NAME: Sampada Sawant

**CODE:**

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Study Session Planner</title>

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

</head>

<body>

    <div class="container">

        <header>

            <h1>Study Session Planner</h1>

            <p>Plan study sessions with your classmates</p>

        </header>

        <main>

            <section class="create-session">

                <h2>Create New Study Session</h2>

                <form id="session-form">

                    <div class="form-group">

                        <label for="session-name">Session Name:</label>

                        <input type="text" id="session-name" required>

                    </div>

                    <div class="form-group">

                        <label for="session-date">Date:</label>

                        <input type="date" id="session-date" required>

                    </div>

                    <div class="form-group">

                        <label for="session-time">Time:</label>

                        <input type="time" id="session-time" required>

                    </div>

                    <div class="form-group">

                        <label for="session-tasks">Tasks (comma separated):</label>

                        <textarea id="session-tasks" rows="3" required></textarea>

                    </div>

                    <button type="submit">Create Session</button>

                </form>

            </section>

            <section class="sessions-list">

                <h2>Upcoming Study Sessions</h2>

                <div id="sessions-container">

                    <!-- Sessions will be added here by JavaScript -->

                    <p class="empty-message">No study sessions planned yet.</p>

                </div>

            </section>

        </main>

    </div>

    <script src="script.js"></script>

</body>

</html>

CSS:

/\* Basic Reset \*/

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

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

    line-height: 1.6;

    background-color: #f5f5f5;

    color: #333;

    padding: 20px;

}

.container {

    max-width: 1000px;

    margin: 0 auto;

    background-color: white;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

}

header {

    text-align: center;

    margin-bottom: 30px;

    padding-bottom: 20px;

    border-bottom: 1px solid #eee;

}

header h1 {

    color: #2c3e50;

    margin-bottom: 10px;

}

main {

    display: grid;

    grid-template-columns: 1fr 1fr;

    gap: 30px;

}

.create-session, .sessions-list {

    background-color: #f9f9f9;

    padding: 20px;

    border-radius: 6px;

}

h2 {

    color: #2c3e50;

    margin-bottom: 15px;

    font-size: 1.3rem;

}

.form-group {

    margin-bottom: 15px;

}

label {

    display: block;

    margin-bottom: 5px;

    font-weight: 600;

}

input, textarea {

    width: 100%;

    padding: 10px;

    border: 1px solid #ddd;

    border-radius: 4px;

    font-family: inherit;

}

button {

    background-color: #3498db;

    color: white;

    border: none;

    padding: 10px 20px;

    border-radius: 4px;

    cursor: pointer;

    font-weight: 600;

    transition: background-color 0.3s;

}

button:hover {

    background-color: #2980b9;

}

.session-card {

    background-color: white;

    border: 1px solid #ddd;

    border-radius: 4px;

    padding: 15px;

    margin-bottom: 15px;

    position: relative;

}

.session-card h3 {

    color: #2c3e50;

    margin-bottom: 10px;

}

.session-card p {

    margin-bottom: 8px;

}

.session-card .tasks {

    margin-top: 10px;

    padding-top: 10px;

    border-top: 1px dashed #eee;

}

.task-item {

    display: flex;

    align-items: center;

    margin-bottom: 5px;

}

.task-item input {

    width: auto;

    margin-right: 10px;

}

.empty-message {

    text-align: center;

    color: #7f8c8d;

    font-style: italic;

}

.delete-btn {

    position: absolute;

    top: 10px;

    right: 10px;

    background-color: #e74c3c;

    color: white;

    border: none;

    width: 25px;

    height: 25px;

    border-radius: 50%;

    display: flex;

    align-items: center;

    justify-content: center;

    cursor: pointer;

    font-size: 12px;

}

.delete-btn:hover {

    background-color: #c0392b;

}

@media (max-width: 768px) {

    main {

        grid-template-columns: 1fr;

    }

}

JAVASCRIPT:

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

    const sessionForm = document.getElementById('session-form');

    const sessionsContainer = document.getElementById('sessions-container');

    // Load sessions from localStorage when page loads

    loadSessions();

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

        e.preventDefault();

