To-Do List Application Documentation

Overview

This To-Do List application is a simple, responsive web app designed to help users manage their daily tasks. It features multiple customizable themes, task creation, completion marking, and deletion functionality, with smooth animations and accessibility considerations.

Features

- **Add Tasks:** Users can add new to-do items via an input field.
- **Mark as Completed:** Clicking on a task toggles its completion status, visually indicated by strikethrough text.
- **Delete Tasks:** Each task includes a delete button to remove it from the list.
- **Theme Switching:** Multiple themes (Default, Dark, Sunset, Cyberpunk) can be toggled dynamically.
- **Responsive Design:** The layout adjusts for smaller screen sizes.
- **Accessibility:** Keyboard focus styles and semantic HTML improve accessibility.
- **Tooltips:** Hover or focus on elements shows helpful tooltips.
- **Smooth Animations:** Entrance animation for the app and hover/focus transitions.

Technologies Used

- **HTML5**: Semantic markup structure.
- **CSS3**: Styling, theming, animations, and responsive design.
- **JavaScript**: Application logic, event handling (not included here, but assumed).
- **CSS Variables:** For theme management and easy color adjustments.

CSS Design Breakdown

Theming

- **CSS Variables** defined in :root and overridden in theme classes like .theme-dark, .theme-sunset.
- Allows dynamic switching by changing the class on <body>.

Layout and Styling

- **#app Container**: Centrally positioned card with shadow and rounded corners.

- **Input and Buttons**: Full width, with accent color borders and backgrounds, smooth hover and focus transitions.
- **Task List (ul and li)**: Styled list with hover effects, completed state styling, and delete buttons.
- **Animations**: slide In keyframe animates app container on load.
- **Responsive**: Adjusts width and padding for small screens.

File Structure

```
todo-list-mvc/
 ├── index.html
 ├── style.css
 ├── model/
|   └── todoModel.js
 ├── view/
|   └── todoView.js
└── controller/
    └── todoController.js
```

Usage Instructions

Adding a Task

1. Type your task description into the input field.
2. Press Enter or click the "Add" button.
3. The task appears in the list below.

Marking Task as Completed

- Click on a task item to toggle its completed state.
- Completed tasks are displayed with a line-through and dimmed color.

Deleting a Task

- Click the red "X" delete button next to a task.
- The task is immediately removed from the list.

Switching Themes

- Click the theme toggle button at the top right corner.
- The theme cycles through the available options, updating colors and backgrounds.

Accessibility Features

- **Keyboard Navigation**: Focus outlines on inputs, buttons, and tasks for clear navigation.
- **Tooltips**: Provide additional information on hover/focus.
- **Color Contrast**: High contrast colors in dark and light themes for readability.

Future Enhancements

- **Persistent Storage:** Save tasks locally using localStorage to retain data on reload.
- **Task Editing:** Enable inline editing of existing tasks.
- **Task Priorities and Deadlines:** Allow users to set priority levels or due dates.
- **Drag and Drop Sorting:** Rearrange tasks by dragging.
- **Animations on Delete:** Smooth fade-out when removing tasks.
- **User Authentication:** Save tasks to user accounts for cloud sync.

Credits

Designed and developed by :

- Paras Verma
- Rahul Tripathi
- Jeet Ashokbhai Sojitra
- Unni Krishna Prasad Endla