**TASK 2**

**To Do List:**

Using HTML,CSS,JS

HTML Code;

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>To-Do List</title>

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

</head>

<body>

<div class="container">

<h1>To-Do List</h1>

<form id="task-form">

<input type="text" id="task-input" placeholder="Add a new task" required>

<button type="submit">Add Task</button>

</form>

<ul id="task-list"></ul>

</div>

<script src="script.js"></script>

</body>

</html>

CSS Code;

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.container {

background: #fff;

padding: 20px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 400px;

text-align: center;

}

h1 {

margin: 0 0 20px;

}

form {

display: flex;

justify-content: space-between;

margin-bottom: 20px;

}

input {

width: 75%;

padding: 10px;

border: 1px solid #ddd;

}

button {

padding: 10px;

border: none;

background: #28a745;

color: #fff;

cursor: pointer;

}

button:hover {

background: #218838;

}

ul {

list-style: none;

padding: 0;

}

li {

background: #f4f4f4;

margin: 5px 0;

padding: 10px;

border: 1px solid #ddd;

display: flex;

justify-content: space-between;

align-items: center;

}

li button {

background: #dc3545;

color: #fff;

border: none;

cursor: pointer;

}

li button:hover {

background: #c82333;

}

Java Script Code;

document.addEventListener('DOMContentLoaded', function () {

const taskForm = document.getElementById('task-form');

const taskInput = document.getElementById('task-input');

const taskList = document.getElementById('task-list');

// Load tasks from local storage

const tasks = JSON.parse(localStorage.getItem('tasks')) || [];

// Function to render tasks

function renderTasks() {

taskList.innerHTML = '';

tasks.forEach((task, index) => {

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

li.innerHTML = `

<span contenteditable="true" onblur="editTask(${index}, this)">${task}</span>

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

`;

taskList.appendChild(li);

});

}

// Add task

taskForm.addEventListener('submit', function (e) {

e.preventDefault();

const newTask = taskInput.value.trim();

if (newTask) {

tasks.push(newTask);

taskInput.value = '';

saveTasks();

renderTasks();

}

});

// Save tasks to local storage

function saveTasks() {

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

}

// Delete task

window.deleteTask = function (index) {

tasks.splice(index, 1);

saveTasks();

renderTasks();

}

// Edit task

window.editTask = function (index, element) {

tasks[index] = element.textContent.trim();

saveTasks();

}

// Initial render

renderTasks();

});

Output;

