**Admin.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Admin Dashboard</title>

    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <script src="https://cdn.sheetjs.com/xlsx-0.19.3/package/dist/xlsx.full.min.js"></script>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

    <style>

        body {

            font-family: 'Segoe UI', Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f0f2f5;

        }

        .dashboard {

            display: flex;

            min-height: 100vh;

        }

        .sidebar {

            width: 250px;

            background: linear-gradient(135deg, #1a237e 0%, #283593 100%);

            color: white;

            padding: 20px 0;

            box-shadow: 2px 0 5px rgba(0,0,0,0.1);

        }

        .sidebar-header {

            padding: 0 20px 20px;

            border-bottom: 1px solid rgba(255,255,255,0.1);

            text-align: center;

        }

        .sidebar-header h2 {

            margin: 0;

            font-size: 1.5rem;

        }

        .sidebar-menu {

            padding: 20px 0;

        }

        .menu-item {

            padding: 12px 20px;

            cursor: pointer;

            transition: all 0.3s;

            display: flex;

            align-items: center;

            gap: 10px;

        }

        .menu-item:hover, .menu-item.active {

            background-color: rgba(255,255,255,0.1);

        }

        .menu-item i {

            width: 20px;

            text-align: center;

        }

        .main-content {

            flex: 1;

            padding: 20px;

            overflow-y: auto;

        }

        .header {

            background-color: white;

            padding: 15px 20px;

            border-radius: 8px;

            box-shadow: 0 2px 4px rgba(0,0,0,0.05);

            margin-bottom: 20px;

            display: flex;

            justify-content: space-between;

            align-items: center;

        }

        .header h1 {

            margin: 0;

            font-size: 1.8rem;

            color: #1a237e;

        }

        .user-info {

            display: flex;

            align-items: center;

            gap: 10px;

        }

        .user-avatar {

            width: 40px;

            height: 40px;

            border-radius: 50%;

            background-color: #1a237e;

            color: white;

            display: flex;

            align-items: center;

            justify-content: center;

            font-weight: bold;

        }

        .section {

            background-color: white;

            border-radius: 8px;

            box-shadow: 0 2px 4px rgba(0,0,0,0.05);

            padding: 20px;

            margin-bottom: 20px;

        }

        .section-header {

            display: flex;

            justify-content: space-between;

            align-items: center;

            margin-bottom: 20px;

            padding-bottom: 10px;

            border-bottom: 1px solid #e0e0e0;

        }

        .section-header h2 {

            margin: 0;

            font-size: 1.5rem;

            color: #1a237e;

        }

        .btn {

            padding: 8px 16px;

            border: none;

            border-radius: 4px;

            cursor: pointer;

            font-weight: 500;

            transition: all 0.3s ease;

        }

        .btn-primary {

            background-color: #1a237e;

            color: white;

        }

        .btn-primary:hover {

            background-color: #283593;

        }

        .btn-danger {

            background-color: #e53935;

            color: white;

        }

        .btn-danger:hover {

            background-color: #c62828;

        }

        .btn-success {

            background-color: #43a047;

            color: white;

        }

        .btn-success:hover {

            background-color: #388e3c;

        }

        table {

            width: 100%;

            border-collapse: collapse;

            margin-top: 10px;

        }

        th, td {

            padding: 12px 15px;

            text-align: left;

            border-bottom: 1px solid #e0e0e0;

        }

        th {

            background-color: #f5f5f5;

            font-weight: 600;

            color: #333;

        }

        tr:hover {

            background-color: #f9f9f9;

        }

        .action-buttons {

            display: flex;

            gap: 5px;

        }

        .form-group {

            margin-bottom: 15px;

        }

        .form-group label {

            display: block;

            margin-bottom: 5px;

            font-weight: 500;

        }

        .form-group input, .form-group select {

            width: 100%;

            padding: 8px 12px;

            border: 1px solid #ddd;

            border-radius: 4px;

            font-size: 14px;

        }

        .form-row {

            display: flex;

            gap: 15px;

        }

        .form-row .form-group {

            flex: 1;

        }

        .modal {

            display: none;

            position: fixed;

            top: 0;

            left: 0;

            width: 100%;

            height: 100%;

            background-color: rgba(0,0,0,0.5);

            z-index: 1000;

            justify-content: center;

            align-items: center;

        }

        .modal-content {

            background-color: white;

            border-radius: 8px;

            width: 500px;

            max-width: 90%;

            padding: 20px;

            box-shadow: 0 4px 20px rgba(0,0,0,0.2);

        }

        .modal-header {

            display: flex;

            justify-content: space-between;

            align-items: center;

            margin-bottom: 20px;

            padding-bottom: 10px;

            border-bottom: 1px solid #e0e0e0;

        }

        .modal-header h3 {

            margin: 0;

            color: #1a237e;

        }

        .close-modal {

            font-size: 24px;

            cursor: pointer;

            color: #666;

        }

        .modal-footer {

            display: flex;

            justify-content: flex-end;

            gap: 10px;

            margin-top: 20px;

            padding-top: 15px;

            border-top: 1px solid #e0e0e0;

        }

        .dashboard-stats {

            display: grid;

            grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

            gap: 20px;

            margin-bottom: 20px;

        }

        .stat-card {

            background-color: white;

            border-radius: 8px;

            padding: 20px;

            box-shadow: 0 2px 4px rgba(0,0,0,0.05);

            display: flex;

            flex-direction: column;

        }

        .stat-value {

            font-size: 2rem;

            font-weight: bold;

            color: #1a237e;

            margin: 10px 0;

        }

        .stat-label {

            color: #666;

            font-size: 0.9rem;

        }

        .stat-icon {

            align-self: flex-end;

            font-size: 1.5rem;

            color: #1a237e;

            opacity: 0.7;

        }

        .chart-container {

            height: 300px;

            margin-top: 20px;

        }

        .hidden {

            display: none;

        }

    </style>

</head>

<body>

    <div class="dashboard">

        <div class="sidebar">

            <div class="sidebar-header">

                <h2>Admin Panel</h2>

            </div>

            <div class="sidebar-menu">

                <div class="menu-item active" onclick="showSection('dashboard')">

                    <i class="fas fa-tachometer-alt"></i> Dashboard

                </div>

                <div class="menu-item" onclick="showSection('students')">

                    <i class="fas fa-user-graduate"></i> Students

                </div>

                <div class="menu-item" onclick="showSection('subjects')">

                    <i class="fas fa-book"></i> Subjects

                </div>

                <div class="menu-item" onclick="showSection('attendance')">

                    <i class="fas fa-calendar-check"></i> Attendance

                </div>

                <div class="menu-item" onclick="showSection('reports')">

                    <i class="fas fa-chart-bar"></i> Reports

                </div>

                <div class="menu-item" onclick="logout()">

                    <i class="fas fa-sign-out-alt"></i> Logout

                </div>

            </div>

        </div>

        <div class="main-content">

            <div class="header">

                <h1>Admin Dashboard</h1>

                <div class="user-info">

                    <div class="user-avatar">A</div>

                    <div>Admin</div>

                </div>

            </div>

            <!-- Dashboard Section -->

            <div id="dashboard-section" class="section-container">

                <div class="dashboard-stats">

                    <div class="stat-card">

                        <div class="stat-icon"><i class="fas fa-user-graduate"></i></div>

                        <div class="stat-value" id="total-students">0</div>

                        <div class="stat-label">Total Students</div>

                    </div>

                    <div class="stat-card">

                        <div class="stat-icon"><i class="fas fa-book"></i></div>

                        <div class="stat-value" id="total-subjects">0</div>

                        <div class="stat-label">Total Subjects</div>

                    </div>

                    <div class="stat-card">

                        <div class="stat-icon"><i class="fas fa-calendar-check"></i></div>

                        <div class="stat-value" id="attendance-rate">0%</div>

                        <div class="stat-label">Attendance Rate</div>

                    </div>

                    <div class="stat-card">

                        <div class="stat-icon"><i class="fas fa-calendar-day"></i></div>

                        <div class="stat-value" id="today-attendance">0</div>

                        <div class="stat-label">Today's Attendance</div>

                    </div>

                </div>

                <div class="section">

                    <div class="section-header">

                        <h2>Attendance Overview</h2>

                    </div>

                    <div class="chart-container">

                        <canvas id="attendanceChart"></canvas>

                    </div>

                </div>

                <div class="section">

                    <div class="section-header">

                        <h2>Recent Activity</h2>

                    </div>

                    <table>

                        <thead>

                            <tr>

                                <th>Date</th>

                                <th>Activity</th>

                                <th>Details</th>

                            </tr>

                        </thead>

                        <tbody id="activity-log">

                            <!-- Activity logs will be populated here -->

                        </tbody>

                    </table>

                </div>

            </div>

            <!-- Students Section -->

            <div id="students-section" class="section-container hidden">

                <div class="section">

                    <div class="section-header">

                        <h2>Manage Students</h2>

                        <button class="btn btn-primary" onclick="openAddStudentModal()">

                            <i class="fas fa-plus"></i> Add Student

                        </button>

                    </div>

                    <table>

                        <thead>

                            <tr>

                                <th>Roll Number</th>

                                <th>Student Name</th>

                                <th>Actions</th>

                            </tr>

                        </thead>

                        <tbody id="students-table">

                            <!-- Students will be populated here -->

                        </tbody>

                    </table>

                </div>

            </div>

            <!-- Subjects Section -->

            <div id="subjects-section" class="section-container hidden">

                <div class="section">

                    <div class="section-header">

                        <h2>Manage Subjects</h2>

                        <button class="btn btn-primary" onclick="openAddSubjectModal()">

                            <i class="fas fa-plus"></i> Add Subject

                        </button>

                    </div>

                    <table>

                        <thead>

                            <tr>

                                <th>Subject Code</th>

                                <th>Subject Name</th>

                                <th>Actions</th>

                            </tr>

                        </thead>

                        <tbody id="subjects-table">

                            <!-- Subjects will be populated here -->

                        </tbody>

                    </table>

                </div>

            </div>

            <!-- Attendance Section -->

            <div id="attendance-section" class="section-container hidden">

                <div class="section">

                    <div class="section-header">

                        <h2>Attendance Records</h2>

                        <div>

                            <input type="date" id="attendance-date-filter" onchange="filterAttendance()">

                            <select id="subject-filter" onchange="filterAttendance()">

                                <option value="">All Subjects</option>

                            </select>

                        </div>

                    </div>

                    <table>

                        <thead>

                            <tr>

                                <th>Date</th>

                                <th>Subject</th>

                                <th>Roll Number</th>

                                <th>Student Name</th>

                                <th>Status</th>

                                <th>Actions</th>

                            </tr>

                        </thead>

                        <tbody id="attendance-table">

                            <!-- Attendance records will be populated here -->

                        </tbody>

                    </table>

                </div>

            </div>

            <!-- Reports Section -->

            <div id="reports-section" class="section-container hidden">

                <div class="section">

                    <div class="section-header">

                        <h2>Attendance Reports</h2>

                        <div>

                            <select id="report-type">

                                <option value="daily">Daily Report</option>

                                <option value="monthly">Monthly Report</option>

                                <option value="subject">Subject-wise Report</option>

                                <option value="student">Student-wise Report</option>

                            </select>

                            <button class="btn btn-primary" onclick="generateReport()">Generate Report</button>

                            <button class="btn btn-success" onclick="exportReport()">

                                <i class="fas fa-file-excel"></i> Export

                            </button>

                        </div>

                    </div>

                    <div id="report-filters" class="form-row">

                        <div class="form-group">

                            <label>Date Range</label>

                            <div style="display: flex; gap: 10px;">

                                <input type="date" id="report-start-date">

                                <input type="date" id="report-end-date">

                            </div>

                        </div>

                        <div class="form-group">

                            <label>Subject</label>

                            <select id="report-subject">

                                <option value="">All Subjects</option>

                            </select>

                        </div>

                        <div class="form-group">

                            <label>Student</label>

                            <select id="report-student">

                                <option value="">All Students</option>

                            </select>

                        </div>

                    </div>

                    <div class="chart-container">

                        <canvas id="reportChart"></canvas>

                    </div>

                    <div id="report-table-container" style="margin-top: 20px;">

                        <table id="report-table">

                            <thead>

                                <tr>

                                    <!-- Report headers will be dynamically generated -->

                                </tr>

                            </thead>

                            <tbody>

                                <!-- Report data will be populated here -->

                            </tbody>

                        </table>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <!-- Add Student Modal -->

