Problem Statement-1 To-Do List Application (DOM Manipulation Project)

**Steps and screenshot**

**HTML Setup**

* Created an input field and “Add Task” button.
* Added a section to display error messages.
* Added an empty list (<ul>) to hold tasks.

**CSS Styling**

* Styled the layout with a centered container.
* Added hover effects for buttons.
* Highlighted completed tasks with strikethrough and gray color.

**JavaScript Functionality**

**Add Task**

* On clicking “Add Task”:
  + Checks if input is empty it shows error.
  + If valid it creates a new <li> with:
    - Checkbox
    - Task text
    - Edit and Delete buttons

**Mark as Completed**

* Checkbox toggles .completed class it applies strikethrough.

**Edit Task**

* Clicking “Edit” opens a prompt.
* If new text is valid → updates the task.

**Delete Task**

* Clicking “Delete” removes the task from the list.

**Html Code**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>To-Do List App</title>

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

</head>

<body>

  <div class="container">

    <h2>To-Do List</h2>

    <div class="input-group">

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

  <button id="addTaskBtn">Add Task</button>

</div>

    <p id="errorMsg" class="error"></p>

    <ul id="taskList"></ul>

  </div>

  <script src="script.js"></script>

</body>

</html>

**Css Code**

/\*  Global Styles \*/

body {

  background-color: #c0ecf1; /\* light cyan \*/

  font-family: Arial, sans-serif;

}

/\* Container \*/

.container {

  max-width: 600px;

  margin: 0 auto; /\* centers horizontally \*/

  padding: 20px;

  background: #fff;

  border-radius: 10px;

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

}

/\*  Heading \*/

h2 {

  text-align: center;

  font-size: 24px;

  margin-bottom: 20px;

}

/\*Input Group \*/

.input-group {

  display: flex;

  justify-content: center;

  gap: 10px;

  margin-bottom: 20px;

}

#taskInput {

  width: 70%;

  padding: 10px;

  border: 1px solid #ccc;

  border-radius: 5px;

}

#addTaskBtn {

  padding: 10px 15px;

  background-color: #007bff;

  color: white;

  border: none;

  border-radius: 5px;

  cursor: pointer;

}

#addTaskBtn:hover {

  background-color: #0056b3;

}

/\*  Error Message \*/

#errorMsg {

  margin-left: 30px; /\* aligns with task list indentation \*/

  color: red;

  font-size: 0.9em;

  margin-top: 20px;

}

/\*  Task List \*/

#taskList {

  list-style-type: none;

  padding-left: 30px; /\* indent the whole list \*/

  margin-top: 20px;

}

/\* Task Item \*/

.task-item {

  background-color: #fffda4;

  border-bottom: 1px solid #ccc;

  border-radius: 8px;

  padding: 15px 10px;

  margin-bottom: 15px;

  width: 100%;

  box-sizing: border-box;

  display: flex;

  flex-direction: column;

  align-items: flex-start;

}

/\* Task Content \*/

.task-content {

  display: flex;

  align-items: center;

  gap: 10px;

  margin-bottom: 10px;

}

.task-text {

  word-break: break-word;

}

/\* Task Actions \*/

.task-actions {

  display: flex;

  gap: 10px;

}

.task-actions button {

  padding: 8px 12px;

  font-size: 0.9em;

  border: none;

  border-radius: 5px;

  cursor: pointer;

  background-color: #007bff;

  color: white;

}

.task-actions button:hover {

  background-color: #0056b3;

  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

}

.task-actions .delete-btn {

  background-color: #dc3545;

}

.task-actions .delete-btn:hover {

  background-color: #c82333;

}

/\* Completed Task \*/

.completed {

  text-decoration: line-through;

  color: gray;

  font-style: italic;

}

**JavaScript Code**

document.getElementById('addTaskBtn').addEventListener('click', addTask);

function addTask() {

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

  const taskText = input.value.trim();

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

  if (taskText === '') {

    errorMsg.innerHTML = 'Task cannot be empty!';

    return;

  }

  errorMsg.innerHTML = '';

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

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

  li.classList.add('task-item');

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

  taskContent.classList.add('task-content');

  const checkbox = document.createElement('input');

  checkbox.type = 'checkbox';

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

  span.classList.add('task-text');

  span.textContent = taskText;

  checkbox.addEventListener('change', () => {

    span.classList.toggle('completed');

  });

  taskContent.appendChild(checkbox);

  taskContent.appendChild(span);

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

  taskActions.classList.add('task-actions');

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

  editBtn.textContent = 'Edit';

  editBtn.addEventListener('click', () => {

    const newText = prompt('Edit your task:', span.textContent);

    if (newText !== null && newText.trim() !== '') {

      span.textContent = newText.trim();

    }

  });

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

  deleteBtn.textContent = 'Delete';

  deleteBtn.classList.add('delete-btn');

  deleteBtn.addEventListener('click', () => {

    taskList.removeChild(li);

  });

  taskActions.appendChild(editBtn);

  taskActions.appendChild(deleteBtn);

  li.appendChild(taskContent);

  li.appendChild(taskActions);

  taskList.appendChild(li);

  input.value = '';

}

**Website Screenshots**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Editing the task**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Checkout tasks**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Deleted tasks**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Error message when input field is empty and user clicks on add task**

**A screenshot of a computer

AI-generated content may be incorrect.**