Name:Taneeshq Cholekar Date:14/4/2025

Assignment-1

Problem Statament:

Develop a Web Application using HML5, CSS3 and JavaScript and perform basic CRUD operations.

1. Create a Record

2. Read Records

3. Update a Record

4. Delete a Record

Notes-Ensure front end validations including Form, Password, Hidden fields are taken care

To demonstrate the given functions I created a simple todo list which allows user to create,read,edit and delete task. As an extra layer of security each task is associated with password. If user enters the correct password only then the task can be edited.

Source code:

**1)index.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Todo List App</title>

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

</head>

<body>

  <div class="container">

    <h1>My ToDo List</h1>

    <form id="todo-form">

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

      <input type="password" id="password" placeholder="Enter password" required minlength="4" />

      <input type="hidden" id="edit-id" />

      <button type="submit" class="submit-btn">Add Task</button>

    </form>

    <ul id="task-list"></ul>

  </div>

  <script src="script.js"></script>

</body>

</html>

**2)styles.css**

body {

    font-family: 'Segoe UI', sans-serif;

    background: #000000;

    padding: 2rem;

  }

  .container {

    width: 400px;

    margin: auto;

    background: #fff;

    padding: 1.5rem;

    border-radius: 10px;

    box-shadow: 0 0 10px #ccc;

  }

  h1 {

    text-align: center;

  }

  form {

    display: flex;

    flex-direction: column;

    gap: 10px;

    margin-bottom: 20px;

  }

  input, button {

    padding: 10px;

    font-size: 16px;

  }

  ul {

    list-style-type: none;

    padding: 0;

  }

  li {

    background: #eee;

    padding: 10px;

    margin-bottom: 10px;

    display: flex;

    justify-content: space-between;

    align-items: center;

  }

  button.edit, button.delete {

    margin-left: 5px;

    padding: 5px 10px;

  }

  .submit-btn{

    background-color: blue;

    color:white;

  }

  .edit{

    background-color: yellow;

    border-radius:20%;

    height:3rem;

    width:4rem;

  }

  .delete{

    background-color: red;

    border-radius:20%;

    height:3rem;

    width:4rem;

  }

  ul li{

    box-shadow: 10px 10px 10px black;

  }

**3)script.js**

let tasks = [];

let form = document.getElementById("todo-form");

let taskInput = document.getElementById("task");

let passwordInput = document.getElementById("password");

let editId = document.getElementById("edit-id");

let taskList = document.getElementById("task-list");

form.addEventListener("submit", function (e) {

  e.preventDefault();

  const task = taskInput.value.trim();

  const password = passwordInput.value.trim();

  const id = editId.value;

  if (task === "" || password.length < 4) {

    alert("Please enter a task and a valid password (min 4 characters)");

    return;

  }

  if (id) {

    // Editing existing task: check password

    if (tasks[id].password === password) {

      tasks[id].task = task;

      alert("Task updated successfully!");

    } else {

      alert("Incorrect password. Cannot update the task.");

      return;

    }

    editId.value = "";

  } else {

    // Creating new task

    tasks.push({ task, password });

  }

  taskInput.value = "";

  passwordInput.value = "";

  renderTasks();

});

function renderTasks() {

  taskList.innerHTML = "";

  tasks.forEach((t, index) => {

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

    li.innerHTML = `

      ${t.task}

      <div class="list-item">

        <button class="edit" onclick="editTask(${index})">Edit</button>

        <button class="delete" onclick="deleteTask(${index})">Delete</button>

      </div>

    `;

    taskList.appendChild(li);

  });

}

function editTask(id) {

  taskInput.value = tasks[id].task;

  editId.value = id;

}

function deleteTask(id) {

  if (confirm("Are you sure you want to delete this task?")) {

    tasks.splice(id, 1);

