# **Name : Aditi Dhepe**

# **Roll No : 2213774**

# Experiment No:2

Create a task management application with drag-and-drop functionality. Users

can move tasks between different boards (e.g., To Do, In Progress, Done) using

drag-and-drop interactions. Implement persistence using local storage.

## Theory:

1. **User Interface (UI):** Design a user-friendly interface with separate boards representing different task states (e.g., To Do, In Progress, Done). Each board should display tasks associated with its corresponding state.
2. **Task Representation:** Define a data structure to represent tasks, including attributes such as task name, description, due date, and status (To Do, In Progress, Done).
3. **Drag-and-Drop Interaction:** Implement drag-and-drop functionality to allow users to move tasks between boards. When a user drags a task from one board to another, the task's status should be updated accordingly.
4. **Local Storage:** Utilize the local storage API to persistently store task data on the user's device. This ensures that tasks remain saved even if the user closes or refreshes the application.
5. **Event Handling:** Implement event handlers to capture drag-and-drop events. Update the task status in the data structure and reflect the changes in the UI.

## Source code:

<!-- Create a task management application with drag-and-drop functionality. Users

can move tasks between different boards (e.g., To Do, In Progress, Done) using

drag-and-drop interactions. Implement persistence using local storage. -->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Task Management</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            display: flex;

            justify-content: center;

            height: 100vh;

            background-color: #fbf6d5;

        }

        .board {

            width: 300px;

            margin: 10px;

            background-color: #ffdb65;

            border-radius: 5px;

            padding: 10px;

            min-height: 300px;

        }

        .task {

            padding: 10px;

            background-color: white;

            border-radius: 5px;

            margin-bottom: 10px;

            cursor: move;

        }

    </style>

</head>

<body>

    <div id="toDo" class="board" ondrop="drop(event)" ondragover="allowDrop(event)">

        <h3>To Do</h3>

    </div>

    <div id="inProgress" class="board" ondrop="drop(event)" ondragover="allowDrop(event)">

        <h3>In Progress</h3>

    </div>

    <div id="done" class="board" ondrop="drop(event)" ondragover="allowDrop(event)">

        <h3>Done</h3>

    </div>

    <script>

        function allowDrop(event) {

            event.preventDefault();

        }

        function drag(event) {

            event.dataTransfer.setData("text", event.target.id);

        }

        function drop(event) {

            event.preventDefault();

            var data = event.dataTransfer.getData("text");

            event.target.appendChild(document.getElementById(data));

            updateLocalStorage();

        }

        function createTask(board, text) {

            var task = document.createElement("div");

            task.className = "task";

            task.id = "task" + new Date().getTime();

            task.draggable = "true";

            task.ondragstart = drag;

            task.textContent = text;

            document.getElementById(board).appendChild(task);

        }

        var tasks = {

            toDo: ["Task 1", "Task 2"],

            inProgress: ["Task 3"],

            done: ["Task 4"]

        };

        function init() {

            if (localStorage.getItem("tasks")) {

                tasks = JSON.parse(localStorage.getItem("tasks"));

            }

            for (var board in tasks) {

                tasks[board].forEach(function(text) {

                    createTask(board, text);

                });

            }

        }

        function updateLocalStorage() {

            var tasks = {

                toDo: Array.from(document.getElementById("toDo").getElementsByClassName("task")).map(task =>

                    task.textContent),

                inProgress: Array.from(document.getElementById("inProgress").getElementsByClassName("task")).map(task =>

                    task.textContent),

                done: Array.from(document.getElementById("done").getElementsByClassName("task")).map(task =>

                    task.textContent)

            };

            localStorage.setItem("tasks", JSON.stringify(tasks));

        }

        window.onload = init;

