**COGNORISE INFOTECH**

**TASK 1**

**TO-DO-LIST**

Build a simple to-do list application that allows users to add,edit, and delete tasks.

You can use HTML, CSS, and JavaScriptfor the front-end and store the tasks in local storage or a simplebackend server using technologies like Node.js or Flask

**CODE:**

Html:

<!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="styles.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:**

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.container {

background-color: #fff;

padding: 20px;

border-radius: 8px;

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

width: 300px;

}

h1 {

text-align: center;

margin-bottom: 20px;

}

form {

display: flex;

}

form input {

flex: 1;

padding: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

form button {

padding: 10px;

background-color: #28a745;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

margin-left: 5px;

}

form button:hover {

background-color: #218838;

}

ul {

list-style-type: none;

padding: 0;

}

li {

background-color: #f9f9f9;

border: 1px solid #ddd;

padding: 10px;

margin-top: 10px;

display: flex;

justify-content: space-between;

align-items: center;

border-radius: 4px;

}

li .edit-btn,

li .delete-btn {

background-color: #007bff;

color: #fff;

border: none;

padding: 5px 10px;

cursor: pointer;

border-radius: 4px;

margin-left: 5px;

}

li .edit-btn:hover {

background-color: #0056b3;

}

li .delete-btn {

background-color: #dc3545;

}

li .delete-btn:hover {

background-color: #c82333;

}

li input[type="text"] {

flex: 1;

padding: 5px;

margin-right: 10px;

}

JAVA SRIPT:

document.addEventListener('DOMContentLoaded', () => {

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

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

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

// Load tasks from local storage

loadTasks();

taskForm.addEventListener('submit', (e) => {

e.preventDefault();

addTask(taskInput.value);

taskInput.value = '';

});

taskList.addEventListener('click', (e) => {

if (e.target.classList.contains('delete-btn')) {

deleteTask(e.target.parentElement);

} else if (e.target.classList.contains('edit-btn')) {

editTask(e.target.parentElement);

}

});

function loadTasks() {

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

tasks.forEach(task => addTask(task, false));

}

function addTask(taskText, save = true) {

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

const taskSpan = document.createElement('span');

taskSpan.textContent = taskText;

const editBtn = document.createElement('button');

editBtn.textContent = 'Edit';

editBtn.classList.add('edit-btn');

const deleteBtn = document.createElement('button');

deleteBtn.textContent = 'Delete';

deleteBtn.classList.add('delete-btn');

li.appendChild(taskSpan);

li.appendChild(editBtn);

li.appendChild(deleteBtn);

taskList.appendChild(li);

if (save) {

saveTask(taskText);

}

}

function saveTask(taskText) {

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

tasks.push(taskText);

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

}

function deleteTask(taskElement) {

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

const taskText = taskElement.firstChild.textContent;

const index = tasks.indexOf(taskText);

if (index > -1) {

tasks.splice(index, 1);

}

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

taskElement.remove();

}

function editTask(taskElement) {

const taskText = taskElement.firstChild.textContent;

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

input.type = 'text';

input.value = taskText;

const saveBtn = document.createElement('button');

saveBtn.textContent = 'Save';

saveBtn.classList.add('edit-btn');

taskElement.firstChild.replaceWith(input);

taskElement.querySelector('.edit-btn').replaceWith(saveBtn);

saveBtn.addEventListener('click', () => {

const newTaskText = input.value;

input.replaceWith(document.createTextNode(newTaskText));

saveBtn.replaceWith(editBtn);

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

const index = tasks.indexOf(taskText);

if (index > -1) {

tasks[index] = newTaskText;

}

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

});

}

});