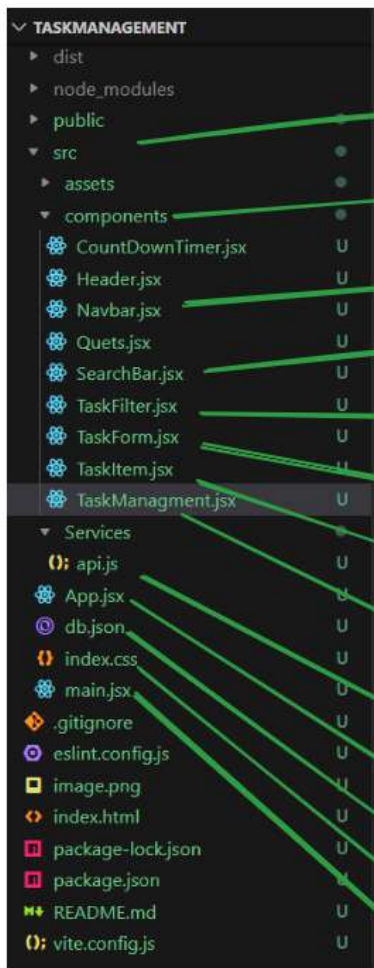
Homepage Design

The homepage of your Task Management App is designed to provide a clean, intuitive interface that welcomes users while offering quick access to key features. Below is a detailed description tailored for your current setup, including a Navbar with Bootstrap and a dark/light mode toggle.

Technologies Used

- **React.js:** A JavaScript library for building dynamic, interactive user interfaces with reusable components.
- **Bootstrap:** A front-end framework for designing responsive, mobile-first web pages with pre-built components.
- **React Toastify:** A library for displaying customizable toast notifications in React apps.

- **Bootstrap Icons:** A collection of vector icons that integrate seamlessly with Bootstrap, customizable via CSS.
- **Confetti:** A JavaScript library for generating fun, celebratory confetti animations.
- **Axios:** A promise-based HTTP client for making API requests and handling responses in JavaScript.
- **JSON Server:** A fast and easy tool for setting up a fake REST API from a JSON file for development purposes.



The src folder in a React app contains all the source code

The components folder in a React app stores reusable UI elements and logic that make up the application's interface.

The Navbar component provides a navigation bar for the app

The SearchBar.jsx component provides a search input field that allows users to filter or search for tasks or other items in the application.

The TextFilter.jsx component allows users to filter tasks based on specific keywords or text input, helping them narrow down the displayed tasks.

The TaskForm.jsx component provides a form for users to input and submit new tasks, allowing task creation or updates in the application.

The TaskItem.jsx component displays individual tasks with options to edit, mark as completed, or delete, representing each task in the task list.

The TaskManagement.jsx component serves as the main container for managing tasks, displaying the list of tasks, and handling interactions like adding, editing, or deleting tasks.

The api.js file inside the services folder is used to define and manage API requests using libraries like Axios, to interact with a backend or mock API.

The App.jsx component serves as the root component that initializes the React app, handles routing between different pages, and includes global states and main layout elements like the Navbar.

The db.json file is used in conjunction with JSON Server to simulate a backend by storing mock data

The index.css file contains global styles that are applied throughout the entire application,

The main.jsx file is the entry point of a React application