# Lahore Garrison University

## DHA Phase 6 Lahore

## Lab Task 9

| | |
|---|---|
| **Student Name** | **Ateeb Qaiser** |
| **Student Roll No** | **Fa23/BSCS/279** |
| **Section** | **G** |
| **Subject** | **Web Technologies Lab** |
| **Instructor Name** | **Mr. M Yousaf** |

# Lab 9: DOM Manipulation — To-Do List

## Objectives:

- Create a dynamic to-do list using DOM APIs: add/remove items, mark complete, persist to localStorage.

## Tools/Tech:

- VS Code, Live Server, localStorage

## Tasks/Steps:

1. Implement add, edit, delete, toggle complete features.
2. Persist list state in localStorage and load on page start.
3. Enhance UI with CSS and keyboard accessibility (Enter to add).

## Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>TODO APP</title>
<style>
   @import
url('https://fonts.googleapis.com/css2?family=Jost:wght@300;400;500;600;700&display=swap');

   body {
      margin: 0;
      font-family: 'Jost', sans-serif;
      background: #e8e9f3;
   }

   .header {
      background: #9e9ed8;
      padding: 20px;
      color: white;
      font-size: 26px;
      font-weight: bold;
   }

   .content { padding: 20px; }

   .task-form {
      display: flex;
      margin-bottom: 20px;
   }

   .task-form input {
      flex: 1;
      padding: 12px;
```

```
    border-radius: 5px;
    border: 1px solid #ccc;
    font-size: 18px;
}

.task-form button {
    padding: 12px 20px;
    background: #9e9ed8;
    border: none;
    color: white;
    font-size: 18px;
    border-radius: 5px;
    cursor: pointer;
    margin-left: 10px;
}

#taskList {
    list-style: none;
    padding: 0;
    margin: 0;
}

.task-item {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 12px 10px;
    background: white;
    border-radius: 8px;
    margin-top: 10px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.1);
    font-size: 18px;
}

.task-title.completed {
    color: gray;
    text-decoration: line-through;
}

.button-group button {
    padding: 6px 12px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    color: white;
    margin-left: 8px;
    font-size: 14px;
}

.edit-btn { background: #007bff; }
.delete-btn { background: #dc3545; }

.footer {
    background: #fff;
    padding: 15px 0;
    position: fixed;
    bottom: 0;
    width: 100%;
    display: flex;
    justify-content: space-around;
```

```
    box-shadow: 0 -2px 5px rgba(0,0,0,0.1);
  }

  .footer button {
    padding: 8px 15px;
    border-radius: 6px;
    border: none;
    background: #9e9ed8;
    color: white;
    cursor: pointer;
    font-weight: bold;
  }
</style>
</head>
<body>

<div class="header">TODO APP</div>

<div class="content">
  <div class="task-form">
    <input type="text" id="taskInput" placeholder="Add a new task">
    <button id="addTaskBtn">Add</button>
  </div>

  <ul id="taskList"></ul>
</div>

<div class="footer">
  <button id="clearTasksBtn">Clear Tasks</button>
  <button id="completedTasksBtn">Completed Tasks</button>
  <button id="backBtn" style="display:none;">Back</button>
</div>

<script>
let tasks = JSON.parse(localStorage.getItem("tasks")) || [];
let showCompletedOnly = false;

function saveTasks() {
  localStorage.setItem("tasks", JSON.stringify(tasks));
}

function renderTasks() {
  const list = document.getElementById("taskList");
  list.innerHTML = "";

  const data = showCompletedOnly ? tasks.filter(t => t.completed) : tasks;

  if (data.length === 0) {
    list.innerHTML = `<li style="text-align:center; padding:10px;">No tasks found</li>`;
    document.getElementById("backBtn").style.display = showCompletedOnly ? "inline-block" : "none";
    return;
  }

  data.forEach((task, index) => {
    const li = document.createElement("li");
    li.className = "task-item";