    <div id="add-student-modal" class="modal">

        <div class="modal-content">

            <div class="modal-header">

                <h3>Add New Student</h3>

                <span class="close-modal" onclick="closeModal('add-student-modal')">&times;</span>

            </div>

            <form id="add-student-form">

                <div class="form-group">

                    <label for="student-roll">Roll Number</label>

                    <input type="text" id="student-roll" required>

                </div>

                <div class="form-group">

                    <label for="student-name">Student Name</label>

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

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn" onclick="closeModal('add-student-modal')">Cancel</button>

                    <button type="submit" class="btn btn-primary">Add Student</button>

                </div>

            </form>

        </div>

    </div>

    <!-- Edit Student Modal -->

    <div id="edit-student-modal" class="modal">

        <div class="modal-content">

            <div class="modal-header">

                <h3>Edit Student</h3>

                <span class="close-modal" onclick="closeModal('edit-student-modal')">&times;</span>

            </div>

            <form id="edit-student-form">

                <input type="hidden" id="edit-student-roll">

                <div class="form-group">

                    <label for="edit-student-name">Student Name</label>

                    <input type="text" id="edit-student-name" required>

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn" onclick="closeModal('edit-student-modal')">Cancel</button>

                    <button type="submit" class="btn btn-primary">Update Student</button>

                </div>

            </form>

        </div>

    </div>

    <!-- Add Subject Modal -->

    <div id="add-subject-modal" class="modal">

        <div class="modal-content">

            <div class="modal-header">

                <h3>Add New Subject</h3>

                <span class="close-modal" onclick="closeModal('add-subject-modal')">&times;</span>

            </div>

            <form id="add-subject-form">

                <div class="form-group">

                    <label for="subject-code">Subject Code</label>

                    <input type="text" id="subject-code" required>

                </div>

                <div class="form-group">

                    <label for="subject-name">Subject Name</label>

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

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn" onclick="closeModal('add-subject-modal')">Cancel</button>

                    <button type="submit" class="btn btn-primary">Add Subject</button>

                </div>

            </form>

        </div>

    </div>

    <!-- Edit Subject Modal -->

    <div id="edit-subject-modal" class="modal">

        <div class="modal-content">

            <div class="modal-header">

                <h3>Edit Subject</h3>

                <span class="close-modal" onclick="closeModal('edit-subject-modal')">&times;</span>

            </div>

            <form id="edit-subject-form">

                <input type="hidden" id="edit-subject-code">

                <div class="form-group">

                    <label for="edit-subject-name">Subject Name</label>

                    <input type="text" id="edit-subject-name" required>

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn" onclick="closeModal('edit-subject-modal')">Cancel</button>

                    <button type="submit" class="btn btn-primary">Update Subject</button>

                </div>

            </form>

        </div>

    </div>

    <script>

        // Initialize data

        if (!localStorage.getItem('studentData')) {

            const defaultStudents = [

                { "Roll Number": "1", "Student Name": "John Smith" },

                { "Roll Number": "2", "Student Name": "Emma Johnson" },

                { "Roll Number": "3", "Student Name": "Michael Brown" },

                { "Roll Number": "4", "Student Name": "Sarah Davis" },

                { "Roll Number": "5", "Student Name": "David Wilson" },

                { "Roll Number": "6", "Student Name": "Lisa Anderson" },

                { "Roll Number": "7", "Student Name": "James Taylor" },

                { "Roll Number": "8", "Student Name": "Emily Thomas" },

                { "Roll Number": "9", "Student Name": "Robert Martin" },

                { "Roll Number": "10", "Student Name": "Jennifer White" }

            ];

            localStorage.setItem('studentData', JSON.stringify(defaultStudents));

        }

        if (!localStorage.getItem('subjectData')) {

            const defaultSubjects = [

                { "Subject Code": "JAVA101", "Subject Name": "Java" },

                { "Subject Code": "PY101", "Subject Name": "Python" },

                { "Subject Code": "WD101", "Subject Name": "Web Development" },

                { "Subject Code": "DB101", "Subject Name": "Database" },

                { "Subject Code": "OS101", "Subject Name": "Operating System" }

            ];

            localStorage.setItem('subjectData', JSON.stringify(defaultSubjects));

        }

        if (!localStorage.getItem('activityLog')) {

            const defaultActivity = [

                { date: new Date().toISOString(), activity: "System Initialization", details: "Admin dashboard initialized" }

            ];

            localStorage.setItem('activityLog', JSON.stringify(defaultActivity));

        }

        // Set today's date as default for filters

        document.getElementById('attendance-date-filter').valueAsDate = new Date();

        document.getElementById('report-start-date').valueAsDate = new Date(new Date().setDate(new Date().getDate() - 30));

        document.getElementById('report-end-date').valueAsDate = new Date();

        // Navigation functions

        function showSection(sectionName) {

            // Hide all sections

            document.querySelectorAll('.section-container').forEach(section => {

                section.classList.add('hidden');

            });

            // Show selected section

            document.getElementById(`${sectionName}-section`).classList.remove('hidden');

            // Update active menu item

            document.querySelectorAll('.menu-item').forEach(item => {

                item.classList.remove('active');

            });

            event.currentTarget.classList.add('active');

            // Load section data

            if (sectionName === 'dashboard') {

                loadDashboard();

            } else if (sectionName === 'students') {

                loadStudents();

            } else if (sectionName === 'subjects') {

                loadSubjects();

            } else if (sectionName === 'attendance') {

                loadAttendance();

            } else if (sectionName === 'reports') {

                loadReportFilters();

            }

        }

        // Modal functions

        function openModal(modalId) {

            document.getElementById(modalId).style.display = 'flex';

        }

        function closeModal(modalId) {

            document.getElementById(modalId).style.display = 'none';

        }

        function openAddStudentModal() {

            document.getElementById('add-student-form').reset();

            openModal('add-student-modal');

        }

        function openEditStudentModal(rollNumber) {

            const students = JSON.parse(localStorage.getItem('studentData'));

            const student = students.find(s => s['Roll Number'] === rollNumber);

            if (student) {

                document.getElementById('edit-student-roll').value = student['Roll Number'];

                document.getElementById('edit-student-name').value = student['Student Name'];

                openModal('edit-student-modal');

            }

        }

        function openAddSubjectModal() {

            document.getElementById('add-subject-form').reset();

            openModal('add-subject-modal');

        }

        function openEditSubjectModal(subjectCode) {

            const subjects = JSON.parse(localStorage.getItem('subjectData'));

            const subject = subjects.find(s => s['Subject Code'] === subjectCode);

            if (subject) {

                document.getElementById('edit-subject-code').value = subject['Subject Code'];

                document.getElementById('edit-subject-name').value = subject['Subject Name'];

                openModal('edit-subject-modal');

            }

        }

        // Dashboard functions

        function loadDashboard() {

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

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

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

            // Update stats

            document.getElementById('total-students').textContent = students.length;

            document.getElementById('total-subjects').textContent = subjects.length;

            // Calculate attendance rate

            if (attendance.length > 0) {

                const presentCount = attendance.filter(a => a.status === 'Present').length;

                const attendanceRate = ((presentCount / attendance.length) \* 100).toFixed(1);

                document.getElementById('attendance-rate').textContent = `${attendanceRate}%`;

            } else {

                document.getElementById('attendance-rate').textContent = '0%';

            }

            // Calculate today's attendance

            const today = new Date().toISOString().split('T')[0];

            const todayAttendance = attendance.filter(a => a.date === today);

            document.getElementById('today-attendance').textContent = todayAttendance.length;

            // Load activity log

            loadActivityLog();

            // Create attendance chart

            createAttendanceChart();

        }

        function loadActivityLog() {

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

            const tbody = document.getElementById('activity-log');

            tbody.innerHTML = '';

            // Sort by date descending and take the last 10 entries

            const recentActivity = activityLog.sort((a, b) => new Date(b.date) - new Date(a.date)).slice(0, 10);

            recentActivity.forEach(activity => {

                const row = document.createElement('tr');

                const date = new Date(activity.date);

                row.innerHTML = `

                    <td>${date.toLocaleDateString()} ${date.toLocaleTimeString()}</td>

                    <td>${activity.activity}</td>

                    <td>${activity.details}</td>

                `;

                tbody.appendChild(row);

            });

        }

        function createAttendanceChart() {

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

            // Get last 7 days

            const dates = [];

            const presentCounts = [];

            const absentCounts = [];

            for (let i = 6; i >= 0; i--) {

                const date = new Date();

                date.setDate(date.getDate() - i);

                const dateString = date.toISOString().split('T')[0];

                const dayAttendance = attendance.filter(a => a.date === dateString);

                const presentCount = dayAttendance.filter(a => a.status === 'Present').length;

                const absentCount = dayAttendance.filter(a => a.status === 'Absent').length;

                dates.push(date.toLocaleDateString('en-US', { weekday: 'short', month: 'short', day: 'numeric' }));

                presentCounts.push(presentCount);

                absentCounts.push(absentCount);

            }

            const ctx = document.getElementById('attendanceChart').getContext('2d');

            if (window.attendanceOverviewChart) {

                window.attendanceOverviewChart.destroy();

            }

            window.attendanceOverviewChart = new Chart(ctx, {

                type: 'bar',

                data: {

                    labels: dates,

                    datasets: [

                        {

                            label: 'Present',

                            data: presentCounts,

                            backgroundColor: '#43a047',

                            borderColor: '#388e3c',

                            borderWidth: 1

                        },

                        {

                            label: 'Absent',

                            data: absentCounts,

                            backgroundColor: '#e53935',

                            borderColor: '#c62828',

                            borderWidth: 1

                        }

                    ]

                },

                options: {

                    responsive: true,

                    maintainAspectRatio: false,

                    scales: {

                        x: {

                            stacked: true

                        },

                        y: {

                            stacked: true,

                            beginAtZero: true,

                            title: {

                                display: true,

                                text: 'Number of Students'

                            }

                        }

                    },

                    plugins: {

                        title: {

                            display: true,

                            text: 'Attendance Overview (Last 7 Days)',

                            font: { size: 16 }

                        },

                        legend: {

                            position: 'bottom'

                        }

                    }

                }

            });

        }

        // Students functions

        function loadStudents() {

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

            const tbody = document.getElementById('students-table');

            tbody.innerHTML = '';

            students.forEach(student => {

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${student['Roll Number']}</td>

                    <td>${student['Student Name']}</td>

                    <td class="action-buttons">

                        <button class="btn btn-primary" onclick="openEditStudentModal('${student['Roll Number']}')">

                            <i class="fas fa-edit"></i>

                        </button>

                        <button class="btn btn-danger" onclick="deleteStudent('${student['Roll Number']}')">

                            <i class="fas fa-trash"></i>

                        </button>

                    </td>

                `;

                tbody.appendChild(row);

            });

        }

        function addStudent(event) {

            event.preventDefault();

            const rollNumber = document.getElementById('student-roll').value;

            const studentName = document.getElementById('student-name').value;

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

            // Check if roll number already exists

            if (students.some(s => s['Roll Number'] === rollNumber)) {

                alert('A student with this roll number already exists!');

                return;

            }

            // Add new student

            students.push({

                'Roll Number': rollNumber,

                'Student Name': studentName

            });

            localStorage.setItem('studentData', JSON.stringify(students));

            // Log activity

            logActivity('Student Added', `Added student ${studentName} with roll number ${rollNumber}`);

            // Reload students table

            loadStudents();

            // Close modal

            closeModal('add-student-modal');

        }

        function updateStudent(event) {

            event.preventDefault();

            const rollNumber = document.getElementById('edit-student-roll').value;

            const studentName = document.getElementById('edit-student-name').value;

            const students = JSON.parse(localStorage.getItem('studentData'));

