# Topic: Introduction to WWW, HTML and CSS

Name : Sanika Zende

Roll no:2213803

TY CC-2[B]

# Experiment No:1

Develop a to-do list application where users can add tasks, mark them as completed, and remove them. Use HTML for the layout, CSS for styling, and JavaScript for handling user interactions and task management.

## Objective:

* To create an interactive To-Do List where users can add, complete, and delete tasks.

## Theory:

* HTML
* CSS
* JavaScript
* Navigation Bar
* To-Do List Functionality
* Drag-and-Drop Task Management

Code:

<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>Practical</title>

        <style>

            body {

                margin: 4rem;

            }

            #buttons {

                margin-bottom: 20px;

            }

            label {

                display: block;

                margin-top: 10px;

            }

            input,

            textarea {

                width: 300px;

                padding: 8px;

                margin-top: 5px;

                border: 1px solid #ccc;

                border-radius: 4px;

            }

            button {

                margin: 10px;

                padding: 8px 16px;

                background-color: #692b42;

                color: #fff;

                border: none;

                border-radius: 4px;

                cursor: pointer;

            }

            button:hover {

                background-color: #250118;

            }

            table {

                border-collapse: collapse;

                margin-bottom: 20px;

                width: 60%;

            }

            th,

            td {

                border: 1px solid #ddd;

                padding: 8px;

                text-align: left;

            }

            th {

                background-color: #cb6e95;

                color: #fff;

            }

            tr.linethrough {

    text-decoration: line-through;

                         }

        .checkbox-label {

        display: flex;

        align-items: center;

    }

    .checkbox-input {

        margin-right: 8px;

    }

        </style>

    </head>

    <body>

        <h1 style="color:rgb(18, 89, 89)">My To Do List</h1>

        <div id="buttons">

            <label >My Tasks :</label>

            <input type="text" id="input" placeholder="enter product's name">

            <button onclick="addData()">Add to list</button>

        </div>

        <table id="optable">

            <tr>

                <th >Check</th>

                <th id="ts">Tasks</th>

                <th id="ts">Action</th>

            </tr>

        </table>

        <script>

        function addData() {

            let name = document.getElementById("input").value;

            // get table

            let table = document.getElementById("optable");

            let newRow = table.insertRow(table.rows.length);

            // data into cells

            newRow.insertCell(0).innerHTML=

             '<label> ' +

            '<input type="checkbox" class="checkbox">' +

            '</label>';

            newRow.insertCell(1).innerHTML = name;

            newRow.insertCell(2).innerHTML =

                '<button onclick="deleteData(this)">Delete</button>';

            // Attach the event listener to the newly added checkbox

            let checkbox = newRow.querySelector(".checkbox");

            checkbox.addEventListener("change", function () {

                let row = this.parentNode.parentNode.parentNode;

                if (this.checked) {

                    row.classList.add("linethrough");

                } else {

                    row.classList.remove("linethrough");

                }

            });

            // clear inputs

            clearInputs();

        }

            function editData(button) {

                let row = button.parentNode.parentNode;

                let nameCell = row.cells[0];

                // prompt to update value

                let nameInput = prompt("Enter the updated name:", nameCell.innerHTML);

                nameCell.innerHTML = nameInput;

            }

            function deleteData(button) {

                // Get the parent row of the clicked button

                let row = button.parentNode.parentNode;

                // Remove the row from the table

                row.parentNode.removeChild(row);

            }

            function clearInputs() {

                // Clear input fields

                document.getElementById("input").value = "";

            }

        </script>

    </body>

</html>

## Conclusion:

This experiment emphasizes on the use of the java script and DOM for the action handling .

## Output (Screenshots):

