Front-End Job Simulation: To-Do List Application

Project Name: **Interactive To-Do List Component**

Internship Role: Front-End Development Intern

Tech Stack: HTML, CSS, JavaScript (No frameworks or libraries)

Project Goal

Create an interactive, browser-based to-do list application that allows users to manage daily tasks. This widget will form a key part of a productivity dashboard, offering users the ability to organize their tasks quickly and efficiently.

Core Features to Implement

You will build the following functionalities using only **vanilla HTML, CSS, and JavaScript**:

- **Add Task:** Users can enter a task and click "Add" to insert it into the list.
- **Edit Task:** Each task can be edited inline.
- **Delete Task:** Users can remove tasks from the list individually.
- **(Optional) Mark Complete:** Allow users to strike through completed tasks.

UI Design Guidelines

- Clean and simple layout using semantic HTML.
- Tasks should appear in **alternating colored rows** for better visibility.
- Buttons must be styled distinctly:
  - **Edit button:** Blue
  - **Delete button:** Red
- The task container should be **scrollable** when the list gets long.
- (Optional) Use icons or emojis for the edit and delete actions.

Technical Requirements

- Use **DOM methods** like document.getElementById() to access elements.
- Use addEventListener() for handling user events like button clicks.
- Create, update, and remove elements dynamically via JavaScript.
- Keep your JavaScript code **modular** using functions.
- Include **comments** in the script to explain important parts, especially the logic for editing and deleting tasks.

Extra Features (Bonus - Optional)

- Add a **"Clear All"** button to remove all tasks.
- Show a **task counter** (e.g., "3 tasks remaining").
- Allow pressing the **Enter key** to add tasks (instead of clicking the button).
- Use **icons or emojis** to make UI more intuitive.

What You Need to Submit

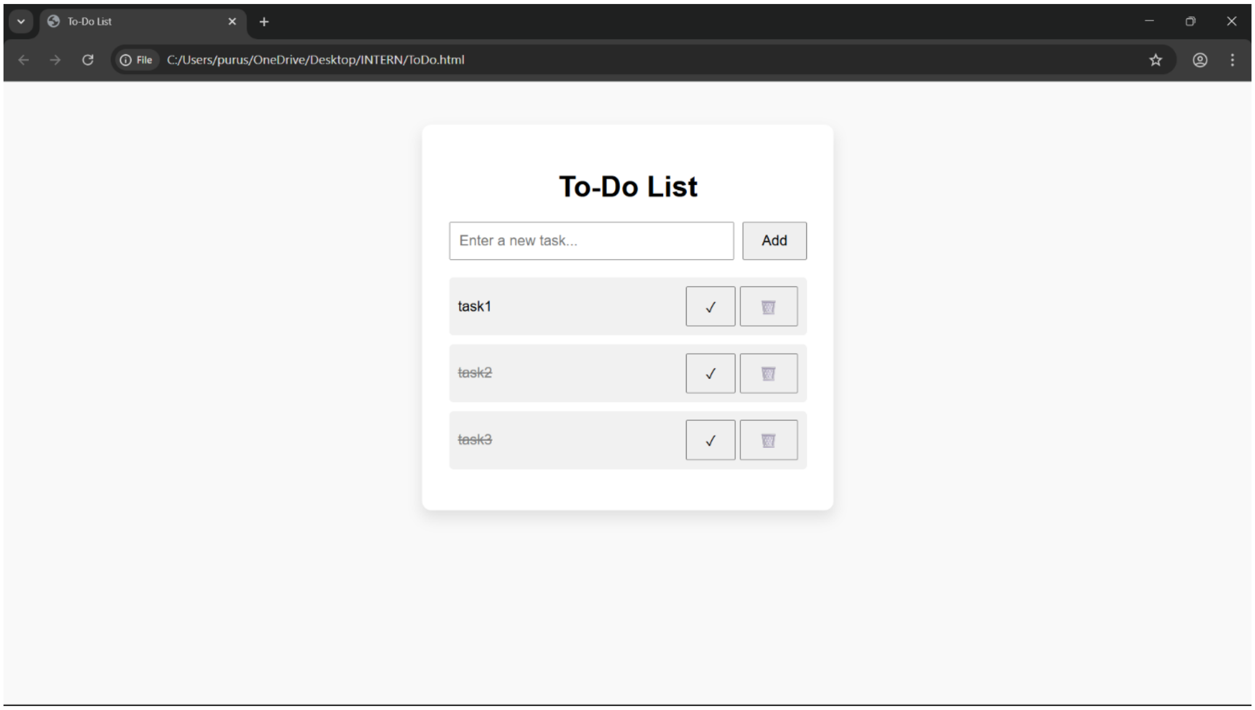
Make sure your project folder includes:

1. **index.html** — Contains the layout and structure of your to-do list.
2. **style.css** — CSS file for styling the list, buttons, and scroll behavior.
3. **script.js** — JavaScript code managing all dynamic interactions.

By completing this project, you'll gain hands-on knowledge of:

- Real-time DOM manipulation with JavaScript
- Handling user interactions with events
- Building responsive and scrollable interfaces
- Writing clean, organized, and functional front-end code

Project



To-Do List

task1

~~task2~~

~~task3~~