My Conversation with LLM’s about To Do List App

A To-Do List App is a simple application that allows users to create, manage, and track tasks. It typically includes features such as adding tasks, marking them as completed, deleting tasks, and organizing them based on priority or deadlines.

Program: Build a To-Do List App (HTML, CSS, JavaScript)

A simple interactive app where users can add and remove tasks.

Step 1: HTML (Structure)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>To-Do List</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h2>To-Do List</h2>
    <input type="text" id="taskInput" placeholder="Enter a task">
    <button onclick="addTask()">Add</button>
    <ul id="taskList"></ul>
  </div>
  <script src="script.js"></script>
```

```
</body>
</html>
```

Explanation:

- Creates a simple UI with an input field, button, and list to display tasks.
- Links CSS for styling and JavaScript for functionality.

Step 2: CSS (Styling)

```
body {
   font-family: Arial, sans-serif;
   display: flex;
   justify-content: center;
   align-items: center;
   height: 100vh;
   background-color: #f0f0f0;
}

.container {
   background: white;
   padding: 20px;
   border-radius: 10px;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   text-align: center;
}
```

```
input {
   padding: 8px;
   width: 60%;
}

button {
   padding: 8px;
   cursor: pointer;
}

ul {
   list-style: none;
   padding: 0;
}

li {
   margin: 5px 0;
   padding: 8px;
   background: #ddd;
   display: flex;
   justify-content: space-between;
}
```

Explanation:

- Uses flexbox to center content.

- Adds a simple box-shadow for a modern look.
- Styles the input, button, and list items.

Step 3: JavaScript (Functionality)

```
function addTask() {
    let taskInput = document.getElementById("taskInput");
    let taskList = document.getElementById("taskList");

    if (taskInput.value.trim() !== "") {
        let li = document.createElement("li");
        li.textContent = taskInput.value;

        let deleteBtn = document.createElement("button");
        deleteBtn.textContent = "❌";
        deleteBtn.onclick = function () {
            li.remove();
        };

        li.appendChild(deleteBtn);
        taskList.appendChild(li);
        taskInput.value = "";
    }
}
```

Explanation:

- Reads user input and creates a new list item (li).
- Adds a delete button (❌) to remove tasks.
- Uses trim() to prevent adding empty tasks.

How It Works (Step-by-Step)

1. User types a task in the input box.
2. Clicks "Add" → The task appears in the list.
3. Each task has a ❌ button → Clicking it removes the task.