**JavaScript Assignment 1**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>

  <title>Task Manager</title>

  <link rel="stylesheet" href="style.css" />

</head>

<body>

  <div class="container">

    <h1>Task Manager</h1>

    <div class="task-input">

      <input type="text" id="taskInput" placeholder="Enter a new task" />

      <button onclick="addTask()">Add Task</button>

    </div>

    <div class="stats" id="stats">

      <!-- Dynamic stats will go here -->

    </div>

    <ul id="taskList"></ul>

  </div>

  <script src="script.js"></script>

</body>

</html>

**style.css**

body {

  font-family: Arial, sans-serif;

  background-color: #f2f2f2;

  display: flex;

  justify-content: center;

  align-items: center;

  height: 100vh;

}

.container {

  background-color: #ffffff;

  padding: 30px;

  border-radius: 10px;

  box-shadow: 0 0 10px rgba(0,0,0,0.1);

  width: 400px;

}

h1 {

  text-align: center;

  color: #333;

}

.task-input {

  display: flex;

  gap: 10px;

  margin-bottom: 20px;

}

#taskInput {

  flex: 1;

  padding: 10px;

  border: 1px solid #ccc;

  border-radius: 5px;

}

button {

  padding: 10px 15px;

  background-color: #28a745;

  color: white;

  border: none;

  border-radius: 5px;

  cursor: pointer;

}

button:hover {

  background-color: #218838;

}

ul {

  list-style: none;

  padding: 0;

}

li {

  padding: 10px;

  border-bottom: 1px solid #ddd;

  display: flex;

  justify-content: space-between;

  align-items: center;

}

li.completed span {

  text-decoration: line-through;

  color: #999;

}

li button {

  margin-left: 10px;

  background-color: #dc3545;

  border: none;

  color: white;

  border-radius: 4px;

  padding: 5px 8px;

  cursor: pointer;

}

li button.toggle {

  background-color: #007bff;

}

li button:hover {

  opacity: 0.9;

}

.stats {

  margin-bottom: 20px;

  padding: 10px;

  background-color: #e9ecef;

  border-radius: 8px;

  font-size: 14px;

  color: #333;

}

.stats span {

  display: inline-block;

  margin-right: 15px;

}

.completed-label {

  background-color: #d4edda;

  color: #155724;

  padding: 6px 10px;

  border-radius: 5px;

  font-size: 13px;

  font-weight: bold;

  border: 1px solid #c3e6cb;

  margin-right: 10px;

}

**script.js**

let tasks = [];

function renderTasks() {

  const taskList = document.getElementById('taskList');

  taskList.innerHTML = ''; // Clear list

  tasks.forEach((task, index) => {

    const li = document.createElement('li');

    li.className = task.completed ? 'completed' : '';

    const span = document.createElement('span');

    span.textContent = task.name;

    const buttonsContainer = document.createElement('div');

    if (!task.completed) {

      const completeBtn = document.createElement('button');

      completeBtn.textContent = 'Complete';

      completeBtn.classList.add('toggle');

      completeBtn.onclick = () => markComplete(index);

      buttonsContainer.appendChild(completeBtn);

    } else {

      const completedLabel = document.createElement('span');

      completedLabel.className = 'completed-label';

      completedLabel.textContent = '✅ Completed';

      buttonsContainer.appendChild(completedLabel);

    }

    const deleteBtn = document.createElement('button');

    deleteBtn.textContent = 'Delete';

    deleteBtn.onclick = () => deleteTask(index);

    buttonsContainer.appendChild(deleteBtn);

    li.appendChild(span);

    li.appendChild(buttonsContainer);

    taskList.appendChild(li);

  });

  updateStats();

}

function addTask() {

  const input = document.getElementById('taskInput');

  const taskName = input.value.trim();

  if (taskName !== '') {

    tasks.push({ name: taskName, completed: false });

    input.value = '';

    renderTasks();

  }

}

function markComplete(index) {

  tasks[index].completed = true;

  renderTasks();

}

function deleteTask(index) {

  tasks = tasks.filter((\_, i) => i !== index);

  renderTasks();

}

function updateStats() {

  const total = tasks.length;

  const completed = tasks.filter(task => task.completed).length;

  const remaining = total - completed;

  const statsDiv = document.getElementById('stats');

  statsDiv.innerHTML = `

    <span>📊 Total Tasks: <strong>${total}</strong></span>

    <span>✅ Completed: <strong>${completed}</strong></span>

    <span>🕒 Remaining: <strong>${remaining}</strong></span>

  `;

}