Task 2: Build a To-Do List Web App (Front-end only, using Vanilla JavaScript)

- **Objective: Create a dynamic To-Do list where users can add, remove, and mark tasks complete.**
- **Tools: VS Code, Chrome Browser, live-server.**
- **Deliverables: HTML, CSS, JS files for a functional To-Do app.**

Hints/Mini Guide:

1. Setup basic HTML with input, add button, and list container.
2. Style layout with CSS for clarity.
3. Use JS to add event listeners on the Add button.
4. Append new tasks as list items dynamically.
5. Implement functionality to mark tasks complete (toggle class).
6. Add remove button for each task to delete it.
7. Test adding, completing, and deleting tasks.
8. Ensure UI updates instantly without page reload.

Outcome: : Learn DOM manipulation, event handling, and state updates with vanilla JS.

Interview Questions:

1. How do you select elements in the DOM?
2. What are event listeners?
3. Explain event delegation.
4. How do you prevent default behavior in JS?
5. What is the difference between var, let, and const?
6. How does bubbling and capturing work in events?
7. How do you add and remove classes in JS?
8. What is closure in JavaScript?
9. Explain arrow functions.
10. What is the difference between == and ===?

Key Concepts: DOM, Event Listeners, JavaScript ES6, Dynamic UI, Arrays (optional).

📤 **Submit Here:**

After completing the task, paste your GitHub repo link and submit it using the link below:

- 👉 [Submission Link]

📌 **Task Submission Guidelines**

- ⏰ **Time Window:**

You can complete the task anytime between 10:00 AM to 10:00 PM on the given day. Submission link closes at 10 :00 PM

- 🔍 **Self-Research Allowed:**

You are free to explore, Google, or refer to tutorials to understand concepts and complete the task effectively.

- 🛠️ **Debug Yourself:**

Try to resolve all errors by yourself. This helps you learn problem-solving and ensures you don't face the same issues in future tasks.

- 💸 **No Paid Tools:**

If the task involves any paid software/tools, do not purchase anything. Just learn the process or find free alternatives.

- 📁 **GitHub Submission:**

Create a new GitHub repository for each task.
Add everything you used for the task — code, datasets, screenshots (if any), and a **short README.md** explaining what you did.

📤 **Submit Here:**

After completing the task, paste your GitHub repo link and submit it using the link below:

- 👉 [Submission Link]