            // Find and update student

            const updatedStudents = students.map(student => {

                if (student['Roll Number'] === rollNumber) {

                    return {

                        'Roll Number': rollNumber,

                        'Student Name': studentName

                    };

                }

                return student;

            });

            localStorage.setItem('studentData', JSON.stringify(updatedStudents));

            // Log activity

            logActivity('Student Updated', `Updated student with roll number ${rollNumber}`);

            // Reload students table

            loadStudents();

            // Close modal

            closeModal('edit-student-modal');

        }

        function deleteStudent(rollNumber) {

            if (!confirm(`Are you sure you want to delete student with roll number ${rollNumber}?`)) {

                return;

            }

            const students = JSON.parse(localStorage.getItem('studentData'));

            // Find student name for activity log

            const student = students.find(s => s['Roll Number'] === rollNumber);

            const studentName = student ? student['Student Name'] : 'Unknown';

            // Remove student

            const updatedStudents = students.filter(student => student['Roll Number'] !== rollNumber);

            localStorage.setItem('studentData', JSON.stringify(updatedStudents));

            // Log activity

            logActivity('Student Deleted', `Deleted student ${studentName} with roll number ${rollNumber}`);

            // Reload students table

            loadStudents();

        }

        // Subjects functions

        function loadSubjects() {

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

            const tbody = document.getElementById('subjects-table');

            tbody.innerHTML = '';

            subjects.forEach(subject => {

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${subject['Subject Code']}</td>

                    <td>${subject['Subject Name']}</td>

                    <td class="action-buttons">

                        <button class="btn btn-primary" onclick="openEditSubjectModal('${subject['Subject Code']}')">

                            <i class="fas fa-edit"></i>

                        </button>

                        <button class="btn btn-danger" onclick="deleteSubject('${subject['Subject Code']}')">

                            <i class="fas fa-trash"></i>

                        </button>

                    </td>

                `;

                tbody.appendChild(row);

            });

        }

        function addSubject(event) {

            event.preventDefault();

            const subjectCode = document.getElementById('subject-code').value;

            const subjectName = document.getElementById('subject-name').value;

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

            // Check if subject code already exists

            if (subjects.some(s => s['Subject Code'] === subjectCode)) {

                alert('A subject with this code already exists!');

                return;

            }

            // Add new subject

            subjects.push({

                'Subject Code': subjectCode,

                'Subject Name': subjectName

            });

            localStorage.setItem('subjectData', JSON.stringify(subjects));

            // Log activity

            logActivity('Subject Added', `Added subject ${subjectName} with code ${subjectCode}`);

            // Reload subjects table

            loadSubjects();

            // Close modal

            closeModal('add-subject-modal');

        }

        function updateSubject(event) {

            event.preventDefault();

            const subjectCode = document.getElementById('edit-subject-code').value;

            const subjectName = document.getElementById('edit-subject-name').value;

            const subjects = JSON.parse(localStorage.getItem('subjectData'));

            // Find and update subject

            const updatedSubjects = subjects.map(subject => {

                if (subject['Subject Code'] === subjectCode) {

                    return {

                        'Subject Code': subjectCode,

                        'Subject Name': subjectName

                    };

                }

                return subject;

            });

            localStorage.setItem('subjectData', JSON.stringify(updatedSubjects));

            // Log activity

            logActivity('Subject Updated', `Updated subject with code ${subjectCode}`);

            // Reload subjects table

            loadSubjects();

            // Close modal

            closeModal('edit-subject-modal');

        }

        function deleteSubject(subjectCode) {

            if (!confirm(`Are you sure you want to delete subject with code ${subjectCode}?`)) {

                return;

            }

            const subjects = JSON.parse(localStorage.getItem('subjectData'));

            // Find subject name for activity log

            const subject = subjects.find(s => s['Subject Code'] === subjectCode);

            const subjectName = subject ? subject['Subject Name'] : 'Unknown';

            // Remove subject

            const updatedSubjects = subjects.filter(subject => subject['Subject Code'] !== subjectCode);

            localStorage.setItem('subjectData', JSON.stringify(updatedSubjects));

            // Log activity

            logActivity('Subject Deleted', `Deleted subject ${subjectName} with code ${subjectCode}`);

            // Reload subjects table

            loadSubjects();

        }

        // Attendance functions

        function loadAttendance() {

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

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

            // Populate subject filter

            const subjectFilter = document.getElementById('subject-filter');

            subjectFilter.innerHTML = '<option value="">All Subjects</option>';

            subjects.forEach(subject => {

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

                option.value = subject['Subject Name'];

                option.textContent = subject['Subject Name'];

                subjectFilter.appendChild(option);

            });

            // Filter attendance records

            filterAttendance();

        }

        function filterAttendance() {

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

            const dateFilter = document.getElementById('attendance-date-filter').value;

            const subjectFilter = document.getElementById('subject-filter').value;

            let filteredAttendance = attendance;

            if (dateFilter) {

                filteredAttendance = filteredAttendance.filter(record => record.date === dateFilter);

            }

            if (subjectFilter) {

                filteredAttendance = filteredAttendance.filter(record => record.subject === subjectFilter);

            }

            // Display filtered records

            const tbody = document.getElementById('attendance-table');

            tbody.innerHTML = '';

            filteredAttendance.forEach(record => {

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${record.date}</td>

                    <td>${record.subject}</td>

                    <td>${record.rollNumber}</td>

                    <td>${record.studentName}</td>

                    <td>${record.status}</td>

                    <td class="action-buttons">

                        <button class="btn btn-danger" onclick="deleteAttendance('${record.date}', '${record.subject}', '${record.rollNumber}')">

                            <i class="fas fa-trash"></i>

                        </button>

                    </td>

                `;

                tbody.appendChild(row);

            });

        }

        function deleteAttendance(date, subject, rollNumber) {

            if (!confirm(`Are you sure you want to delete this attendance record?`)) {

                return;

            }

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

            // Remove attendance record

            const updatedAttendance = attendance.filter(record =>

                !(record.date === date && record.subject === subject && record.rollNumber === rollNumber)

            );

            localStorage.setItem('attendanceData', JSON.stringify(updatedAttendance));

            // Log activity

            logActivity('Attendance Deleted', `Deleted attendance record for ${subject} on ${date} for roll number ${rollNumber}`);

            // Reload attendance table

            filterAttendance();

        }

        // Reports functions

        function loadReportFilters() {

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

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

            // Populate subject filter

            const subjectFilter = document.getElementById('report-subject');

            subjectFilter.innerHTML = '<option value="">All Subjects</option>';

            subjects.forEach(subject => {

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

                option.value = subject['Subject Name'];

                option.textContent = subject['Subject Name'];

                subjectFilter.appendChild(option);

            });

            // Populate student filter

            const studentFilter = document.getElementById('report-student');

            studentFilter.innerHTML = '<option value="">All Students</option>';

            students.forEach(student => {

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

                option.value = student['Roll Number'];

                option.textContent = `${student['Roll Number']} - ${student['Student Name']}`;

                studentFilter.appendChild(option);

            });

        }

        function generateReport() {

            const reportType = document.getElementById('report-type').value;

            const startDate = document.getElementById('report-start-date').value;

            const endDate = document.getElementById('report-end-date').value;

            const subject = document.getElementById('report-subject').value;

            const student = document.getElementById('report-student').value;

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

            // Filter attendance records

            let filteredAttendance = attendance;

            if (startDate && endDate) {

                filteredAttendance = filteredAttendance.filter(record =>

                    record.date >= startDate && record.date <= endDate

                );

            }

            if (subject) {

                filteredAttendance = filteredAttendance.filter(record => record.subject === subject);

            }

            if (student) {

                filteredAttendance = filteredAttendance.filter(record => record.rollNumber === student);

            }

            // Generate report based on type

            if (reportType === 'daily') {

                generateDailyReport(filteredAttendance);

            } else if (reportType === 'monthly') {

                generateMonthlyReport(filteredAttendance);

            } else if (reportType === 'subject') {

                generateSubjectReport(filteredAttendance);

            } else if (reportType === 'student') {

                generateStudentReport(filteredAttendance);

            }

            // Log activity

            logActivity('Report Generated', `Generated ${reportType} report`);

        }

        function generateDailyReport(attendance) {

            // Group by date

            const groupedByDate = {};

            attendance.forEach(record => {

                if (!groupedByDate[record.date]) {

                    groupedByDate[record.date] = {

                        total: 0,

                        present: 0,

                        absent: 0

                    };

                }

                groupedByDate[record.date].total++;

                if (record.status === 'Present') {

                    groupedByDate[record.date].present++;

                } else {

                    groupedByDate[record.date].absent++;

                }

            });

            // Prepare data for chart and table

            const dates = Object.keys(groupedByDate).sort();

            const presentCounts = dates.map(date => groupedByDate[date].present);

            const absentCounts = dates.map(date => groupedByDate[date].absent);

            const attendanceRates = dates.map(date =>

                ((groupedByDate[date].present / groupedByDate[date].total) \* 100).toFixed(1)

            );

            // Create chart

            createReportChart('Daily Attendance Report', dates, presentCounts, absentCounts);

            // Create table

            const tableHead = document.querySelector('#report-table thead tr');

            tableHead.innerHTML = `

                <th>Date</th>

                <th>Total Students</th>

                <th>Present</th>

                <th>Absent</th>

                <th>Attendance Rate</th>

            `;

            const tableBody = document.querySelector('#report-table tbody');

            tableBody.innerHTML = '';

            dates.forEach((date, index) => {

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${date}</td>

                    <td>${groupedByDate[date].total}</td>

                    <td>${groupedByDate[date].present}</td>

                    <td>${groupedByDate[date].absent}</td>

                    <td>${attendanceRates[index]}%</td>

                `;

                tableBody.appendChild(row);

            });

        }

        function generateMonthlyReport(attendance) {

            // Group by month

            const groupedByMonth = {};

            attendance.forEach(record => {

                const month = record.date.substring(0, 7); // YYYY-MM

                if (!groupedByMonth[month]) {

                    groupedByMonth[month] = {

                        total: 0,

                        present: 0,

                        absent: 0

                    };

                }

                groupedByMonth[month].total++;

                if (record.status === 'Present') {

                    groupedByMonth[month].present++;

                } else {

                    groupedByMonth[month].absent++;

                }

            });

            // Prepare data for chart and table

            const months = Object.keys(groupedByMonth).sort();

            const presentCounts = months.map(month => groupedByMonth[month].present);

            const absentCounts = months.map(month => groupedByMonth[month].absent);

            const attendanceRates = months.map(month =>

                ((groupedByMonth[month].present / groupedByMonth[month].total) \* 100).toFixed(1)

            );

            // Format month labels

            const monthLabels = months.map(month => {

                const date = new Date(month + '-01');

                return date.toLocaleDateString('en-US', { year: 'numeric', month: 'long' });

            });

            // Create chart

            createReportChart('Monthly Attendance Report', monthLabels, presentCounts, absentCounts);

            // Create table

            const tableHead = document.querySelector('#report-table thead tr');

            tableHead.innerHTML = `

                <th>Month</th>

                <th>Total Records</th>

                <th>Present</th>

                <th>Absent</th>

                <th>Attendance Rate</th>

            `;

            const tableBody = document.querySelector('#report-table tbody');

            tableBody.innerHTML = '';

            months.forEach((month, index) => {

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${monthLabels[index]}</td>

                    <td>${groupedByMonth[month].total}</td>

                    <td>${groupedByMonth[month].present}</td>

                    <td>${groupedByMonth[month].absent}</td>

                    <td>${attendanceRates[index]}%</td>

                `;

                tableBody.appendChild(row);

            });

        }

        function generateSubjectReport(attendance) {

            // Group by subject

            const groupedBySubject = {};

            attendance.forEach(record => {

                if (!groupedBySubject[record.subject]) {

                    groupedBySubject[record.subject] = {

                        total: 0,

                        present: 0,

                        absent: 0

                    };

                }

                groupedBySubject[record.subject].total++;

                if (record.status === 'Present') {

                    groupedBySubject[record.subject].present++;

                } else {

                    groupedBySubject[record.subject].absent++;

                }

            });

