**Practical 07**

**To-Do List App (Add, Edit, Delete)**

<h2>To-Do List</h2>

<input type="text" id="taskInput" placeholder="Enter a task"> <button onclick="addTask()">Add</button>

<ul id="todoList"></ul>

<script>

function addTask() {

const input = document.getElementById('taskInput'); const taskText = input.value.trim();

if (!taskText) return;

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

const span = document.createElement('span'); span.textContent = taskText;

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

editBtn.onclick = function () {

const newTask = prompt('Edit task:', span.textContent); if (newTask !== null) span.textContent = newTask; };

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

deleteBtn.onclick = function () {

li.remove();

};

li.appendChild(span);

li.appendChild(editBtn);

li.appendChild(deleteBtn);

document.getElementById('todoList').appendChild(li); input.value = '';

}

</script>

**Practice Tasks for To-Do List App**

1. Modify the code so that when editing a task, the new task cannot be empty or just spaces.

2. Add a "Complete" button that toggles a task between *normal* and *completed* (e.g., with a strikethrough style).

3. Ensure that if a task with the same text already exists, it should not be added again.

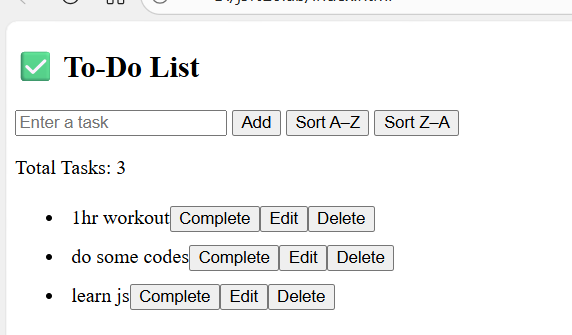
4. Display the total number of tasks and update the count dynamically when tasks are added, edited, or deleted.

5. Allow the user to press **Enter** instead of clicking the Add button to add a task. 6. Make the background color of a <li> change when the mouse hovers over it. 7. Show a confirmation (confirm()) before deleting a task to avoid accidental deletion.

8. Replace the prompt() with an inline <input> field that appears inside the list item for editing.

9. Save the tasks in localStorage so that they remain even after refreshing the page.

10. Add two buttons: one to sort tasks alphabetically (A–Z) and one to sort them in reverse (Z–A).



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>To-Do List App</title>

<style>

li {

padding: 5px;

transition: background 0.3s;

}

li:hover {

background: #f0f0f0;

}

.completed {

text-decoration: line-through;

color: gray;

}

</style>

</head>

<body>

<h2>✅ To-Do List</h2>

<input type="text" id="taskInput" placeholder="Enter a task">

<button onclick="addTask()">Add</button>

<button onclick="sortTasks(true)">Sort A–Z</button>

<button onclick="sortTasks(false)">Sort Z–A</button>

<p>Total Tasks: <span id="taskCount">0</span></p>

<ul id="todoList"></ul>

<script>

const taskInput = document.getElementById("taskInput");

const todoList = document.getElementById("todoList");

const taskCount = document.getElementById("taskCount");

// Load tasks from localStorage

window.onload = function () {

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

saved.forEach(task => createTask(task.text, task.completed));

updateCount();

};

// Add Task

function addTask() {

const taskText = taskInput.value.trim();

if (!taskText) return;

// Prevent duplicate tasks

const tasks = [...document.querySelectorAll("#todoList span")].map(s => s.textContent);

if (tasks.includes(taskText)) {

alert("Task already exists!");

return;

}

createTask(taskText, false);

taskInput.value = "";

saveTasks();

updateCount();

}

// Create Task Function

function createTask(taskText, completed) {

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

const span = document.createElement("span");

span.textContent = taskText;

if (completed) span.classList.add("completed");

// Complete button

const completeBtn = document.createElement("button");

completeBtn.textContent = "Complete";

completeBtn.onclick = function () {

span.classList.toggle("completed");

saveTasks();

};

// Edit button (inline input)

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

editBtn.textContent = "Edit";

editBtn.onclick = function () {

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

input.type = "text";

input.value = span.textContent;

li.replaceChild(input, span);

input.addEventListener("blur", function () {

const newTask = input.value.trim();

if (newTask) {

span.textContent = newTask;

li.replaceChild(span, input);

saveTasks();

updateCount();

} else {

li.replaceChild(span, input);

}

});

input.focus();

};

// Delete button (with confirmation)

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

deleteBtn.textContent = "Delete";

deleteBtn.onclick = function () {

if (confirm("Are you sure you want to delete this task?")) {

li.remove();

saveTasks();

updateCount();

}

};

li.appendChild(span);

li.appendChild(completeBtn);

li.appendChild(editBtn);

li.appendChild(deleteBtn);

todoList.appendChild(li);

}

// Update Task Count

function updateCount() {

taskCount.textContent = document.querySelectorAll("#todoList li").length;

}

// Save to localStorage

function saveTasks() {

const tasks = [];

document.querySelectorAll("#todoList li").forEach(li => {

tasks.push({

text: li.querySelector("span").textContent,

completed: li.querySelector("span").classList.contains("completed")

});

});

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

}

// Sort tasks

function sortTasks(ascending) {

const tasks = [...document.querySelectorAll("#todoList li")];

tasks.sort((a, b) => {

const textA = a.querySelector("span").textContent.toLowerCase();

const textB = b.querySelector("span").textContent.toLowerCase();

return ascending ? textA.localeCompare(textB) : textB.localeCompare(textA);

});

todoList.innerHTML = "";

tasks.forEach(li => todoList.appendChild(li));

saveTasks();

}

// Add task with Enter key

taskInput.addEventListener("keypress", function (e) {

if (e.key === "Enter") {

addTask();

}

});

</script>

</body>

</html>