KARL ANDREW GOJAR  
BIT32

S-ITPE003LA: EA1 Finals

GOJAR\_FinalsEA1.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 Application</title>

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

</head>

<body>

    <div class="container">

        <div class="header">

            <h1 class="todo-heading">To Do List</h1>

            <div class="form-group">

                <input type="text" id="taskInput" placeholder="Add new task...">

                <button id="addTaskBtn">Add Task</button>

            </div>

        </div>

        <div class="task-section">

            <div class="added-tasks">

                <h2>Added Tasks</h2>

                <ul id="taskList" class="task-list">

                </ul>

            </div>

            <div class="removed-tasks">

                <h2>Removed Tasks</h2>

                <ul id="removedTaskList" class="task-list">

                </ul>

            </div>

        </div>

    </div>

    <script src="script.js"></script>

</body>

</html>

styles.css

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

    background-color: #f2f2f2;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

}

.container {

    width: 100%;

    max-width: 800px;

    background-color: #fff;

    border-radius: 8px;

    box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);

    padding: 20px;

}

.header {

    text-align: center;

    margin-bottom: 20px;

}

.todo-heading {

    color: #08490b;

    margin-bottom: 10px;

}

.form-group {

    display: flex;

    align-items: center;

    justify-content: center;

    margin-bottom: 20px;

}

.form-group input[type="text"] {

    padding: 10px;

    width: calc(100% - 120px);

    border: 1px solid #ccc;

    border-radius: 4px;

    margin-right: 10px;

}

.form-group button {

    cursor: pointer;

    border: none;

    border-radius: 4px;

    padding: 10px 20px;

    background-color: #08490b;

    color: #fff;

}

.form-group button:hover {

    background-color: #062b07;

}

.task-section {

    display: grid;

    grid-template-columns: 1fr 1fr;

    gap: 20px;

}

.added-tasks,

.removed-tasks {

    padding: 20px;

    background-color: #f9f9f9;

    border-radius: 6px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.added-tasks h2,

.removed-tasks h2 {

    color: #08490b;

    margin-bottom: 10px;

}

.task-list {

    list-style-type: none;

    padding: 0;

}

.task-item {

    display: flex;

    align-items: center;

    padding: 10px;

    background-color: #fff;

    border-radius: 4px;

    margin-bottom: 10px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.task-item .task-checkbox {

    margin-right: 10px;

}

.task-item .task-text {

    margin-left: 10px;

    flex: 1;

    color: #333;

}

.task-item .remove-btn {

    cursor: pointer;

    border: none;

    border-radius: 20px;

    padding: 10px 20px;

    background-color: #e74c3c;

    color: #fff;

    outline: none;

}

.task-item .remove-btn:hover {

    background-color: #c0392b;

}

.task-item .remove-btn:focus {

    box-shadow: 0 0 0 2px #e74c3c;

}

.task-item .restore-btn {

    cursor: pointer;

    border: none;

    border-radius: 4px;

    padding: 10px 20px;

}

.task-item .remove-btn {

    background-color: #e74c3c;

    color: #fff;

}

.task-item .remove-btn:hover {

    background-color: #c0392b;

}

.task-item .restore-btn {

    background-color: #2ecc71;

    color: #fff;

    margin-left: auto;

    outline: none;

    border: 2px solid #2ecc71;

}

.task-item .restore-btn:hover {

    background-color: #27ae60;

}

.task-item.completed .task-text {

    text-decoration: line-through;

    color: #999;

}

.task-item.completed .remove-btn,

.task-item.completed .restore-btn {

    display: none;

}

.removed-task-item {

    display: flex;

    align-items: center;

    padding: 10px;

    background-color: #f9f9f9;

    border-radius: 4px;

    margin-bottom: 10px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.removed-task-item .task-text {

    flex: 1;

    color: #666;

}

.removed-task-item .restore-btn {

    background-color: #2ecc71;

    color: #fff;

    padding: 10px 20px;

    border-radius: 25px;

    margin-left: auto;

    outline: none;

    border: 2px solid #2ecc71;

}

.removed-task-item .restore-btn:hover {

    background-color: #27ae60;

}

script.js

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

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

    const addTaskBtn = document.getElementById('addTaskBtn');

    const taskList = document.getElementById('taskList');

    const removedTaskList = document.getElementById('removedTaskList');

    addTaskBtn.addEventListener('click', function () {

        const taskText = taskInput.value.trim();

        if (taskText === '') {

            alert('Please enter a task!');

            return;

        }

        const taskItem = createTaskItem(taskText);

        taskList.appendChild(taskItem);

        taskInput.value = '';

    });

    function createTaskItem(taskText) {

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

        taskItem.classList.add('task-item');

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

        checkbox.type = 'checkbox';

        checkbox.classList.add('task-checkbox');

        taskItem.appendChild(checkbox);

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

        taskTextSpan.textContent = taskText;

        taskTextSpan.classList.add('task-text');

        taskItem.appendChild(taskTextSpan);

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

        removeBtn.textContent = 'Remove Task';

        removeBtn.classList.add('remove-btn');

        removeBtn.style.display = 'none';

        removeBtn.addEventListener('click', function () {

            taskList.removeChild(taskItem);

            const removedTaskItem = createRemovedTaskItem(taskText);

            removedTaskList.appendChild(removedTaskItem);

        });

        taskItem.appendChild(removeBtn);

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

            if (checkbox.checked) {

                taskTextSpan.classList.add('completed-task');

                removeBtn.style.display = 'inline-block';

            } else {

                taskTextSpan.classList.remove('completed-task');

                removeBtn.style.display = 'none';

            }

        });

        return taskItem;

    }

    function createRemovedTaskItem(taskText) {

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

        removedTaskItem.classList.add('removed-task-item');

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

        taskTextSpan.textContent = taskText;

        removedTaskItem.appendChild(taskTextSpan);

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

        restoreBtn.textContent = 'Restore';

        restoreBtn.classList.add('restore-btn');

        restoreBtn.addEventListener('click', function () {

            removedTaskList.removeChild(removedTaskItem);

            const taskItem = createTaskItem(taskText);

            taskList.appendChild(taskItem);

        });

        removedTaskItem.appendChild(restoreBtn);

        return removedTaskItem;

    }

});

app.js

const express = require('express');

const path = require('path');

const app = express();

const port = 8080;

const publicPath = path.join(\_\_dirname, 'public');

const router = express.Router();

router.get('/', (req, res) => {

    res.sendFile(path.join(publicPath, 'GOJAR\_FinalsEA1.html'));

});

app.use(express.static(publicPath));

app.use('/', router);

app.listen(port, () => {

    console.log(`To-Do List app listening at http://localhost:${port}`);

});

Execution Screenshot

A screenshot of a to do list

Description automatically generated

No Inputted Task Error

A screenshot of a computer

Description automatically generated

Accomplished Task (Checked Box) with Remove Feature

A screenshot of a to do list

Description automatically generated

Removed Tasks

A screenshot of a to do list

Description automatically generated

Restore Feature

A screenshot of a to do list

Description automatically generated