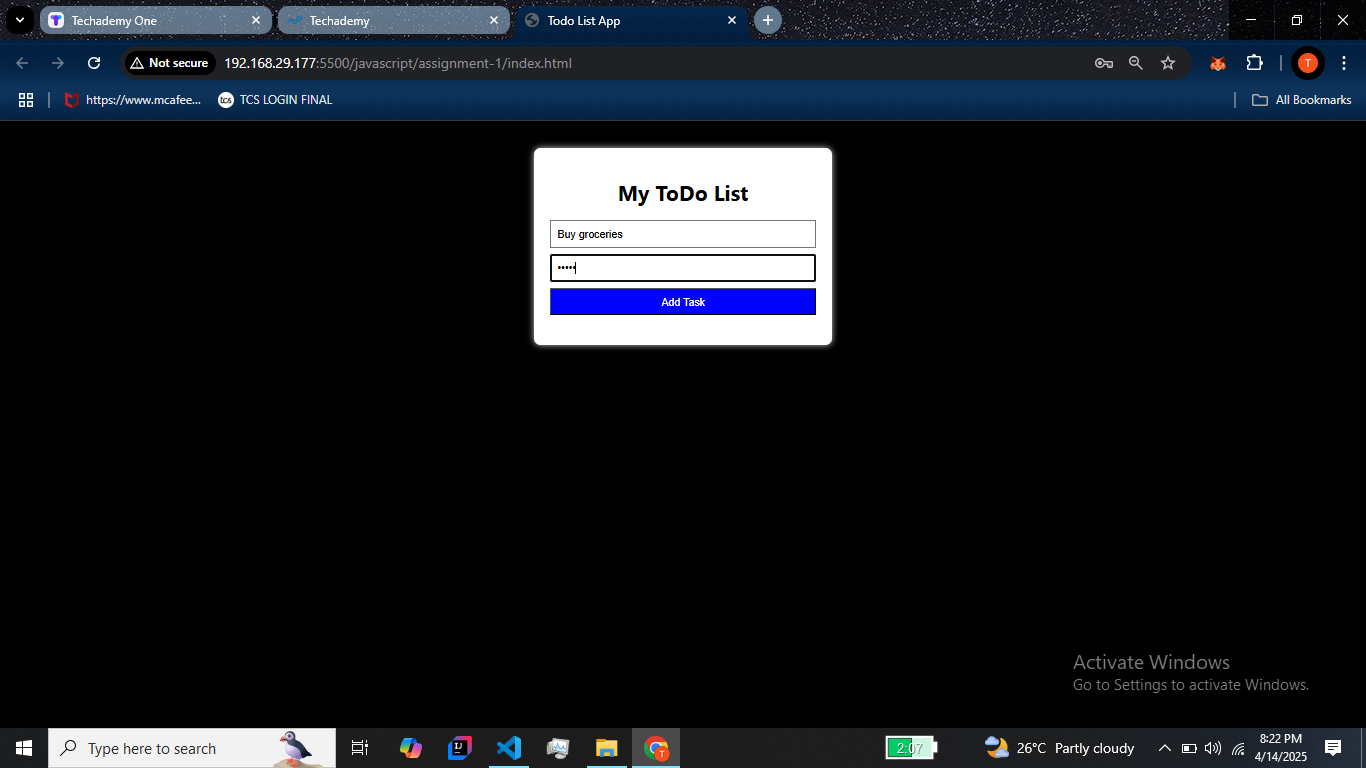
    renderTasks();