    const chk = document.createElement("input");
    chk.type = "checkbox";
    chk.checked = task.completed;
```

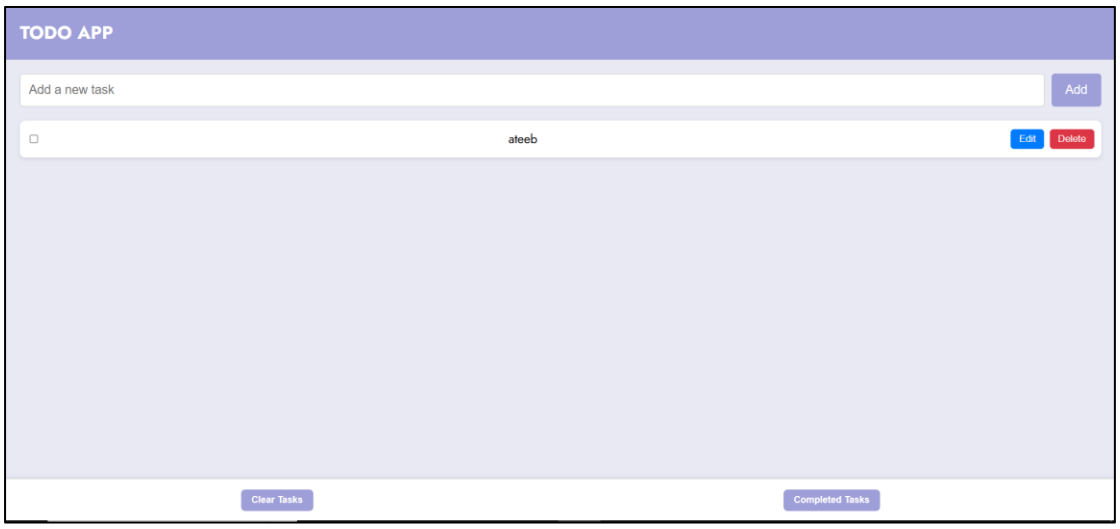
```
      chk.onclick = () => {
        task.completed = chk.checked;
        saveTasks();
        renderTasks();
      };

      const title = document.createElement("span");
      title.className = "task-title";
      title.textContent = task.title;
      if (task.completed) title.classList.add("completed");

      const btnGroup = document.createElement("div");
      btnGroup.className = "button-group";

      const edit = document.createElement("button");
      edit.textContent = "Edit";
      edit.className = "edit-btn";
      edit.onclick = () => {
        const newTitle = prompt("Edit task:", task.title);
        if (newTitle && newTitle.trim()) {
          task.title = newTitle.trim();
          saveTasks();
          renderTasks();
          alert("Task edited successfully!");
        }
      };

      const del = document.createElement("button");
      del.textContent = "Delete";
      del.className = "delete-btn";
      del.onclick = () => {
        tasks.splice(index, 1);
        saveTasks();
        renderTasks();
        alert("Task deleted!");
      };

      btnGroup.appendChild(edit);
      btnGroup.appendChild(del);

      li.appendChild(chk);
      li.appendChild(title);
      li.appendChild(btnGroup);

      list.appendChild(li);
    });

    document.getElementById("backBtn").style.display = showCompletedOnly ? "inline-block" : "none";
}

document.getElementById("addTaskBtn").onclick = () => {
    const input = document.getElementById("taskInput");
    const title = input.value.trim();
    if (!title) return alert("Please enter a task!");

    tasks.push({ title, completed: false });
    saveTasks();
    renderTasks();
    alert("Task added!");
    input.value = "";
```