            // Prepare data for chart and table

            const subjects = Object.keys(groupedBySubject).sort();

            const presentCounts = subjects.map(subject => groupedBySubject[subject].present);

            const absentCounts = subjects.map(subject => groupedBySubject[subject].absent);

            const attendanceRates = subjects.map(subject =>

                ((groupedBySubject[subject].present / groupedBySubject[subject].total) \* 100).toFixed(1)

            );

            // Create chart

            createReportChart('Subject-wise Attendance Report', subjects, presentCounts, absentCounts);

            // Create table

            const tableHead = document.querySelector('#report-table thead tr');

            tableHead.innerHTML = `

                <th>Subject</th>

                <th>Total Records</th>

                <th>Present</th>

                <th>Absent</th>

                <th>Attendance Rate</th>

            `;

            const tableBody = document.querySelector('#report-table tbody');

            tableBody.innerHTML = '';

            subjects.forEach((subject, index) => {

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${subject}</td>

                    <td>${groupedBySubject[subject].total}</td>

                    <td>${groupedBySubject[subject].present}</td>

                    <td>${groupedBySubject[subject].absent}</td>

                    <td>${attendanceRates[index]}%</td>

                `;

                tableBody.appendChild(row);

            });

        }

        function generateStudentReport(attendance) {

            // Group by student

            const groupedByStudent = {};

            attendance.forEach(record => {

                const studentKey = `${record.rollNumber} - ${record.studentName}`;

                if (!groupedByStudent[studentKey]) {

                    groupedByStudent[studentKey] = {

                        total: 0,

                        present: 0,

                        absent: 0

                    };

                }

                groupedByStudent[studentKey].total++;

                if (record.status === 'Present') {

                    groupedByStudent[studentKey].present++;

                } else {

                    groupedByStudent[studentKey].absent++;

                }

            });

            // Prepare data for chart and table

            const students = Object.keys(groupedByStudent).sort();

            const presentCounts = students.map(student => groupedByStudent[student].present);

            const absentCounts = students.map(student => groupedByStudent[student].absent);

            const attendanceRates = students.map(student =>

                ((groupedByStudent[student].present / groupedByStudent[student].total) \* 100).toFixed(1)

            );

            // Create chart (limit to top 10 students for readability)

            const top10Students = students.slice(0, 10);

            const top10Present = presentCounts.slice(0, 10);

            const top10Absent = absentCounts.slice(0, 10);

            createReportChart('Student-wise Attendance Report (Top 10)', top10Students, top10Present, top10Absent);

            // Create table

            const tableHead = document.querySelector('#report-table thead tr');

            tableHead.innerHTML = `

                <th>Student</th>

                <th>Total Records</th>

                <th>Present</th>

                <th>Absent</th>

                <th>Attendance Rate</th>

            `;

            const tableBody = document.querySelector('#report-table tbody');

            tableBody.innerHTML = '';

            students.forEach((student, index) => {

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${student}</td>

                    <td>${groupedByStudent[student].total}</td>

                    <td>${groupedByStudent[student].present}</td>

                    <td>${groupedByStudent[student].absent}</td>

                    <td>${attendanceRates[index]}%</td>

                `;

                tableBody.appendChild(row);

            });

        }

        function createReportChart(title, labels, presentData, absentData) {

            const ctx = document.getElementById('reportChart').getContext('2d');

            if (window.reportChart) {

                window.reportChart.destroy();

            }

            window.reportChart = new Chart(ctx, {

                type: 'bar',

                data: {

                    labels: labels,

                    datasets: [

                        {

                            label: 'Present',

                            data: presentData,

                            backgroundColor: '#43a047',

                            borderColor: '#388e3c',

                            borderWidth: 1

                        },

                        {

                            label: 'Absent',

                            data: absentData,

                            backgroundColor: '#e53935',

                            borderColor: '#c62828',

                            borderWidth: 1

                        }

                    ]

                },

                options: {

                    responsive: true,

                    maintainAspectRatio: false,

                    scales: {

                        x: {

                            stacked: true

                        },

                        y: {

                            stacked: true,

                            beginAtZero: true,

                            title: {

                                display: true,

                                text: 'Number of Students'

                            }

                        }

                    },

                    plugins: {

                        title: {

                            display: true,

                            text: title,

                            font: { size: 16 }

                        },

                        legend: {

                            position: 'bottom'

                        }

                    }

                }

            });

        }

        function exportReport() {

            const reportType = document.getElementById('report-type').value;

            const table = document.getElementById('report-table');

            // Get headers

            const headerCells = table.querySelectorAll('thead th');

            const headers = Array.from(headerCells).map(cell => cell.textContent);

            // Get data rows

            const rows = table.querySelectorAll('tbody tr');

            const data = [headers];

            rows.forEach(row => {

                const rowData = Array.from(row.cells).map(cell => {

                    // Remove % sign from percentage values

                    const text = cell.textContent;

                    return text.endsWith('%') ? text.slice(0, -1) : text;

                });

                data.push(rowData);

            });

            // Create workbook and worksheet

            const ws = XLSX.utils.aoa\_to\_sheet(data);

            const wb = XLSX.utils.book\_new();

            XLSX.utils.book\_append\_sheet(wb, ws, "Report");

            // Generate filename

            const date = new Date().toISOString().split('T')[0];

            const filename = `${reportType}\_report\_${date}.xlsx`;

            // Generate Excel file and trigger download

            XLSX.writeFile(wb, filename);

            // Log activity

            logActivity('Report Exported', `Exported ${reportType} report to Excel`);

        }

        // Utility functions

        function logActivity(activity, details) {

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

            activityLog.push({

                date: new Date().toISOString(),

                activity: activity,

                details: details

            });

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

        }

        function logout() {

            if (confirm('Are you sure you want to logout?')) {

                logActivity('Logout', 'Admin logged out');

                window.location.href = 'login.html';

            }

        }

        // Event listeners

        document.getElementById('add-student-form').addEventListener('submit', addStudent);

        document.getElementById('edit-student-form').addEventListener('submit', updateStudent);

        document.getElementById('add-subject-form').addEventListener('submit', addSubject);

        document.getElementById('edit-subject-form').addEventListener('submit', updateSubject);

        // Initialize dashboard

        loadDashboard();

    </script>

</body>

</html>

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Student Dashboard</title>

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

    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

</head>

<body>

    <div class="container">

        <nav class="sidebar">

            <div class="logo">

                <h2>Student Portal</h2>

            </div>

            <ul class="nav-links">

                <li class="active" data-tab="dashboard"><i class="fas fa-home"></i> Dashboard</li>

                <li data-tab="attendance"><i class="fas fa-calendar-check"></i> Attendance</li>

                <li data-tab="reports"><i class="fas fa-chart-bar"></i> Reports</li>

                <li data-tab="defaulters"><i class="fas fa-exclamation-triangle"></i> Defaulters</li>

                <li data-tab="notes"><i class="fas fa-sticky-note"></i> Notes</li>

            </ul>

            <div class="user-info">

                <img src="https://via.placeholder.com/50" alt="User">

                <div>

                    <h4>John Doe</h4>

                    <p>Student ID: S12345</p>

                </div>

            </div>

        </nav>

        <main class="content">

            <header>

                <div class="search-bar">

                    <i class="fas fa-search"></i>

                    <input type="text" placeholder="Search...">

                </div>

                <div class="header-right">

                    <div class="notifications">

                        <i class="fas fa-bell"></i>

                        <span class="badge">3</span>

                    </div>

                    <div class="settings">

                        <i class="fas fa-cog"></i>

                    </div>

                </div>

            </header>

            <div class="tab-content active" id="dashboard">

                <h1>Dashboard</h1>

                <div class="dashboard-summary">

                    <div class="summary-card">

                        <div class="card-icon attendance-icon">

                            <i class="fas fa-calendar-check"></i>

                        </div>

                        <div class="card-info">

                            <h3>Attendance</h3>

                            <p class="percentage">85%</p>

                        </div>

                    </div>

                    <div class="summary-card">

                        <div class="card-icon performance-icon">

                            <i class="fas fa-chart-line"></i>

                        </div>

                        <div class="card-info">

                            <h3>Performance</h3>

                            <p class="percentage">78%</p>

                        </div>

                    </div>

                    <div class="summary-card">

                        <div class="card-icon assignments-icon">

                            <i class="fas fa-tasks"></i>

                        </div>

                        <div class="card-info">

                            <h3>Assignments</h3>

                            <p>12/15 Completed</p>

                        </div>

                    </div>

                    <div class="summary-card">

                        <div class="card-icon exams-icon">

                            <i class="fas fa-file-alt"></i>

                        </div>

                        <div class="card-info">

                            <h3>Upcoming Exams</h3>

                            <p>2 Exams</p>

                        </div>

                    </div>

                </div>

                <div class="dashboard-charts">

                    <div class="chart-container">

                        <h2>Attendance Overview</h2>

                        <canvas id="attendanceChart"></canvas>

                    </div>

                    <div class="chart-container">

                        <h2>Performance Trend</h2>

                        <canvas id="performanceChart"></canvas>

                    </div>

                </div>

                <div class="recent-activities">

                    <h2>Recent Activities</h2>

                    <div class="activity-list">

                        <div class="activity-item">

                            <div class="activity-icon"><i class="fas fa-file-upload"></i></div>

                            <div class="activity-details">

                                <h4>Assignment Submitted</h4>

                                <p>You submitted Data Structure Assignment #5</p>

                                <span class="activity-time">2 hours ago</span>

                            </div>

                        </div>

                        <div class="activity-item">

                            <div class="activity-icon"><i class="fas fa-calendar-check"></i></div>

                            <div class="activity-details">

                                <h4>Attendance Marked</h4>

                                <p>You were marked present for all classes today</p>

                                <span class="activity-time">Yesterday</span>

                            </div>

                        </div>

                        <div class="activity-item">

                            <div class="activity-icon"><i class="fas fa-clipboard-check"></i></div>

                            <div class="activity-details">

                                <h4>Test Result</h4>

                                <p>You scored 85% in Mathematics Quiz</p>

                                <span class="activity-time">2 days ago</span>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

