NAME: TANKISO MASOEBE

WEB DESIGN WEEK 12

1. Code: Interactive To-Do List

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Interactive To-Do List</title>
 <style>
  body {
   font-family: Arial, sans-serif;
   margin: 40px;
   background-color: #f5f5f5;
  }
  h1 {
   text-align: center;
  }
  #todo-container {
   width: 400px;
   margin: 0 auto;
   background: white;
   padding: 20px;
   border-radius: 12px;
   box-shadow: 0 0 10px rgba(0,0,0,0.1);
  }
  input {
   width: 70%;
   padding: 8px;
```

```
  font-size: 16px;
}
button {
  padding: 8px 12px;
  font-size: 16px;
  margin-left: 5px;
  cursor: pointer;
}
ul {
  list-style-type: none;
  padding: 0;
  margin-top: 20px;
}
li {
  background-color: #e9e9e9;
  padding: 10px;
  margin-bottom: 8px;
  border-radius: 6px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
.delete-btn {
  background-color: #ff4b4b;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  padding: 5px 10px;
```

```
    }
  </style>
</head>
<body>
  <div id="todo-container">
    <h1>My To-Do List</h1>

    <!-- Input field and Add button -->
    <input type="text" id="taskInput" placeholder="Enter a new task">
    <button id="addBtn">Add</button>

    <!-- Unordered list to hold all tasks -->
    <ul id="taskList"></ul>
  </div>

  <script>
    // Select DOM elements
    const addBtn = document.getElementById("addBtn");  // The "Add" button
    const taskInput = document.getElementById("taskInput");  // Input field for new tasks
    const taskList = document.getElementById("taskList");  // The <ul> that holds list items

    // Add click event listener to "Add" button
    addBtn.addEventListener("click", function() {
      // Get the text entered by the user
      const taskText = taskInput.value.trim();

      // If input is empty, alert user and stop
      if (taskText === "") {
        alert("Please enter a task!");
```

```
  return;
}

// 1 Create a new <li> element (a new list item)
const li = document.createElement("li");

// 2 Create a text node with the user's input
const taskNode = document.createTextNode(taskText);

// 3 Append the text node to the <li>
li.appendChild(taskNode);

// 4 Create a "Delete" button for this task
const deleteBtn = document.createElement("button");
deleteBtn.textContent = "Delete";  // Button text
deleteBtn.className = "delete-btn";  // Add a CSS class for styling

// 5 Add a click event to the Delete button to remove the task
deleteBtn.addEventListener("click", function() {
  taskList.removeChild(li); // Remove the clicked task from the list
});

// 6 Append the Delete button to the <li>
li.appendChild(deleteBtn);

// 7 Append the <li> (with text and delete button) to the <ul>
taskList.appendChild(li);

// 8 Clear the input field after adding the task
```

```
    taskInput.value = "";
  });
 </script>
</body>
</html>
```

2. **Event Delegation Summary**

Event delegation is a JavaScript technique that allows you to use **one event listener** to handle events for **many child elements** by taking advantage of **event bubbling**. When an event like a click happens on a child element, it automatically "bubbles up" to its parent element. By placing a single listener on the parent, you can detect which child triggered the event using `event.target`.

This method is very useful when working with **dynamic web pages**, such as a to-do list where new items are created after the page loads. Instead of adding a new event listener every time you create a "Delete" button, you can attach one listener to the parent `<ul>` and let it manage all delete actions. Event delegation makes your code **simpler**, **faster**, and **more efficient**, especially when handling a large number of elements that may change over time.

Code

```
document.getElementById("taskList").addEventListener("click", function(event) {
  if (event.target.classList.contains("delete-btn")) {
    event.target.parentElement.remove(); // Delete that task
  }
});
```