  }

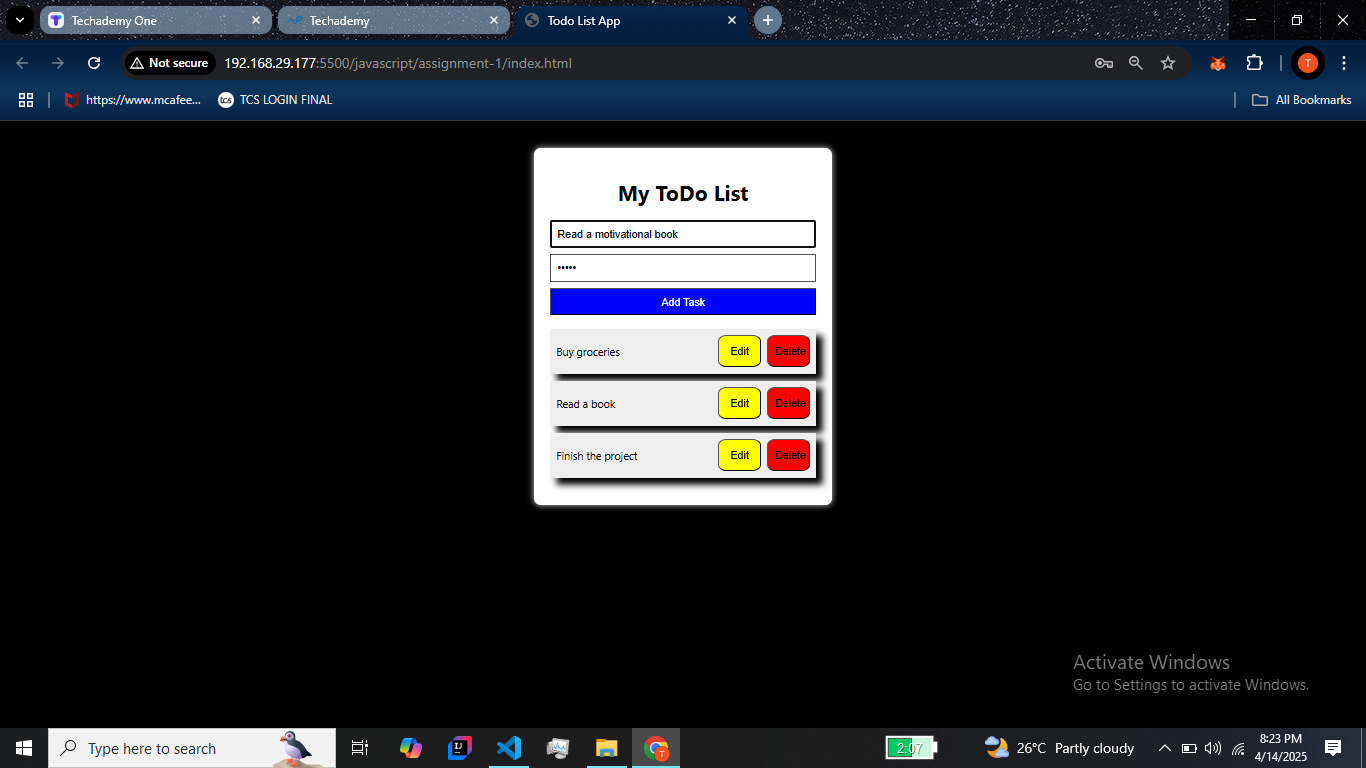
}

**4)Output**

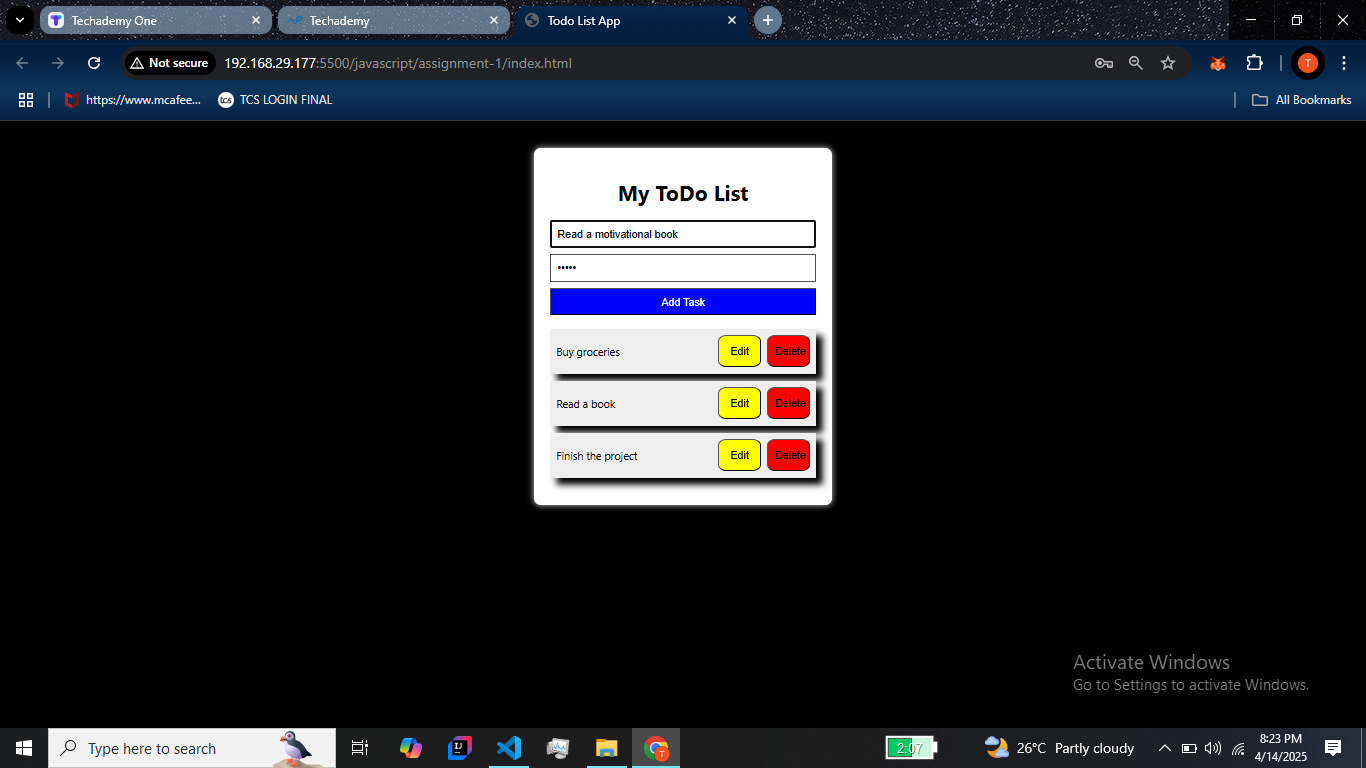