            <div class="tab-content" id="attendance">

                <h1>Attendance Record</h1>

                <div class="attendance-filters">

                    <select id="monthFilter">

                        <option value="all">All Months</option>

                        <option value="jan">January</option>

                        <option value="feb">February</option>

                        <option value="mar">March</option>

                        <option value="apr">April</option>

                        <option value="may">May</option>

                        <option value="jun">June</option>

                    </select>

                    <select id="subjectFilter">

                        <option value="all">All Subjects</option>

                        <option value="math">Mathematics</option>

                        <option value="physics">Physics</option>

                        <option value="chemistry">Chemistry</option>

                        <option value="biology">Biology</option>

                        <option value="english">English</option>

                    </select>

                </div>

                <div class="attendance-summary-cards">

                    <div class="summary-card">

                        <h3>Overall Attendance</h3>

                        <p class="percentage">85%</p>

                    </div>

                    <div class="summary-card">

                        <h3>Present Days</h3>

                        <p>34 days</p>

                    </div>

                    <div class="summary-card">

                        <h3>Absent Days</h3>

                        <p>6 days</p>

                    </div>

                    <div class="summary-card">

                        <h3>Leave Days</h3>

                        <p>2 days</p>

                    </div>

                </div>

                <div class="attendance-chart-container">

                    <h2>Monthly Attendance</h2>

                    <canvas id="monthlyAttendanceChart"></canvas>

                </div>

                <div class="attendance-table">

                    <h2>Detailed Attendance</h2>

                    <table>

                        <thead>

                            <tr>

                                <th>Date</th>

                                <th>Subject</th>

                                <th>Status</th>

                                <th>Remarks</th>

                            </tr>

                        </thead>

                        <tbody>

                            <tr>

                                <td>15 May 2023</td>

                                <td>Mathematics</td>

                                <td class="status-present">Present</td>

                                <td>-</td>

                            </tr>

                            <tr>

                                <td>15 May 2023</td>

                                <td>Physics</td>

                                <td class="status-present">Present</td>

                                <td>-</td>

                            </tr>

                            <tr>

                                <td>14 May 2023</td>

                                <td>Chemistry</td>

                                <td class="status-absent">Absent</td>

                                <td>Medical Leave</td>

                            </tr>

                            <tr>

                                <td>14 May 2023</td>

                                <td>Biology</td>

                                <td class="status-absent">Absent</td>

                                <td>Medical Leave</td>

                            </tr>

                            <tr>

                                <td>13 May 2023</td>

                                <td>Mathematics</td>

                                <td class="status-present">Present</td>

                                <td>-</td>

                            </tr>

                        </tbody>

                    </table>

                </div>

            </div>

            <div class="tab-content" id="reports">

                <h1>Academic Reports</h1>

                <div class="report-summary">

                    <div class="summary-card">

                        <h3>Current GPA</h3>

                        <p class="percentage">3.7/4.0</p>

                    </div>

                    <div class="summary-card">

                        <h3>Completed Credits</h3>

                        <p>45/120</p>

                    </div>

                    <div class="summary-card">

                        <h3>Assignments</h3>

                        <p>12/15 Completed</p>

                    </div>

                    <div class="summary-card">

                        <h3>Projects</h3>

                        <p>3/4 Completed</p>

                    </div>

                </div>

                <div class="report-charts">

                    <div class="chart-container">

                        <h2>Subject Performance</h2>

                        <canvas id="subjectPerformanceChart"></canvas>

                    </div>

                    <div class="chart-container">

                        <h2>GPA Trend</h2>

                        <canvas id="gpaTrendChart"></canvas>

                    </div>

                </div>

                <div class="report-table">

                    <h2>Recent Test Results</h2>

                    <table>

                        <thead>

                            <tr>

                                <th>Subject</th>

                                <th>Test Type</th>

                                <th>Date</th>

                                <th>Score</th>

                                <th>Class Average</th>

                            </tr>

                        </thead>

                        <tbody>

                            <tr>

                                <td>Data Sturture</td>

                                <td>Quiz</td>

                                <td>10 May 2023</td>

                                <td>85%</td>

                                <td>72%</td>

                            </tr>

                            <tr>

                                <td>DBMS</td>

                                <td>Mid-term</td>

                                <td>5 May 2023</td>

                                <td>78%</td>

                                <td>68%</td>

                            </tr>

                            <tr>

                                <td>WDT</td>

                                <td>Lab Report</td>

                                <td>2 May 2023</td>

                                <td>92%</td>

                                <td>75%</td>

                            </tr>

                            <tr>

                                <td>Biology</td>

                                <td>Assignment</td>

                                <td>28 Apr 2023</td>

                                <td>88%</td>

                                <td>80%</td>

                            </tr>

                            <tr>

                                <td>English</td>

                                <td>Essay</td>

                                <td>25 Apr 2023</td>

                                <td>75%</td>

                                <td>70%</td>

                            </tr>

                        </tbody>

                    </table>

                </div>

            </div>

            <div class="tab-content" id="defaulters">

                <h1>Attendance Defaulters</h1>

                <div class="defaulter-warning">

                    <i class="fas fa-exclamation-circle"></i>

                    <div>

                        <h3>Attendance Warning</h3>

                        <p>Your attendance in Data Structure is below the required threshold (75%). Please improve your attendance to avoid academic penalties.</p>

                    </div>

                </div>

                <div class="defaulter-subjects">

                    <div class="subject-card warning">

                        <h3>Data Sturture</h3>

                        <p class="percentage">68%</p>

                        <p class="required">Required: 75%</p>

                        <p class="shortage">7% below requirement</p>

                        <button class="action-btn">Request Consideration</button>

                    </div>

                    <div class="subject-card good">

                        <h3>DBMS</h3>

                        <p class="percentage">92%</p>

                        <p class="required">Required: 75%</p>

                        <p class="surplus">17% above requirement</p>

                    </div>

                    <div class="subject-card good">

                        <h3>Chemistry</h3>

                        <p class="percentage">85%</p>

                        <p class="required">Required: 75%</p>

                        <p class="surplus">10% above requirement</p>

                    </div>

                    <div class="subject-card good">

                        <h3>Biology</h3>

                        <p class="percentage">88%</p>

                        <p class="required">Required: 75%</p>

                        <p class="surplus">13% above requirement</p>

                    </div>

                </div>

                <div class="defaulter-history">

                    <h2>Attendance Improvement Plan</h2>

                    <div class="improvement-plan">

                        <div class="plan-item">

                            <h3>Current Status</h3>

                            <p>Your Data Structure attendance is 68%, which is 7% below the required 75%.</p>

                        </div>

                        <div class="plan-item">

                            <h3>Required Improvement</h3>

                            <p>You need to attend at least 8 consecutive Data Structure classes to reach the minimum requirement.</p>

                        </div>

                        <div class="plan-item">

                            <h3>Consequences</h3>

                            <p>Failure to meet the attendance requirement may result in being barred from the final examination.</p>

                        </div>

                        <div class="plan-item">

                            <h3>Support Available</h3>

                            <p>Contact your academic advisor for assistance or to discuss any issues affecting your attendance.</p>

                        </div>

                    </div>

                </div>

            </div>

            <div class="tab-content" id="notes">

                <h1>Notes & Reminders</h1>

                <div class="notes-container">

                    <div class="notes-sidebar">

                        <div class="notes-header">

                            <h3>My Notes</h3>

                            <button id="addNoteBtn"><i class="fas fa-plus"></i> New Note</button>

                        </div>

                        <div class="notes-list">

                            <div class="note-item active">

                                <h4>Physics Assignment</h4>

                                <p>Due: May 20, 2023</p>

                            </div>

                            <div class="note-item">

                                <h4>Math Formulas</h4>

                                <p>Updated: May 15, 2023</p>

                            </div>

                            <div class="note-item">

                                <h4>Chemistry Lab Report</h4>

                                <p>Due: May 25, 2023</p>

                            </div>

                            <div class="note-item">

                                <h4>Study Schedule</h4>

                                <p>Updated: May 10, 2023</p>

                            </div>

                        </div>

                    </div>

                    <div class="note-editor">

                        <div class="editor-header">

                            <input type="text" value="Physics Assignment" placeholder="Note Title">

                            <div class="editor-actions">

                                <button><i class="fas fa-save"></i> Save</button>

                                <button><i class="fas fa-trash"></i> Delete</button>

                            </div>

                        </div>

                        <textarea placeholder="Write your note here...">Complete the following problems:

1. Calculate the time complexity for merge sort algorithm.

2. Implement a binary search tree with insertion and deletion operations.

3. Explain the difference between stack and queue data structures.

Remember to show all work and include proper implementations.

Due Date: May 20, 2023

Submission: Online portal</textarea>

                    </div>

                </div>

            </div>

        </main>

    </div>

    <script src="script.js"></script>

</body>

</html>

**Login.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Login Page</title>

    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap" rel="stylesheet">

    <script src="https://unpkg.com/@tailwindcss/browser@4"></script>

    <style>

        :root {

            /\* Dark theme variables (default) \*/

            --bg-color: #0a0a0a;

            --bg-gradient: linear-gradient(to bottom right, #000000, #111827, #000000);

            --container-bg: rgba(255, 255, 255, 0.05);

            --container-border: rgba(255, 255, 255, 0.1);

            --text-color: #ffffff;

            --label-color: #e5e7eb;

            --tab-inactive-bg: rgba(255, 255, 255, 0.03);

            --tab-inactive-text: #d1d5db;

            --tab-hover-bg: rgba(255, 255, 255, 0.1);

            --tab-active-bg: linear-gradient(to bottom right, #1e40af, #3b82f6);

            --tab-active-border: #3b82f6;

            --input-bg: rgba(255, 255, 255, 0.05);

            --input-text: #ffffff;

            --input-border: rgba(255, 255, 255, 0.2);

            --input-focus-border: #3b82f6;

            --button-bg: linear-gradient(to right, #1e40af, #2563eb);

            --button-hover-bg: linear-gradient(to right, #2563eb, #60a5fa);

            --link-color: #d1d5db;

            --link-hover-color: #3b82f6;

            --shadow-color: rgba(0, 0, 0, 0.5);

        }

        /\* Light theme variables \*/

        [data-theme="light"] {

            --bg-color: #f5f5f5;

            --bg-gradient: linear-gradient(to bottom right, #ffffff, #e5e7eb, #ffffff);

            --container-bg: rgba(255, 255, 255, 0.9);

            --container-border: rgba(0, 0, 0, 0.1);

            --text-color: #1f2937;

            --label-color: #374151;

            --tab-inactive-bg: #e5e7eb;

            --tab-inactive-text: #4b5563;

            --tab-hover-bg: #d1d5db;

            --tab-active-bg: linear-gradient(to bottom right, #1e40af, #3b82f6);

            --tab-active-border: #3b82f6;

            --input-bg: #ffffff;

            --input-text: #1f2937;

            --input-border: rgba(0, 0, 0, 0.2);

            --input-focus-border: #3b82f6;

            --button-bg: linear-gradient(to right, #1e40af, #2563eb);

            --button-hover-bg: linear-gradient(to right, #2563eb, #60a5fa);

            --link-color: #4b5563;

            --link-hover-color: #3b82f6;

            --shadow-color: rgba(0, 0, 0, 0.1);

        }

        body {

            font-family: 'Inter', sans-serif;

            margin: 0;

            padding: 0;

            display: flex;

            justify-content: center;

            align-items: center;

            min-height: 100vh;

            background-color: var(--bg-color);

            background-image: var(--bg-gradient);

            transition: background-color 0.3s ease, background-image 0.3s ease;

        }

        .login-container {

            position: relative;

            background: var(--container-bg);

            border-radius: 1rem;

            box-shadow: 0 12px 24px var(--shadow-color);

            padding: 2.5rem;

            width: 100%;

            max-width: 32rem;

            backdrop-filter: blur(10px);

            border: 1px solid var(--container-border);

            transition: background 0.3s ease, border 0.3s ease;

        }

        .theme-toggle {

            position: absolute;

            top: 1rem;

            right: 1rem;

            padding: 0.5rem 1rem;

            border-radius: 0.5rem;

            font-weight: 600;

            color: var(--text-color);

            background-color: rgba(255, 255, 255, 0.1);

            border: 1px solid var(--input-border);

            cursor: pointer;

            transition: all 0.3s ease;

        }

        .theme-toggle:hover {

            background-color: var(--tab-hover-bg);

            transform: translateY(-2px);

        }

        .logo-container {

            display: flex;

            justify-content: center;

            margin-bottom: 1.5rem;

        }

        .logo {

            height: 4rem;

        }

        .title {

            font-size: 2.25rem;

            font-weight: 600;

            color: var(--text-color);

            margin-bottom: 1rem;

            text-align: center;

            text-shadow: 0 1px 3px var(--shadow-color);

            transition: color 0.3s ease;

        }

        .tab-buttons {

            display: flex;

            justify-content: center;

            margin-bottom: 1.5rem;

            gap: 0.75rem;

        }

        .tab-button {

            padding: 0.75rem 1.5rem;

            border-radius: 0.5rem;

            font-weight: 600;

            color: var(--tab-inactive-text);

            background-color: var(--tab-inactive-bg);

            transition: all 0.3s ease;

            cursor: pointer;

            border: 1px solid var(--input-border);

            box-shadow: 0 6px 12px var(--shadow-color);

            text-shadow: 0 1px 3px var(--shadow-color);

        }

        .tab-button:hover {

            background-color: var(--tab-hover-bg);

            transform: translateY(-2px);

            border-color: var(--input-focus-border);

        }

        .tab-button.active {

            background-image: var(--tab-active-bg);

            color: #ffffff;

            border-color: var(--tab-active-border);

            box-shadow: 0 8px 16px var(--shadow-color);

            transform: translateY(-4px);

        }

        .form-group {

            margin-bottom: 1.25rem;

        }

        .form-label {

            display: block;

            font-size: 0.875rem;

            font-weight: 600;

            color: var(--label-color);

            margin-bottom: 0.5rem;

            text-shadow: 0 1px 2px var(--shadow-color);

            transition: color 0.3s ease;

