To-Do List Web Application

Overview

The **To-Do List Web Application** is a simple and interactive tool designed to help users manage their daily tasks efficiently. With an intuitive and user-friendly interface, users can add, edit, delete, and mark tasks as completed or pending, ensuring better task management and productivity.

Features

- **Task Management**: Users can add new tasks, edit existing ones, or remove completed tasks.
- **Task Status Tracking**: Mark tasks as completed or pending to keep track of progress.
- **Responsive Design**: The application adapts to both mobile and desktop screens for a seamless user experience.
- **Clear and Simple UI**: A minimalistic design ensures focus remains on task management.

Prerequisites

- A modern web browser (Google Chrome, Firefox, Safari, etc.).
- Internet connection (only required for initial loading if hosted online).

Installation

- Clone the repository:

  1) git clone https://github.com/yourusername/todo-list-app.git

1. Navigate to the project directory:
   cd todo-list-app
2.
3. Open `index.html` in any modern web browser to get started.

How to Use

1. **Open the App**: Launch the application in your web browser.
2. **Add a Task**: Enter a task in the input field and click the "Add" button.
3. **Edit a Task**: Click on an existing task to modify its text and press "Save."
4. **Delete a Task**: Click the delete button next to a task to remove it.
5. **Mark as Completed**: Click on the checkbox to mark the task as completed or uncheck it to mark it as pending.

Code Walkthrough

HTML

Input Field: Allows users to enter new tasks.

```
<input type="text" id="taskInput" placeholder="Enter a task">
```

-

Button: Adds the task to the list.

```
<button onclick="addTask()">Add Task</button>
```

-

CSS

Styling: Provides a clean and modern design.

```
body {
    font-family: Arial, sans-serif;
    background: #f4f4f4;
    color: #333;
}
```

-

JavaScript

Add, Edit, and Delete Tasks: Handles task management.

```
function addTask() {
    let taskText = document.getElementById("taskInput").value;
    let taskList = document.getElementById("taskList");
    let taskItem = document.createElement("li");
    taskItem.innerHTML = `${taskText} <button onclick="removeTask(this)">Delete</button>`;
    taskList.appendChild(taskItem);
}
```

-

Future Enhancements

- Implement local storage to save tasks even after the browser is closed.
- Add categories for better task organization.
- Introduce due dates and reminders for tasks.
- Implement drag-and-drop functionality for task prioritization.

License