**a)Creating a task**

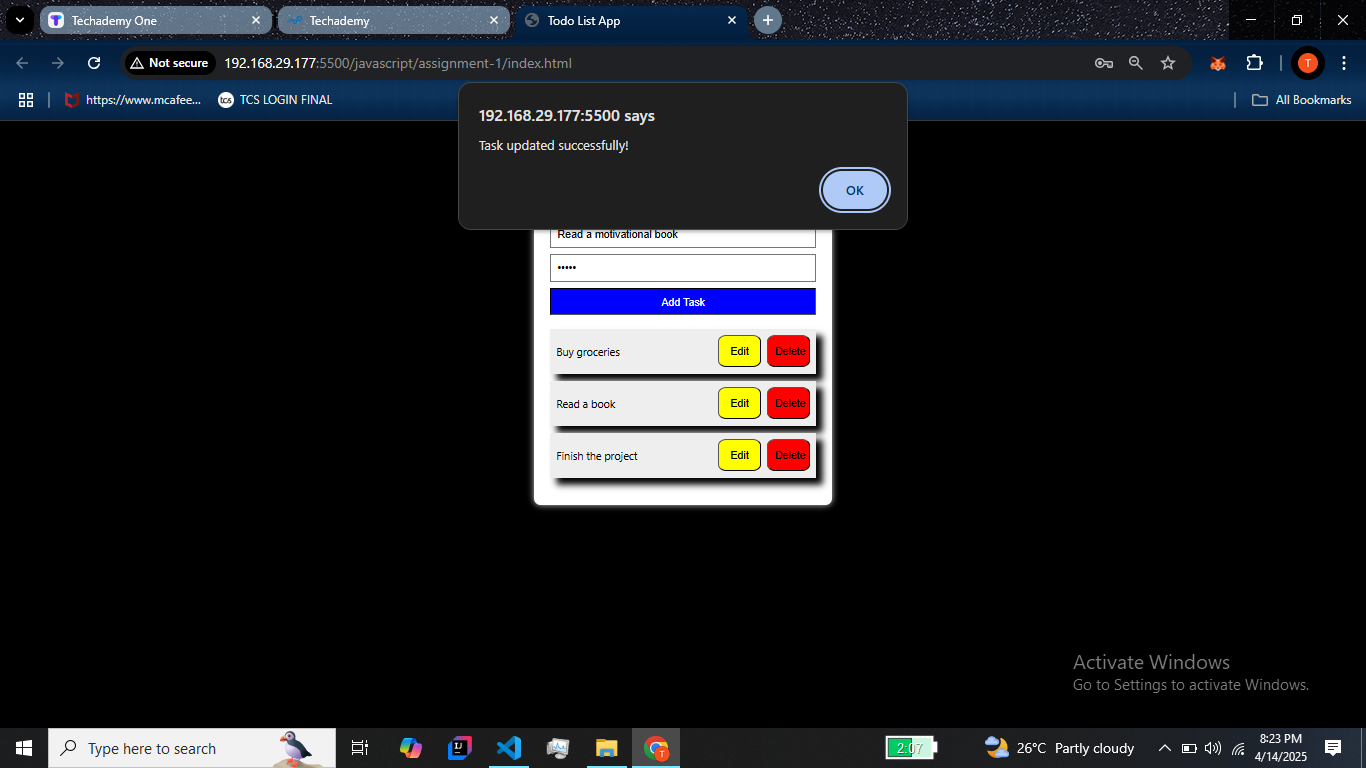
****

**b)List of all tasks**