        }

        .input-field {

            padding: 0.75rem;

            border-radius: 0.5rem;

            border: 1px solid var(--input-border);

            width: 100%;

            background-color: var(--input-bg);

            color: var(--input-text);

            font-size: 1rem;

            transition: all 0.3s ease;

            outline: none;

            box-shadow: inset 0 2px 4px var(--shadow-color);

        }

        .input-field:focus {

            border-color: var(--input-focus-border);

            box-shadow: 0 6px 12px var(--shadow-color);

            transform: translateY(-2px);

        }

        .login-button {

            padding: 0.75rem 2rem;

            border-radius: 0.5rem;

            font-weight: 600;

            color: #ffffff;

            background-image: var(--button-bg);

            transition: all 0.3s ease;

            width: 100%;

            border: none;

            cursor: pointer;

            box-shadow: 0 6px 12px var(--shadow-color);

            text-shadow: 0 1px 3px var(--shadow-color);

        }

        .login-button:hover {

            background-image: var(--button-hover-bg);

            transform: translateY(-2px);

            box-shadow: 0 8px 16px var(--shadow-color);

        }

        .login-button:active {

            transform: translateY(0);

            box-shadow: 0 3px 6px var(--shadow-color);

        }

        .admin-login-link {

            margin-top: 1rem;

            text-align: center;

            font-size: 0.875rem;

            color: var(--link-color);

            transition: color 0.3s ease;

            text-shadow: 0 1px 2px var(--shadow-color);

        }

        .admin-login-link:hover {

            color: var(--link-hover-color);

        }

        .hidden {

            display: none;

        }

        .form-group.admin-code-group {

            display: none;

        }

    </style>

</head>

<body>

    <div class="login-container">

        <button class="theme-toggle" id="theme-toggle">Toggle Theme</button>

        <div class="logo-container">

        </div>

        <h2 class="title">AttendanceOS Login</h2>

        <div class="tab-buttons">

            <button class="tab-button student-tab active" data-user-type="student">Student</button>

            <button class="tab-button teacher-tab" data-user-type="teacher">Teacher</button>

            <button class="tab-button admin-tab" data-user-type="admin">Admin</button>

        </div>

        <form class="space-y-4" id="login-form">

            <div class="form-group">

                <label for="username" class="form-label">Username</label>

                <input type="text" id="username" name="username" placeholder="Enter your username" class="input-field">

            </div>

            <div class="form-group">

                <label for="password" class="form-label">Password</label>

                <input type="password" id="password" name="password" placeholder="Enter your password" class="input-field">

            </div>

            <div class="form-group admin-code-group">

                <label for="adminCode" class="form-label">Admin Code</label>

                <input type="text" id="adminCode" name="adminCode" placeholder="Enter admin authorization code" class="input-field">

            </div>

            <button type="submit" class="login-button">Login</button>

        </form>

        <div class="admin-login-link">

            <a href="#">Admin Login</a>

        </div>

    </div>

    <script>

        const tabButtons = document.querySelectorAll('.tab-button');

        const loginForm = document.getElementById('login-form');

        const adminCodeGroup = document.querySelector('.admin-code-group');

        const adminTabButton = document.querySelector('.admin-tab');

        const themeToggle = document.getElementById('theme-toggle');

        const body = document.body;

        let activeTab = 'student';

        let isDarkTheme = true;

        // Tab switching logic

        tabButtons.forEach(tab => {

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

                tabButtons.forEach(t => t.classList.remove('active'));

                this.classList.add('active');

                activeTab = this.dataset.userType;

                if (activeTab === 'admin') {

                    adminCodeGroup.style.display = 'block';

                } else {

                    adminCodeGroup.style.display = 'none';

                }

            });

        });

        // Form submission logic

        loginForm.addEventListener('submit', (event) => {

            event.preventDefault();

            const username = event.target.username.value;

            const password = event.target.password.value;

            const adminCode = event.target.adminCode ? event.target.adminCode.value : '';

            if (activeTab === 'admin') {

                if (!adminCode) {

                    alert('Please enter the admin code.');

                    return;

                }

            }

            console.log('Logging in as:', activeTab);

            console.log('Username:', username);

            console.log('Password:', password);

            if (activeTab === 'admin') {

                console.log('Admin code:', adminCode);

            }

            alert('Logged in as ' + activeTab + '!');

            // Redirect based on user type

            if (activeTab === 'teacher') {

                window.location.href = 'teacher.html';

            } else if (activeTab === 'student') {

                window.location.href = 'index.html';

            } else {

                window.location.href = 'admin.html';

            }

        });

        // Redundant admin tab listener (kept for consistency with original)

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

            adminCodeGroup.style.display = 'block';

        });

        // Theme toggle logic

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

            if (isDarkTheme) {

                body.setAttribute('data-theme', 'light');

                themeToggle.textContent = 'Dark Theme';

            } else {

                body.removeAttribute('data-theme');

                themeToggle.textContent = 'Light Theme';

            }

            isDarkTheme = !isDarkTheme;

        });

    </script>

</body>

</html>

**Script.js**

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

    // Tab Navigation

    const navLinks = document.querySelectorAll('.nav-links li');

    const tabContents = document.querySelectorAll('.tab-content');

    navLinks.forEach(link => {

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

            // Remove active class from all links and tabs

            navLinks.forEach(item => item.classList.remove('active'));

            tabContents.forEach(tab => tab.classList.remove('active'));

            // Add active class to clicked link

            this.classList.add('active');

            // Show corresponding tab content

            const tabId = this.getAttribute('data-tab');

            document.getElementById(tabId).classList.add('active');

        });

    });

    // Initialize Charts

    initializeAttendanceChart();

    initializePerformanceChart();

    initializeMonthlyAttendanceChart();

    initializeSubjectPerformanceChart();

    initializeGpaTrendChart();

    // Notes Functionality

    const noteItems = document.querySelectorAll('.note-item');

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

    noteItems.forEach(item => {

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

            noteItems.forEach(note => note.classList.remove('active'));

            this.classList.add('active');

            // In a real app, you would load the note content here

        });

    });

    if (addNoteBtn) {

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

            // In a real app, you would create a new note here

            alert('Creating a new note...');

        });

    }

});

// Chart Initialization Functions

function initializeAttendanceChart() {

    const ctx = document.getElementById('attendanceChart');

    if (!ctx) return;

    new Chart(ctx, {

        type: 'line',

        data: {

            labels: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'],

            datasets: [{

                label: 'Attendance %',

                data: [92, 88, 95, 89, 91, 87],

                backgroundColor: 'rgba(67, 97, 238, 0.1)',

                borderColor: 'rgba(67, 97, 238, 1)',

                borderWidth: 2,

                tension: 0.4,

                fill: true

            }]

        },

        options: {

            responsive: true,

            maintainAspectRatio: false,

            scales: {

                y: {

                    beginAtZero: false,

                    min: 70,

                    max: 100

                }

            }

        }

    });

}

function initializePerformanceChart() {

    const ctx = document.getElementById('performanceChart');

    if (!ctx) return;

    new Chart(ctx, {

        type: 'bar',

        data: {

            labels: ['Quiz 1', 'Assignment 1', 'Mid-term', 'Quiz 2', 'Assignment 2', 'Final'],

            datasets: [{

                label: 'Score %',

                data: [85, 92, 78, 88, 90, 82],

                backgroundColor: [

                    'rgba(76, 175, 80, 0.7)',

                    'rgba(76, 175, 80, 0.7)',

                    'rgba(255, 152, 0, 0.7)',

                    'rgba(76, 175, 80, 0.7)',

                    'rgba(76, 175, 80, 0.7)',

                    'rgba(76, 175, 80, 0.7)'

                ],

                borderColor: [

                    'rgba(76, 175, 80, 1)',

                    'rgba(76, 175, 80, 1)',

                    'rgba(255, 152, 0, 1)',

                    'rgba(76, 175, 80, 1)',

                    'rgba(76, 175, 80, 1)',

                    'rgba(76, 175, 80, 1)'

                ],

                borderWidth: 1

            }]

        },

        options: {

            responsive: true,

            maintainAspectRatio: false,

            scales: {

                y: {

                    beginAtZero: true,

                    max: 100

                }

            }

        }

    });

}

function initializeMonthlyAttendanceChart() {

    const ctx = document.getElementById('monthlyAttendanceChart');

    if (!ctx) return;

    new Chart(ctx, {

        type: 'line',

        data: {

            labels: ['Week 1', 'Week 2', 'Week 3', 'Week 4'],

            datasets: [

                {

                    label: 'Mathematics',

                    data: [100, 80, 100, 100],

                    borderColor: 'rgba(67, 97, 238, 1)',

                    backgroundColor: 'rgba(67, 97, 238, 0.1)',

                    tension: 0.3

                },

                {

                    label: 'Physics',

                    data: [80, 60, 80, 60],

                    borderColor: 'rgba(244, 67, 54, 1)',

                    backgroundColor: 'rgba(244, 67, 54, 0.1)',

                    tension: 0.3

                },

                {

                    label: 'Chemistry',

                    data: [100, 100, 80, 100],

                    borderColor: 'rgba(76, 175, 80, 1)',

                    backgroundColor: 'rgba(76, 175, 80, 0.1)',

                    tension: 0.3

                }

            ]

        },

        options: {

            responsive: true,

            maintainAspectRatio: false,

            scales: {

                y: {

                    beginAtZero: true,

                    max: 100,

                    title: {

                        display: true,

                        text: 'Attendance %'

                    }

                }

            }

        }

    });

}

function initializeSubjectPerformanceChart() {

    const ctx = document.getElementById('subjectPerformanceChart');

    if (!ctx) return;

    new Chart(ctx, {

        type: 'radar',

        data: {

            labels: ['Mathematics', 'Physics', 'Chemistry', 'Biology', 'English'],

            datasets: [{

                label: 'Current Performance',

                data: [85, 78, 92, 88, 75],

                backgroundColor: 'rgba(67, 97, 238, 0.2)',

                borderColor: 'rgba(67, 97, 238, 1)',

                borderWidth: 2,

                pointBackgroundColor: 'rgba(67, 97, 238, 1)'

            }]

        },

        options: {

            responsive: true,

            maintainAspectRatio: false,

            scales: {

                r: {

                    angleLines: {

                        display: true

                    },

                    suggestedMin: 50,

                    suggestedMax: 100

                }

            }

        }

    });

}

function initializeGpaTrendChart() {

    const ctx = document.getElementById('gpaTrendChart');

    if (!ctx) return;

    new Chart(ctx, {

        type: 'line',

        data: {

            labels: ['Semester 1', 'Semester 2', 'Semester 3', 'Current'],

            datasets: [{

                label: 'GPA',

                data: [3.5, 3.6, 3.8, 3.7],

                backgroundColor: 'rgba(67, 97, 238, 0.1)',

                borderColor: 'rgba(67, 97, 238, 1)',

                borderWidth: 2,

                tension: 0.4,

                fill: true

            }]

        },

        options: {

            responsive: true,

            maintainAspectRatio: false,

            scales: {

                y: {

                    beginAtZero: false,

                    min: 2.0,

                    max: 4.0,

                    title: {

                        display: true,

                        text: 'GPA (4.0 Scale)'

                    }

                }

            }

        }

    });

}

**Styles.css**

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

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

}

:root {

    --primary-color: #4361ee;

    --secondary-color: #3f37c9;

    --accent-color: #4895ef;

    --text-color: #333;

    --light-text: #666;

    --background-color: #f5f7fb;

    --card-color: #ffffff;

    --success-color: #4caf50;

    --warning-color: #ff9800;

    --danger-color: #f44336;

    --border-radius: 10px;

    --box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

body {

    background-color: var(--background-color);

    color: var(--text-color);

}

.container {

    display: flex;

    min-height: 100vh;

}

/\* Sidebar Styles \*/

.sidebar {

    width: 250px;

    background-color: var(--primary-color);

    color: white;

    padding: 20px;

    display: flex;

    flex-direction: column;

}

.logo h2 {

    margin-bottom: 30px;

    font-weight: 600;

}

.nav-links {

    list-style: none;

    margin-bottom: auto;

}

.nav-links li {

    padding: 12px 15px;

    margin-bottom: 5px;

    border-radius: var(--border-radius);

    cursor: pointer;

    display: flex;

    align-items: center;

    transition: background-color 0.3s;

