HTML code of the TO DO LIST project

<!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 href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap" rel="stylesheet">

<style>

body {

font-family: 'Roboto', Arial, sans-serif;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-image: url('C:/Users/admin/Downloads/pexels-jeshoots-com-147458-714701.jpg'); /\* Keep your custom background \*/

background-size: cover;

background-position: center;

background-repeat: no-repeat;

}

.todo-container {

background: rgba(255, 255, 255, 0.9);

border-radius: 8px;

padding: 20px;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);

text-align: center;

}

h1 {

font-size: 2rem;

margin-bottom: 20px;

}

.todo-input {

width: 80%;

padding: 10px;

font-size: 1rem;

margin-bottom: 10px;

border: 1px solid #ddd;

border-radius: 4px;

}

.todo-button {

padding: 10px 20px;

font-size: 1rem;

color: white;

background-color: #28a745;

border: none;

border-radius: 4px;

cursor: pointer;

}

.todo-button:hover {

background-color: #218838;

}

.todo-list {

list-style: none;

padding: 0;

}

.todo-item {

background: #f8f9fa;

margin: 10px 0;

padding: 10px;

border-radius: 4px;

display: flex;

justify-content: space-between;

align-items: center;

border: 1px solid #ddd;

}

.todo-item.completed {

text-decoration: line-through;

opacity: 0.7;

}

.todo-actions button {

background: none;

border: none;

cursor: pointer;

font-size: 1.2rem;

margin-left: 10px;

}

.todo-actions button:hover {

color: red;

}

</style>

</head>

<body>

<div class="todo-container">

<h1>To-Do List</h1>

<input type="text" class="todo-input" placeholder="Add a new task...">

<button class="todo-button">Add Task</button>

<ul class="todo-list"></ul>

</div>

<script>

const input = document.querySelector('.todo-input');

const addButton = document.querySelector('.todo-button');

const todoList = document.querySelector('.todo-list');

// Load sounds

const addTaskSound = new Audio('success\_bell-6776.mp3'); // Ensure correct path

const deleteSound = new Audio('error-126627.mp3'); // Ensure correct path

const loadTasks = () => {

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

tasks.forEach(task => addTask(task.text, task.completed));

};

const saveTasks = () => {

const tasks = Array.from(todoList.children).map(item => ({

text: item.querySelector('.task-text').textContent,

completed: item.classList.contains('completed'),

}));

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

};

const addTask = (taskText, completed = false) => {

if (!taskText.trim()) {

alert('Task cannot be empty.');

return;

}

const MAX\_TASKS = 5;

if (todoList.children.length >= MAX\_TASKS) {

alert('You cannot add more than 5 tasks.');

return;

}

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

listItem.classList.add('todo-item');

if (completed) listItem.classList.add('completed');

listItem.innerHTML = `

<span class="task-text">${taskText}</span>

<div class="todo-actions">

<button class="edit">&#9998;</button>

<button class="delete">&#10005;</button>

</div>

`;

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

listItem.classList.toggle('completed');

saveTasks();

});

listItem.querySelector('.delete').addEventListener('click', (e) => {

e.stopPropagation();

deleteSound.play();

listItem.remove();

saveTasks();

});

todoList.appendChild(listItem);

addTaskSound.play();

saveTasks();

};

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

addTask(input.value);

input.value = '';

});

input.addEventListener('keydown', (e) => {

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

addTask(input.value);

input.value = '';

}

});

loadTasks();

</script>

</body>

</html>