    </script>

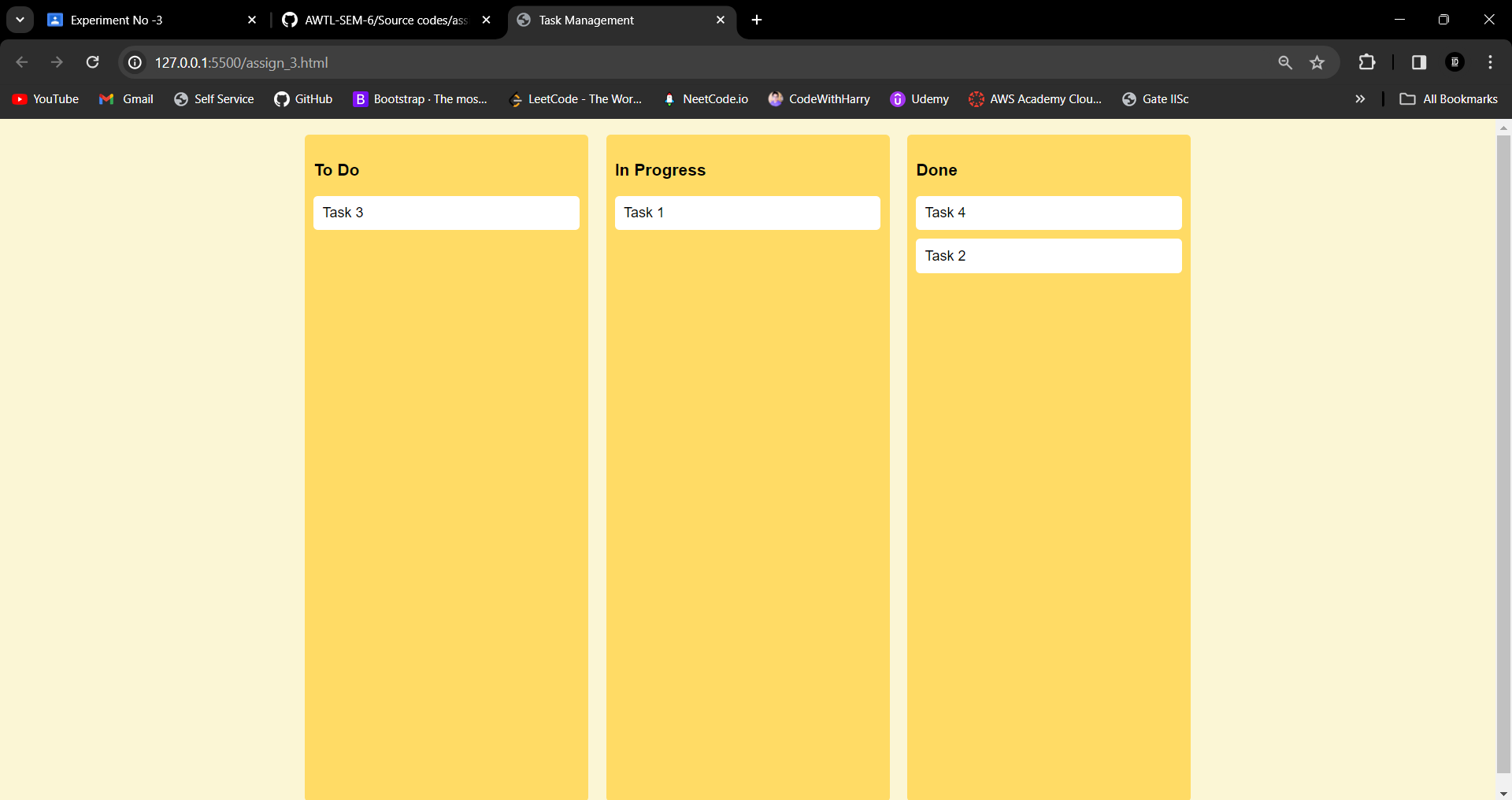
</body>

</html>

## Conclusion:

The completed task management application provides users with an intuitive and efficient way to manage their tasks through drag-and-drop interactions. The use of local storage ensures persistent data storage, enhancing the user experience by retaining task information between sessions.

## Output (Screenshots):



Github : <https://github.com/howlcat25/AWTL-SEM-6/blob/main/Source%20codes/assign_3.html>