}

.nav-links li i {

    margin-right: 10px;

    font-size: 1.1rem;

}

.nav-links li:hover, .nav-links li.active {

    background-color: var(--secondary-color);

}

.user-info {

    display: flex;

    align-items: center;

    padding: 15px 0;

    border-top: 1px solid rgba(255, 255, 255, 0.1);

}

.user-info img {

    width: 40px;

    height: 40px;

    border-radius: 50%;

    margin-right: 10px;

}

.user-info h4 {

    font-size: 0.9rem;

    margin-bottom: 3px;

}

.user-info p {

    font-size: 0.8rem;

    opacity: 0.8;

}

/\* Content Styles \*/

.content {

    flex: 1;

    padding: 20px;

    overflow-y: auto;

}

header {

    display: flex;

    justify-content: space-between;

    align-items: center;

    margin-bottom: 30px;

    padding-bottom: 15px;

    border-bottom: 1px solid #eee;

}

.search-bar {

    display: flex;

    align-items: center;

    background-color: var(--card-color);

    border-radius: 20px;

    padding: 8px 15px;

    width: 300px;

    box-shadow: var(--box-shadow);

}

.search-bar i {

    color: var(--light-text);

    margin-right: 10px;

}

.search-bar input {

    border: none;

    outline: none;

    background: transparent;

    width: 100%;

}

.header-right {

    display: flex;

    align-items: center;

}

.notifications, .settings {

    position: relative;

    margin-left: 20px;

    cursor: pointer;

}

.notifications i, .settings i {

    font-size: 1.2rem;

    color: var(--light-text);

}

.badge {

    position: absolute;

    top: -5px;

    right: -5px;

    background-color: var(--danger-color);

    color: white;

    font-size: 0.7rem;

    width: 18px;

    height: 18px;

    border-radius: 50%;

    display: flex;

    align-items: center;

    justify-content: center;

}

/\* Tab Content Styles \*/

.tab-content {

    display: none;

}

.tab-content.active {

    display: block;

}

h1 {

    margin-bottom: 20px;

    color: var(--text-color);

}

h2 {

    margin-bottom: 15px;

    color: var(--text-color);

    font-size: 1.3rem;

}

/\* Dashboard Styles \*/

.dashboard-summary {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));

    gap: 20px;

    margin-bottom: 30px;

}

.summary-card {

    background-color: var(--card-color);

    border-radius: var(--border-radius);

    padding: 20px;

    box-shadow: var(--box-shadow);

    display: flex;

    align-items: center;

}

.card-icon {

    width: 50px;

    height: 50px;

    border-radius: 50%;

    display: flex;

    align-items: center;

    justify-content: center;

    margin-right: 15px;

}

.card-icon i {

    font-size: 1.5rem;

    color: white;

}

.attendance-icon {

    background-color: var(--primary-color);

}

.performance-icon {

    background-color: var(--success-color);

}

.assignments-icon {

    background-color: var(--warning-color);

}

.exams-icon {

    background-color: var(--danger-color);

}

.card-info h3 {

    font-size: 1rem;

    margin-bottom: 5px;

    color: var(--light-text);

}

.percentage {

    font-size: 1.5rem;

    font-weight: bold;

    color: var(--primary-color);

}

.dashboard-charts {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(450px, 1fr));

    gap: 20px;

    margin-bottom: 30px;

}

.chart-container {

    background-color: var(--card-color);

    border-radius: var(--border-radius);

    padding: 20px;

    box-shadow: var(--box-shadow);

}

.chart-container canvas {

    width: 100%;

    height: 250px;

}

/\* Canvas dimensions \*/

canvas {

    max-width: 100%;

}

canvas#attendanceChart,

canvas#performanceChart,

canvas#monthlyAttendanceChart,

canvas#subjectPerformanceChart,

canvas#gpaTrendChart {

    height: 250px !important;

    width: 100% !important;

}

.attendance-chart-container {

    background-color: var(--card-color);

    border-radius: var(--border-radius);

    padding: 20px;

    box-shadow: var(--box-shadow);

    margin-bottom: 30px;

}

.attendance-table {

    background-color: var(--card-color);

    border-radius: var(--border-radius);

    padding: 20px;

    box-shadow: var(--box-shadow);

}

table {

    width: 100%;

    border-collapse: collapse;

    margin-top: 15px;

}

table th, table td {

    padding: 12px 15px;

    text-align: left;

    border-bottom: 1px solid #eee;

}

table th {

    font-weight: 600;

    color: var(--light-text);

}

.status-present {

    color: var(--success-color);

    font-weight: 500;

}

.status-absent {

    color: var(--danger-color);

    font-weight: 500;

}

/\* Reports Styles \*/

.report-summary {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

    gap: 20px;

    margin-bottom: 30px;

}

.report-charts {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(450px, 1fr));

    gap: 20px;

    margin-bottom: 30px;

}

.report-table {

    background-color: var(--card-color);

    border-radius: var(--border-radius);

    padding: 20px;

    box-shadow: var(--box-shadow);

}

/\* Defaulters Styles \*/

.defaulter-warning {

    background-color: rgba(244, 67, 54, 0.1);

    border-left: 4px solid var(--danger-color);

    padding: 15px;

    border-radius: 5px;

    display: flex;

    align-items: center;

    margin-bottom: 30px;

}

.defaulter-warning i {

    font-size: 2rem;

    color: var(--danger-color);

    margin-right: 15px;

}

.defaulter-subjects {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));

    gap: 20px;

    margin-bottom: 30px;

}

.subject-card {

    background-color: var(--card-color);

    border-radius: var(--border-radius);

    padding: 20px;

    box-shadow: var(--box-shadow);

    border-top: 4px solid;

}

.subject-card.warning {

    border-top-color: var(--danger-color);

}

.subject-card.good {

    border-top-color: var(--success-color);

}

.subject-card h3 {

    margin-bottom: 15px;

}

.subject-card .percentage {

    font-size: 2rem;

    margin-bottom: 10px;

}

.subject-card.warning .percentage {

    color: var(--danger-color);

}

.subject-card.good .percentage {

    color: var(--success-color);

}

.subject-card .required {

    color: var(--light-text);

    margin-bottom: 5px;

}

.subject-card .shortage {

    color: var(--danger-color);

    margin-bottom: 15px;

}

.subject-card .surplus {

    color: var(--success-color);

}

.action-btn {

    background-color: var(--primary-color);

    color: white;

    border: none;

    padding: 8px 15px;

    border-radius: 5px;

    cursor: pointer;

    font-size: 0.9rem;

    margin-top: 10px;

}

.defaulter-history {

    background-color: var(--card-color);

    border-radius: var(--border-radius);

    padding: 20px;

    box-shadow: var(--box-shadow);

}

.improvement-plan {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

    gap: 20px;

    margin-top: 15px;

}

.plan-item {

    padding: 15px;

    background-color: rgba(67, 97, 238, 0.05);

    border-radius: 5px;

}

.plan-item h3 {

    font-size: 1rem;

    margin-bottom: 10px;

    color: var(--primary-color);

}

/\* Notes Styles \*/

.notes-container {

    display: flex;

    background-color: var(--card-color);

    border-radius: var(--border-radius);

    box-shadow: var(--box-shadow);

    height: 500px;

}

.notes-sidebar {

    width: 300px;

    border-right: 1px solid #eee;

    padding: 20px;

    overflow-y: auto;

}

.notes-header {

    display: flex;

    justify-content: space-between;

    align-items: center;

    margin-bottom: 20px;

}

#addNoteBtn {

    background-color: var(--primary-color);

    color: white;

    border: none;

    padding: 8px 12px;

    border-radius: 5px;

    cursor: pointer;

    font-size: 0.9rem;

    display: flex;

    align-items: center;

}

#addNoteBtn i {

    margin-right: 5px;

}

.notes-list {

    display: flex;

    flex-direction: column;

    gap: 10px;

}

.note-item {

    padding: 12px;

    border-radius: 5px;

    cursor: pointer;

    transition: background-color 0.3s;

}

.note-item:hover {

    background-color: rgba(67, 97, 238, 0.05);

}

.note-item.active {

    background-color: rgba(67, 97, 238, 0.1);

    border-left: 3px solid var(--primary-color);

}

.note-item h4 {

    margin-bottom: 5px;

    font-size: 1rem;

}

.note-item p {

    font-size: 0.8rem;

    color: var(--light-text);

}

.note-editor {

    flex: 1;

    display: flex;

    flex-direction: column;

    padding: 20px;

}

.editor-header {

    display: flex;

    justify-content: space-between;

    align-items: center;

    margin-bottom: 20px;

    padding-bottom: 15px;

    border-bottom: 1px solid #eee;

}

.editor-header input {

    border: none;

    outline: none;

    font-size: 1.2rem;

    font-weight: 600;

    width: 60%;

}

.editor-actions {

    display: flex;

    gap: 10px;

}

.editor-actions button {

    padding: 8px 12px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    display: flex;

    align-items: center;

    font-size: 0.9rem;

}

.editor-actions button:first-child {

    background-color: var(--primary-color);

    color: white;

}

.editor-actions button:last-child {

    background-color: rgba(244, 67, 54, 0.1);

    color: var(--danger-color);

}

.editor-actions button i {

    margin-right: 5px;

}

.note-editor textarea {

    flex: 1;

    border: none;

    outline: none;

    resize: none;

    font-size: 1rem;

    line-height: 1.6;

}

/\* Responsive Styles \*/

@media (max-width: 1024px) {

    .dashboard-charts, .report-charts {

        grid-template-columns: 1fr;

    }

}

@media (max-width: 768px) {

    .container {

        flex-direction: column;

    }

    .sidebar {

        width: 100%;

        padding: 15px;

    }

    .nav-links {

        display: flex;

        overflow-x: auto;

        margin-bottom: 15px;

    }

    .nav-links li {

        white-space: nowrap;

    }

    .user-info {

        display: none;

    }

    .dashboard-summary, .attendance-summary-cards, .report-summary, .defaulter-subjects {

        grid-template-columns: repeat(auto-fit, minmax(100%, 1fr));

    }

    .notes-container {

        flex-direction: column;

        height: auto;

    }

    .notes-sidebar {

        width: 100%;

        border-right: none;

        border-bottom: 1px solid #eee;

    }

    .note-editor {

        height: 400px;

    }

}

**Teacher.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Student Roll Numbers</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 20px;

            background-color: #f5f5f5;

        }

        .container {

            max-width: 800px;

            margin: 0 auto;

            background-color: white;

            padding: 20px;

            border-radius: 8px;

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

        }

        table {

            width: 100%;

            border-collapse: collapse;

            margin: 20px 0;

        }

        th, td {

            border: 1px solid #ddd;

            padding: 12px;

            text-align: left;

        }

        th {

            background-color: #f8f9fa;

        }

        .button-group {

            margin-bottom: 20px;

            display: flex;

            gap: 10px;

        }

        .import-btn, .export-btn, .attendance-btn {

            padding: 10px 20px;

            border: none;

            border-radius: 4px;

            cursor: pointer;

            font-size: 16px;

        }

        .import-btn {

            background-color: #007bff;

            color: white;

        }

        .export-btn {

            background-color: #28a745;

            color: white;

        }

        .attendance-btn {

            background-color: #6c757d;

            color: white;

        }

        .import-btn:hover {

            background-color: #0056b3;

        }

        .export-btn:hover {

            background-color: #218838;

        }

        .attendance-btn:hover {

            background-color: #5a6268;

        }

        .tab-content {

            display: none;

        }

        .tab-content.active {

            display: block;

        }

        .status-present {

            background-color: #d4edda;

        }

        .status-absent {

            background-color: #f8d7da;

        }

        .status-late {

            background-color: #fff3cd;

        }

        .attendance-header {

            display: flex;

            justify-content: space-between;

            align-items: center;

            margin-bottom: 15px;

        }

        .attendance-filters {

            display: flex;

            gap: 10px;

            margin-bottom: 15px;

        }

        .attendance-filters select, .attendance-filters input {

            padding: 8px;

            border: 1px solid #ddd;

            border-radius: 4px;