        // Get form values

        const sessionName = document.getElementById('session-name').value;

        const sessionDate = document.getElementById('session-date').value;

        const sessionTime = document.getElementById('session-time').value;

        const sessionTasks = document.getElementById('session-tasks').value;

        // Validate inputs

        if (!sessionName || !sessionDate || !sessionTime || !sessionTasks) {

            alert('Please fill in all fields');

            return;

        }

        // Create session object

        const session = {

            id: Date.now(), // Simple unique ID

            name: sessionName,

            date: sessionDate,

            time: sessionTime,

            tasks: sessionTasks.split(',').map(task => ({

                name: task.trim(),

                completed: false

            }))

        };

        // Save session

        saveSession(session);

        // Clear form

        sessionForm.reset();

    });

    function saveSession(session) {

        let sessions = JSON.parse(localStorage.getItem('studySessions')) || [];

        sessions.push(session);

        localStorage.setItem('studySessions', JSON.stringify(sessions));

        loadSessions();

    }

    function loadSessions() {

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

        sessionsContainer.innerHTML = '';

        if (sessions.length === 0) {

            sessionsContainer.innerHTML = '<p class="empty-message">No study sessions planned yet.</p>';

            return;

        }

        // Sort sessions by date and time

        sessions.sort((a, b) => {

            const dateA = new Date(`${a.date} ${a.time}`);

            const dateB = new Date(`${b.date} ${b.time}`);

            return dateA - dateB;

        });

        sessions.forEach(session => {

            const sessionCard = document.createElement('div');

            sessionCard.className = 'session-card';

            sessionCard.dataset.id = session.id;

            // Format date for display

            const dateObj = new Date(`${session.date} ${session.time}`);

            const options = {

                weekday: 'long',

                year: 'numeric',

                month: 'long',

                day: 'numeric',

                hour: '2-digit',

                minute: '2-digit'

            };

            const formattedDate = dateObj.toLocaleDateString('en-US', options);

            let tasksHTML = '';

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

                tasksHTML += `

                    <div class="task-item">

                        <input type="checkbox" id="task-${session.id}-${index}"

                               ${task.completed ? 'checked' : ''}

                               data-session-id="${session.id}"

                               data-task-index="${index}">

                        <label for="task-${session.id}-${index}"

                               class="${task.completed ? 'completed' : ''}">${task.name}</label>

                    </div>

                `;

            });

            sessionCard.innerHTML = `

                <button class="delete-btn" data-id="${session.id}">×</button>

                <h3>${session.name}</h3>

                <p><strong>When:</strong> ${formattedDate}</p>

                <div class="tasks">

                    <p><strong>Tasks:</strong></p>

                    ${tasksHTML}

                </div>

            `;

            sessionsContainer.appendChild(sessionCard);

        });

        // Add event listeners to delete buttons

        document.querySelectorAll('.delete-btn').forEach(btn => {

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

                const sessionId = parseInt(this.dataset.id);

                deleteSession(sessionId);

            });

        });

        // Add event listeners to task checkboxes

        document.querySelectorAll('.task-item input[type="checkbox"]').forEach(checkbox => {

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

                const sessionId = parseInt(this.dataset.sessionId);

                const taskIndex = parseInt(this.dataset.taskIndex);

                toggleTaskCompletion(sessionId, taskIndex, this.checked);

            });

        });

    }

    function deleteSession(sessionId) {

        let sessions = JSON.parse(localStorage.getItem('studySessions')) || [];

        sessions = sessions.filter(session => session.id !== sessionId);

        localStorage.setItem('studySessions', JSON.stringify(sessions));

        loadSessions();

    }

    function toggleTaskCompletion(sessionId, taskIndex, isCompleted) {

        let sessions = JSON.parse(localStorage.getItem('studySessions')) || [];

        const sessionIndex = sessions.findIndex(session => session.id === sessionId);

        if (sessionIndex !== -1) {

            sessions[sessionIndex].tasks[taskIndex].completed = isCompleted;

            localStorage.setItem('studySessions', JSON.stringify(sessions));

        }

    }

});

**WEBSITE:**

**https://682dd5f1350b0d06db4b6157--dapper-chimera-9dc7da.netlify.app/**