Brief Documentation for the Task Manager Project

Overview:

The Task Manager project is a simple web application built with Vue.js and styled with SCSS. It allows users to add, edit, delete, and mark tasks as complete or incomplete.

Key Features:

- Task Input: Users can input tasks in the provided input field and add them to the list.
- Edit and Delete: Tasks can be edited or deleted individually.
- Completion Status: Users can mark tasks as complete or incomplete by toggling checkboxes.
- Dynamic Editing: Editing functionality allows for seamless modification of task content.
- Responsive Design: The application is designed to be responsive, adapting to various screen sizes.

Design Choices and Rationale:

- Color Scheme: The color scheme primarily consists of blue and white tones, providing a clean and professional look. Blue is commonly associated with productivity and calmness, which aligns well with a task management application.
- Typography: The 'Montserrat' font from Google Fonts is chosen for its readability and modern aesthetic.
- Layout: The layout is designed to be intuitive, with the task input field prominently displayed at the top and task items listed below. Buttons for common actions (add, edit, delete) are strategically placed for easy access.
- Task Items: Each task item is displayed as a list element with options to mark as complete, edit, or delete. Completed tasks are visually differentiated with a strike-through effect.
- Responsive Design: Media queries are used to adjust the layout for smaller screens, ensuring a seamless user experience across devices.
- Transition Effects: Transition effects are applied for smooth visual changes, enhancing the overall user experience.

Development Choices:

- Vue.js: Vue.js is chosen for its simplicity, reactivity system, and ease of integration with other libraries or frameworks.
- SCSS: SCSS is used to write more maintainable and organized CSS code, utilizing variables, mixins, and nested rules for enhanced styling capabilities.
- Transitions: Vue.js transition components are utilized for animating task items, providing a polished look and feel to the application.
- Computed Properties: Computed properties are used to calculate and display the total number of tasks and the number of completed tasks dynamically.
- Event Handling: Event handling is implemented using Vue.js directives (@submit, @click) to manage user interactions with the application.
- Conditional Rendering: Conditional rendering is employed to display different UI components based on user actions, such as editing a task or adding a new task.

Conclusion:

The Task Manager project aims to provide users with a user-friendly interface for managing their tasks efficiently. By incorporating intuitive design, responsive layout, and seamless functionality, the application offers a hassle-free task management experience for users.