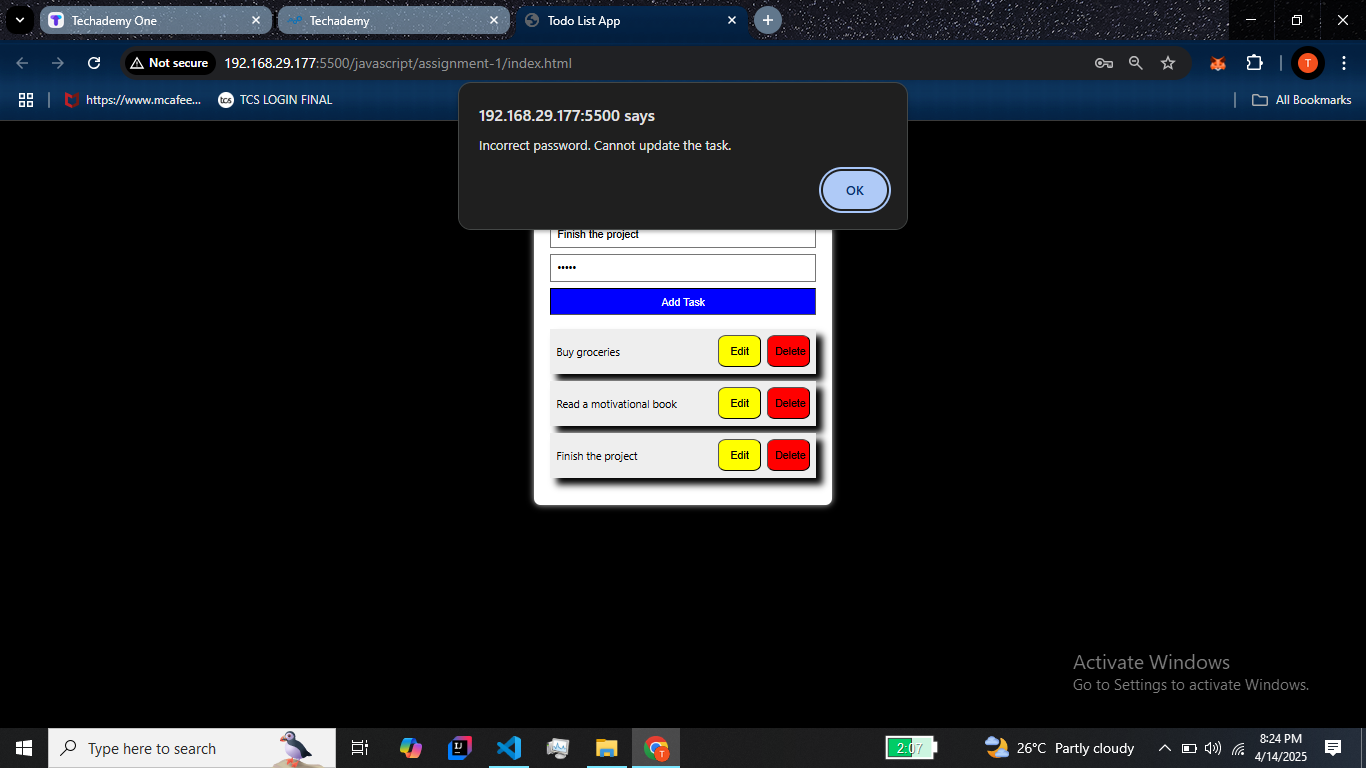
****

**c)Editing a task with correct password**

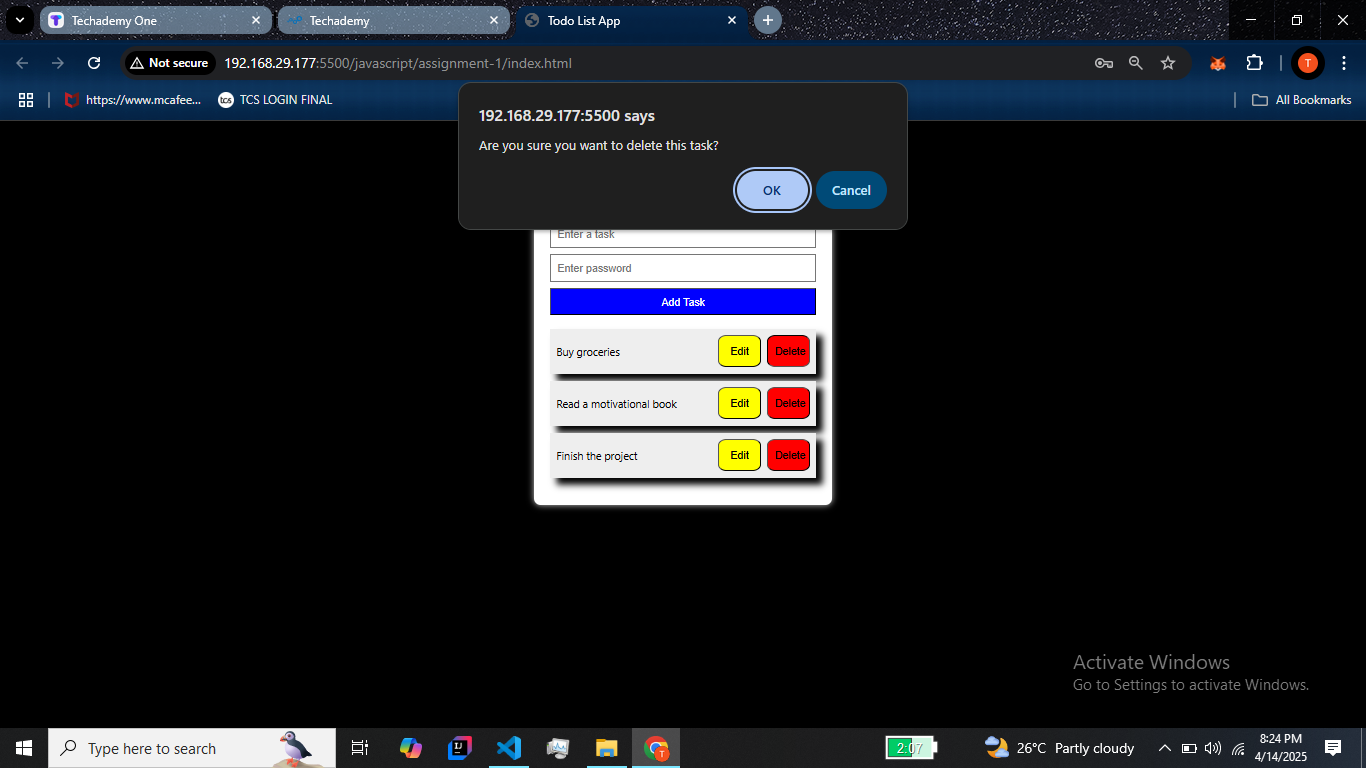
****

****

**d)Editing a task with wrong password**

****

**e)Deleting a task**

****