```
};

document.getElementById("clearTasksBtn").onclick = () => {
  if (confirm("Delete all tasks?")) {
    tasks = [];
    saveTasks();
    renderTasks();
    alert("All tasks cleared!");
  }
};

document.getElementById("completedTasksBtn").onclick = () => {
  showCompletedOnly = true;
  renderTasks();
};

document.getElementById("backBtn").onclick = () => {
  showCompletedOnly = false;
  renderTasks();
};

renderTasks();
</script>

</body>
</html>
```

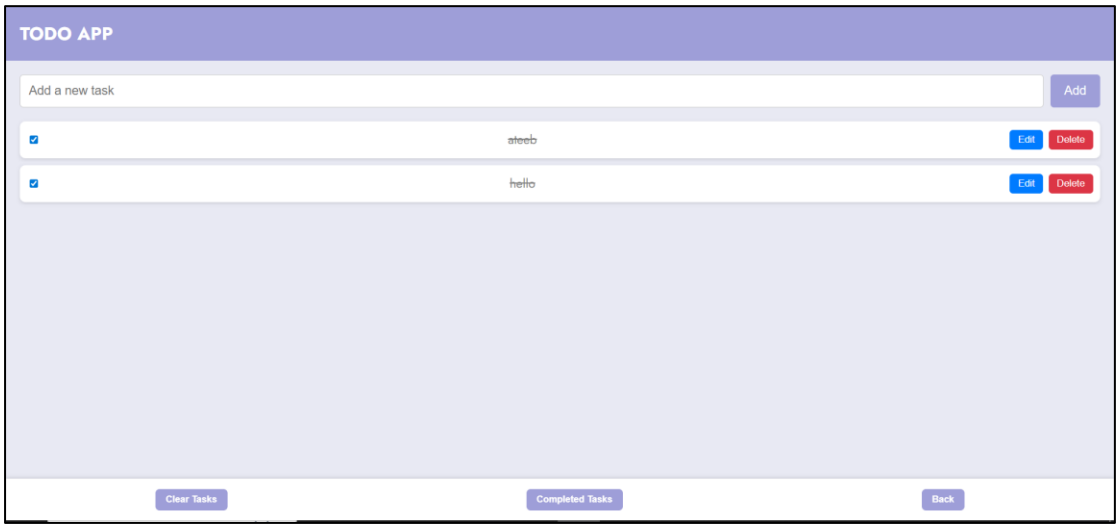
## Output:

## Home Screen:

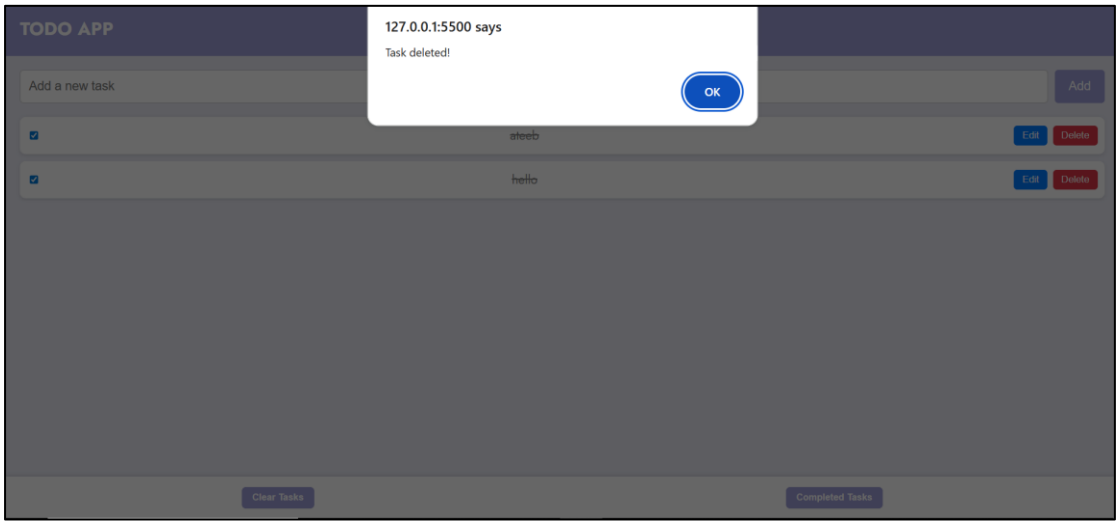
## Add Task:



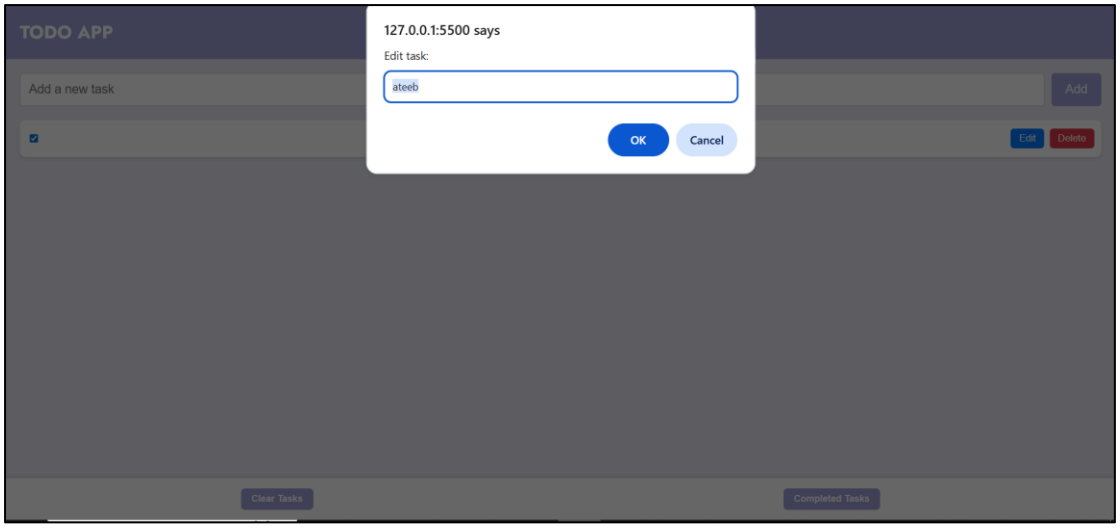

## Completed Task:



## Delete Task:

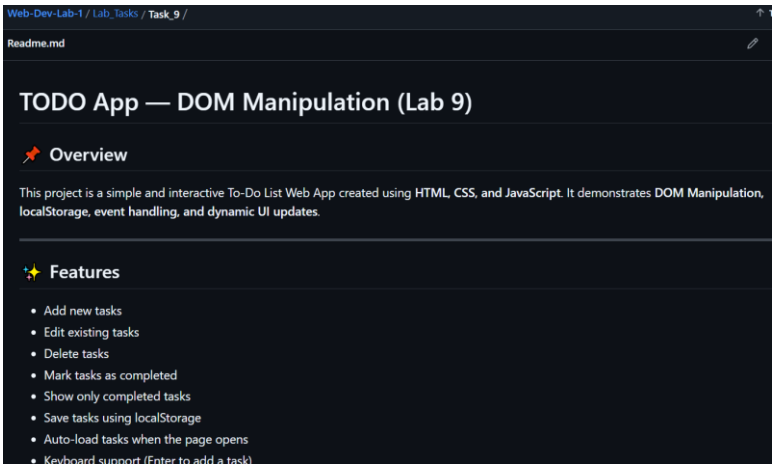



## Edit Task:



## Deliverables:

- To-do list web app hosted in repo; README describing features.




## CONCLUSION:

**In this lab, I created a functional To-Do List using DOM Manipulation with features like add, edit, delete, and mark complete.Using localStorage, tasks persist after refresh, helping me understand dynamic web pages and JavaScript DOM interaction.**

## RUBRICS:

| Performance | | | Lab Report | | |
|---|---|---|---|---|---|
| Description | Total Marks | Marks Obtained | Description | Total Marks | Marks Obtained |
| Ability to Conduct practical | 5 | | Structure | 5 | |
| Data Analysis & Interpretation | 5 | | Efficiency | 5 | |
| Total Marks obtained | | | Total Marks Obtained | | |

Instructor Signature _____