# **Name : Aditi Dhepe**

# **Roll No : 2213774**

# Topic: Introduction to WWW, HTML and CSS

# 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

## Source code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Task Checklist</title>

<style>

body {

padding: left 50px;

align-items: center;

margin: 2rem;

background-color: rgb(248, 246, 255);

}

h1{

font-style:normal;

}

table{

table-layout: auto;

}

button{

width: px;

padding: 7px;

}

button:hover {

background-color: #c1aeff;

}

input,

textarea {

width: 400px;

padding: 8px;

margin-top: 5px;

border: 1px solid #ccc;

border-radius: 4px;

}

table {

border-collapse: collapse;

margin-bottom: 20px;

width: 60%;

}

th,

td {

border: 1px solid #ddd;

padding: 4px;

text-align: center;

}

th {

background-color: #ab93fc;

color: #fff;

}

input[type="checkbox"] {

margin-right: 5px;

width: fit-content;

}

td.task-complete {

text-decoration: line-through;

}

</style>

</head>

<body>

<h1 style="color: blueviolet">Task CheckList</h1>

<div>

<br>

<label for="object">

Enter Task:

</label>

<input type="text"

id="object"

placeholder="Enter task">

<br>

<br>

<label for="quantity">

Enter Deadline:

</label>

<input type="date"

id="quantity"

placeholder="Enter deadline">

<br>

<br>

<button onclick="addObject()">

Queue task

</button>

<br>

<br>

</div>

<table id="outputTable">

<tr>

<th>Status</th>

<th>Task Scheduled</th>

<th>Deadline</th>

<th>Action</th>

</tr>

</table>

<script>

function addObject(){

let object = document.getElementById("object").value;

let quantity = document.getElementById("quantity").value;

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

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

let checkboxCell = newRow.insertCell(0);

let objectCell = newRow.insertCell(1);

let quantityCell = newRow.insertCell(2);

newRow.insertCell(3).innerHTML =

'<button onclick="removeObject(this)">remove</button>';

let checkbox = document.createElement("input");

checkbox.type = "checkbox";

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

checkStatus(this, newRow);

});

checkboxCell.appendChild(checkbox);

objectCell.innerHTML = object;

quantityCell.innerHTML = quantity;

clearInputs();

}

function removeObject(button) {

let row = button.parentNode.parentNode;

row.parentNode.removeChild(row);

}

function checkStatus(checkbox, row) {

let taskCell = row.cells[1];

if (checkbox.checked) {

taskCell.classList.add("task-complete");

} else {

taskCell.classList.remove("checked-incomplete");

}

}

function clearInputs() {

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

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

}

</script>

</body>

</html>

## Conclusion:

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

Add Task:

When the user clicks "Queue task," it adds a new row to the table with a checkbox, task description, deadline, and a remove button.

Remove Task:

Clicking the "remove" button in a row deletes that specific task from the table.

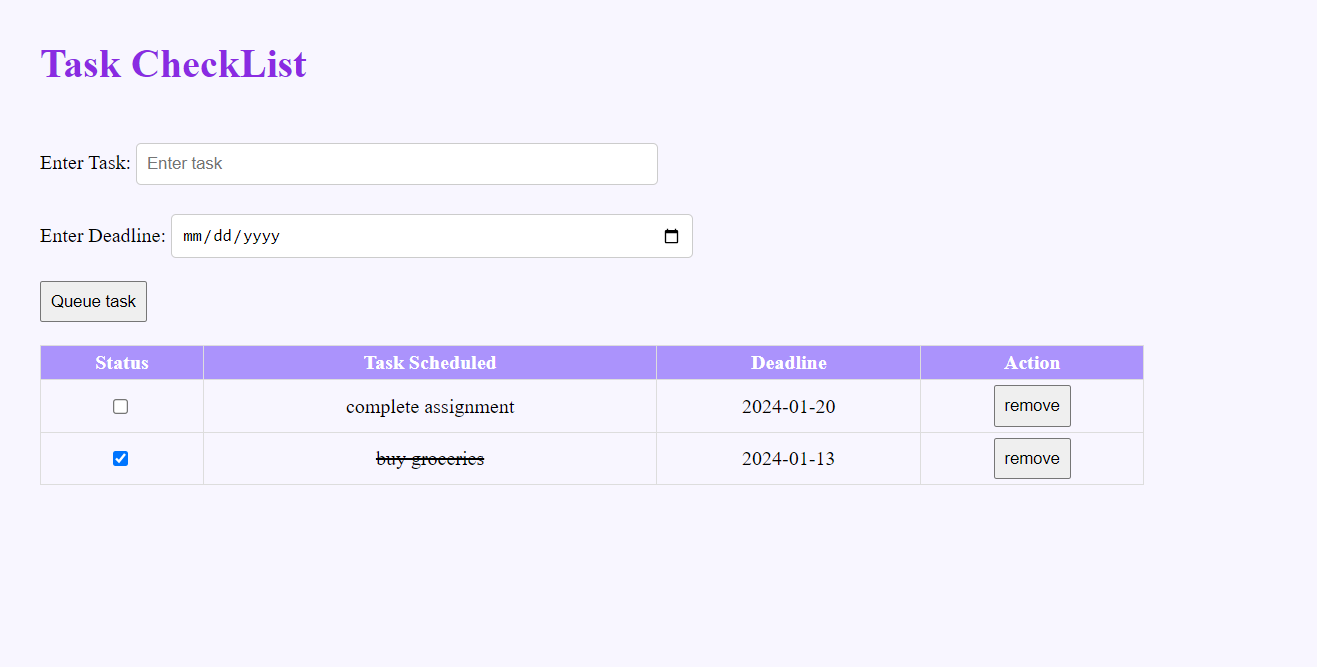
Task Completion:

Checkbox state changes trigger the checkStatus function, adding or removing a strikethrough effect on the task description based on the checkbox state.

Clear Inputs:

After adding a task, input fields are cleared for the user to enter new tasks.

## Output (Screenshots):



Github : <https://github.com/howlcat25/AWTL-SEM-6/blob/main/Source%20codes/Assign_1.html>