GitHub Copilot

PROJECT

- Creating a complete detailed project using GitHub Copilot in JavaScript is a great way to leverage its capabilities. Let's build a simple **To-Do List Application**. This project will cover several key aspects: setting up a project, implementing features, and using GitHub Copilot for code generation and guidance.

- **Project Overview**
- **To-Do List Application**
- **Features:**
  - Add new tasks
  - Mark tasks as completed
  - Delete tasks
  - Filter tasks (all, completed, active)

Step-by-Step Guide

Step 1: Set Up the Project

- **Create a Project Directory**

```
mkdir todo-list-app
cd todo-list-app
```

- **Initialize a Git Repository**

```
git init
```

Create HTML, CSS, and JavaScript Files

```
touch index.html styles.css script.js
```

Set Up Basic HTML Structure Open `index.html` and add the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="styles.css">
    <title>To-Do List</title>
</head>
```

```
<body>
    <div class="container">
        <h1>To-Do List</h1>
        <input type="text" id="taskInput" placeholder="Add a new task">
        <button id="addTaskBtn">Add Task</button>
        <ul id="taskList"></ul>
        <button id="filterAll">All</button>
        <button id="filterCompleted">Completed</button>
        <button id="filterActive">Active</button>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

Style the Application

- **Add Basic Styles** Open `styles.css` and add some basic styling:

```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
.container {
    background: white;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
```

```
input {
    padding: 10px;
    width: 70%;
    margin-right: 10px;
}
button {
    padding: 10px;
}
ul {
    list-style: none;
    padding: 0;
}
```

```
li {
    display: flex;
    justify-content: space-between;
    padding: 10px;
    border-bottom: 1px solid #ccc;
}
```

Implement JavaScript Functionality

- **Add Event Listeners** Open `script.js` and start by adding event listeners for adding tasks.

```
document.getElementById('addTaskBtn').addEventListener
('click', addTask);
```

Implement the addTask Function

- Use a comment to guide Copilot.

```
// Function to add a new task
function addTask() {
    const taskInput =
document.getElementById('taskInput');
    const taskText = taskInput.value.trim();
    if (taskText === '') return; // Do nothing if
input is empty

    // Create a new list item
    const li = document.createElement('li');
    li.textContent = taskText;
```

```
// Create a delete button
    const deleteBtn = document.createElement('button');
    deleteBtn.textContent = 'Delete';
    deleteBtn.addEventListener('click', () => {
        li.remove(); // Remove task when delete button is clicked
    });

    li.appendChild(deleteBtn);
    document.getElementById('taskList').appendChild(li);
    taskInput.value = ''; // Clear input field
}
```

Mark Tasks as Completed

- Modify the addTask function to allow marking tasks as completed.

```
li.addEventListener('click', () => {
    li.classList.toggle('completed');
});
```

Filter Tasks

Implement the filtering functionality.

```
document.getElementById('filterAll').addEventListener('click', () => {
    const tasks = document.querySelectorAll('#taskList li');
    tasks.forEach(task => task.style.display = 'flex');
});
```

```
document.getElementById('filterCompleted').addEventListener('click', () => {
    const tasks = document.querySelectorAll('#taskList li');
    tasks.forEach(task => {
        if (task.classList.contains('completed')) {
            task.style.display = 'flex';
        } else {
            task.style.display = 'none';
        }
    });
});
```

```
document.getElementById('filterActive').addEventListener('click', () => {
    const tasks = document.querySelectorAll('#taskList li');
    tasks.forEach(task => {
        if (!task.classList.contains('completed')) {
            task.style.display = 'flex';
        } else {
            task.style.display = 'none';
        }
    });
});
```

Add CSS for Completed Tasks

- Update `styles.css` to add styling for completed tasks.

```
.completed {
    text-decoration: line-through;
    color: gray;
}
```

Testing and Debugging

- **Test Functionality**
  - Open `index.html` in a browser and test adding, deleting, and filtering tasks.

- **Debugging with GitHub Copilot**
  - If you encounter issues, write comments to guide Copilot in generating debugging code or logging errors.

Version Control

- **Commit Your Changes**

```
git add .
git commit -m "Initial commit: To-Do List app with basic functionality"
```

- **Push to GitHub (Optional)**
  - Create a repository on GitHub and follow the instructions to push your local repository.

Future Enhancements

Consider adding more features:

- **Persistent Storage**: Use `localStorage` to save tasks.
- **Edit Tasks**: Allow users to edit existing tasks.
- **Due Dates**: Add the ability to assign due dates to tasks.

NOTE :

This To-Do List Application is a straightforward project to practice using GitHub Copilot alongside JavaScript. By leveraging comments and prompts, you can guide Copilot to assist with coding efficiently. As you continue to build and enhance the project, you can explore more advanced features and functionalities, deepening your understanding of both JavaScript and GitHub Copilot's capabilities.