**Interactive To-Do List using jQuery**

**1️⃣ Functional Requirements**

These define the **features** and **behavior** of the to-do list.

**✅ 1.1 Add New Task**

* The user should be able to **enter a task** in the input field.
* Clicking the **"Add Task"** button should:
  + **Create a new task** (displayed in a list).
  + **Attach action buttons** (✅ Complete & ❌ Delete).
  + **Clear the input field** after adding the task.
* If the input field is empty, show an **alert message**:❌ "Please enter a task!"

**✅ 1.2 Mark a Task as Completed**

* Each task should have a ✅ **Complete button**.
* Clicking it should:
  + Apply a **strikethrough effect** (text-decoration: line-through).
  + Change the text color to gray (for a visual cue).
  + Clicking it **again** should remove the effect (toggle).

**✅ 1.3 Delete a Task**

* Each task should have a ❌ **Delete button**.
* Clicking it should:
  + **Remove** the task from the list.
  + Ensure deleted tasks **do not return** when adding new tasks.

**✅ 1.4 Show/Hide Task List**

* The **"Show/Hide Tasks"** button should:
  + **Toggle** the entire task list **visibility**.
  + If the list is visible, it should **hide** when clicked.
  + If the list is hidden, it should **show** when clicked.

**2️⃣ Technical Requirements**

These define the **technologies and implementation details**.

**🔹 2.1 Technologies Used**

* **Frontend:**
  + **HTML** → Structure of the page.
  + **CSS** → Basic styling (layout, colors, and text effects).
  + **jQuery (JavaScript Library)** → Handles user interactions (adding, deleting, completing tasks).
* **CDN for jQuery**:
  + The project should include **jQuery via a CDN** instead of downloading files:
  + <script src="<https://code.jquery.com/jquery-3.6.0.min.js>"></script>

**🔹 2.2 File Structure**

bash

CopyEdit

/todo-list-project

│── index.html # Main HTML file

│── script.js # Contains jQuery logic

│── style.css # Stylesheet (optional)

* index.html → Defines UI elements (buttons, input field, list).
* script.js → Handles dynamic behavior (adding, deleting, toggling tasks).
* style.css → Provides a clean design (optional but recommended).

**🔹 2.3 jQuery Usage**

* The script should use $(document).ready() to ensure:✅ Code runs only **after** the page loads.✅ Elements exist before being manipulated.
* Example:
* $(document).ready(function(){
* // jQuery logic here
* });

**🔹 2.4 Event Handling**

Each button must **respond to user actions**:

* click() → To handle clicks on **Add Task**, **Complete**, **Delete**, and **Show/Hide** buttons.
* val() → To read & reset input field values.
* append() → To add new tasks dynamically.
* remove() → To delete tasks.
* toggleClass() → To mark tasks as complete/incomplete.
* toggle() → To show/hide tasks.

**3️⃣ Optional Enhancements (Advanced Features)**

These **improve user experience** but are **not required**.

**✨ 3.1 Animate Task Addition & Removal**

* Use .fadeIn() and .fadeOut() for smooth effects.
* Example:
* $("#taskList").fadeToggle();

**✨ 3.2 Store Tasks in Local Storage**

* Saves tasks even if the user refreshes the page.
* Uses localStorage.setItem() and localStorage.getItem().

**✨ 3.3 Add Edit Feature**

* Allow users to **edit** a task instead of deleting & re-adding.