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>Teacher Dashboard</h1>

        <div class="button-group">

            <button class="attendance-btn" onclick="showTab('students')">Students</button>

            <button class="attendance-btn" onclick="showTab('daily-attendance')">Daily Attendance</button>

            <button class="attendance-btn" onclick="showTab('monthly-report')">Monthly Report</button>

        </div>

        <!-- Students Tab -->

        <div id="students" class="tab-content active">

            <div class="button-group">

                <input type="file" id="fileInput" accept=".xlsx, .xls" style="display: none;" onchange="importFromExcel(this)">

                <button class="import-btn" onclick="document.getElementById('fileInput').click()">Import Excel</button>

                <button class="export-btn" onclick="exportToExcel()">Export to Excel</button>

                <button class="import-btn" onclick="toggleAddForm()">Add Student</button>

            </div>

            <!-- Add Student Form -->

            <div id="addStudentForm" style="display: none; margin-bottom: 20px; padding: 15px; background-color: #f8f9fa; border-radius: 4px;">

                <div style="display: grid; grid-template-columns: repeat(2, 1fr); gap: 10px; margin-bottom: 10px;">

                    <input type="text" id="rollNumber" placeholder="Roll Number" style="padding: 8px; border: 1px solid #ddd; border-radius: 4px;">

                    <input type="text" id="studentName" placeholder="Student Name" style="padding: 8px; border: 1px solid #ddd; border-radius: 4px;">

                </div>

                <button onclick="addStudent()" style="padding: 8px 16px; background-color: #28a745; color: white; border: none; border-radius: 4px; cursor: pointer;">Save</button>

            </div>

            <table id="studentTable">

                <thead>

                    <tr>

                        <th>Roll Number</th>

                        <th>Student Name</th>

                        <th>Class</th>

                        <th>Section</th>

                        <th>Date</th>

                        <th>Action</th>

                    </tr>

                </thead>

                <tbody>

                </tbody>

            </table>

        </div>

        <!-- Daily Attendance Tab -->

        <div id="daily-attendance" class="tab-content">

            <div class="attendance-header">

                <h2>Daily Attendance</h2>

                <button class="export-btn" onclick="exportDailyAttendance()">Export Attendance</button>

            </div>

            <div class="attendance-filters">

                <input type="date" id="attendanceDate" value="">

                <select id="subjectSelect">

                    <option value="">Select Subject</option>

                    <option value="Java">Java</option>

                    <option value="Python">Python</option>

                    <option value="Web Development">Web Development</option>

                    <option value="Database">Database</option>

                    <option value="Operating System">Operating System</option>

                </select>

                <button class="import-btn" onclick="loadAttendanceSheet()">Load Students</button>

            </div>

            <table id="attendanceTable">

                <thead>

                    <tr>

                        <th>Roll Number</th>

                        <th>Student Name</th>

                        <th>Status</th>

                        <th>Remarks</th>

                    </tr>

                </thead>

                <tbody>

                </tbody>

            </table>

            <div style="margin-top: 15px; text-align: right;">

                <button class="export-btn" onclick="saveAttendance()">Save Attendance</button>

            </div>

        </div>

        <!-- Monthly Report Tab -->

        <div id="monthly-report" class="tab-content">

            <div class="attendance-header">

                <h2>Monthly Attendance Report</h2>

                <button class="export-btn" onclick="exportMonthlyReport()">Export Report</button>

            </div>

            <div class="attendance-filters">

                <select id="monthPicker" onchange="generateMonthlyReport()">

                    <option value="2023-01">January 2023</option>

                    <option value="2023-02">February 2023</option>

                    <option value="2023-03">March 2023</option>

                    <option value="2023-04">April 2023</option>

                    <option value="2023-05">May 2023</option>

                    <option value="2023-06">June 2023</option>

                    <option value="2023-07">July 2023</option>

                    <option value="2023-08">August 2023</option>

                    <option value="2023-09">September 2023</option>

                    <option value="2023-10">October 2023</option>

                    <option value="2023-11">November 2023</option>

                    <option value="2023-12">December 2023</option>

                </select>

                <select id="monthlySubjectSelect" onchange="generateMonthlyReport()">

                    <option value="">All Subjects</option>

                    <option value="Java">Java</option>

                    <option value="Python">Python</option>

                    <option value="Web Development">Web Development</option>

                    <option value="Database">Database</option>

                    <option value="Operating System">Operating System</option>

                </select>

            </div>

            <table id="monthlyReportTable">

                <thead>

                    <tr>

                        <th>Roll Number</th>

                        <th>Student Name</th>

                        <th>Total Classes</th>

                        <th>Present</th>

                        <th>Absent</th>

                        <th>Attendance %</th>

                    </tr>

                </thead>

                <tbody>

                </tbody>

            </table>

        </div>

        <script>

            function importFromExcel(input) {

                const file = input.files[0];

                const reader = new FileReader();

                reader.onload = function(e) {

                    const data = new Uint8Array(e.target.result);

                    const workbook = XLSX.read(data, { type: 'array' });

                    const firstSheet = workbook.Sheets[workbook.SheetNames[0]];

                    const jsonData = XLSX.utils.sheet\_to\_json(firstSheet, {

                        raw: false,

                        dateNF: 'yyyy-mm-dd',

                        header: ['Roll Number', 'Student Name', 'Class', 'Section'],

                        defval: ''

                    });

                    const tbody = document.querySelector('#studentTable tbody');

                    tbody.innerHTML = '';

                    jsonData.forEach((record, index) => {

                        if (index > 0) {

                            const today = new Date().toISOString().split('T')[0];

                            const row = document.createElement('tr');

                            row.innerHTML = `

                                <td>${record['Roll Number'] || ''}</td>

                                <td>${record['Student Name'] || ''}</td>

                                <td>${record['Class'] || 'BCA'}</td>

                                <td>${record['Section'] || 'Management'}</td>

                                <td>${today}</td>

                                <td><button onclick="deleteStudent('${record['Roll Number'] || ''}')"

                                    style="background-color: #dc3545; color: white; border: none;

                                    border-radius: 4px; padding: 5px 10px; cursor: pointer;">Delete</button></td>

                            `;

                            tbody.appendChild(row);

                        }

                    });

                    // Update storage data with default values

                    localStorage.setItem('studentData', JSON.stringify(jsonData.slice(1).map(record => ({

                        'Roll Number': record['Roll Number'] || '',

                        'Student Name': record['Student Name'] || '',

                        'Class': record['Class'] || 'BCA',

                        'Section': record['Section'] || 'Management'

                    }))));

                    alert('Data imported successfully!');

                };

                reader.readAsArrayBuffer(file);

            }

            window.onload = function() {

                // Set today's date as default for attendance

                document.getElementById('attendanceDate').valueAsDate = new Date();

                // Set current month as default for monthly report

                const now = new Date();

                const month = now.getMonth() + 1;

                const year = now.getFullYear();

                const monthStr = month < 10 ? `0${month}` : month;

                document.getElementById('monthPicker').value = `${year}-${monthStr}`;

                const studentData = localStorage.getItem('studentData');

                if (studentData) {

                    const tbody = document.querySelector('#studentTable tbody');

                    JSON.parse(studentData).forEach(record => {

                        const row = document.createElement('tr');

                        row.innerHTML = `

                            <td>${record['Roll Number'] || ''}</td>

                            <td>${record['Student Name'] || ''}</td>

                            <td>${record['Class'] || 'BCA'}</td>

                            <td>${record['Section'] || 'Management'}</td>

                            <td>${record['Date'] || new Date().toISOString().split('T')[0]}</td>

                            <td><button onclick="deleteStudent('${record['Roll Number']}')"

                                style="background-color: #dc3545; color: white; border: none;

                                border-radius: 4px; padding: 5px 10px; cursor: pointer;">Delete</button></td>

                        `;

                        tbody.appendChild(row);

                    });

                } else {

                    const tbody = document.querySelector('#studentTable tbody');

                    const row = document.createElement('tr');

                    const today = new Date().toISOString().split('T')[0];

                    row.innerHTML = `

                        <td></td>

                        <td></td>

                        <td>BCA</td>

                        <td>Management</td>

                        <td>${today}</td>

                        <td></td>

                    `;

                    tbody.appendChild(row);

                }

            };

            function addStudent() {

                const rollNumber = document.getElementById('rollNumber').value;

                const studentName = document.getElementById('studentName').value;

                if (!rollNumber || !studentName) {

                    alert('Please fill in both Roll Number and Student Name');

                    return;

                }

                // Get existing data

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

                const newStudent = {

                    'Roll Number': rollNumber,

                    'Student Name': studentName,

                    'Class': 'BCA',

                    'Section': 'Management',

                    'Date': new Date().toISOString().split('T')[0]

                };

                // Add new student and sort by roll number

                studentData.push(newStudent);

                studentData.sort((a, b) => a['Roll Number'].localeCompare(b['Roll Number']));

                // Update table with sorted data

                const tbody = document.querySelector('#studentTable tbody');

                tbody.innerHTML = '';

                studentData.forEach(record => {

                    const row = document.createElement('tr');

                    row.innerHTML = `

                        <td>${record['Roll Number']}</td>

                        <td>${record['Student Name']}</td>

                        <td>${record['Class']}</td>

                        <td>${record['Section']}</td>

                        <td>${record['Date']}</td>

                        <td><button onclick="deleteStudent('${record['Roll Number']}')"

                            style="background-color: #dc3545; color: white; border: none;

                            border-radius: 4px; padding: 5px 10px; cursor: pointer;">Delete</button></td>

                    `;

                    tbody.appendChild(row);

                });

                // Update localStorage

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

                // Clear input fields and hide form

                document.getElementById('rollNumber').value = '';

                document.getElementById('studentName').value = '';

                document.getElementById('addStudentForm').style.display = 'none';

            }

            function deleteStudent(rollNumber) {

                if (confirm('Are you sure you want to delete this student?')) {

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

                    const updatedData = studentData.filter(student => student['Roll Number'] !== rollNumber);

                    localStorage.setItem('studentData', JSON.stringify(updatedData));

                    // Refresh table

                    const tbody = document.querySelector('#studentTable tbody');

                    tbody.innerHTML = '';

                    updatedData.forEach(record => {

                        const row = document.createElement('tr');

                        row.innerHTML = `

                            <td>${record['Roll Number']}</td>

                            <td>${record['Student Name']}</td>

                            <td>${record['Class']}</td>

                            <td>${record['Section']}</td>

                            <td>${record['Date']}</td>

                            <td><button onclick="deleteStudent('${record['Roll Number']}')"

                                style="background-color: #dc3545; color: white; border: none;

                                border-radius: 4px; padding: 5px 10px; cursor: pointer;">Delete</button></td>

                        `;

                        tbody.appendChild(row);

                    });

                }

            }

            // Tab navigation

            function showTab(tabId) {

                // Hide all tabs

                document.querySelectorAll('.tab-content').forEach(tab => {

                    tab.classList.remove('active');

                });

                // Show selected tab

                document.getElementById(tabId).classList.add('active');

                // If monthly report tab, generate report

                if (tabId === 'monthly-report') {

                    generateMonthlyReport();

                }

            }

            // Attendance functions

            function loadAttendanceSheet() {

                const date = document.getElementById('attendanceDate').value;

                const subject = document.getElementById('subjectSelect').value;

                if (!date) {

                    alert('Please select a date');

                    return;

                }

                if (!subject) {

                    alert('Please select a subject');

                    return;

                }

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

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

                // Check if attendance already exists for this date and subject

                const existingAttendance = attendanceData.filter(record =>

                    record.date === date && record.subject === subject

                );

                const tbody = document.querySelector('#attendanceTable tbody');

                tbody.innerHTML = '';

                if (existingAttendance.length > 0) {

                    // Display existing attendance

                    studentData.forEach(student => {

                        const attendanceRecord = existingAttendance.find(record =>

                            record.rollNumber === student['Roll Number']

                        );

                        const status = attendanceRecord ? attendanceRecord.status : 'Present';

                        const remarks = attendanceRecord ? attendanceRecord.remarks : '';

                        addAttendanceRow(student['Roll Number'], student['Student Name'], status, remarks);

                    });

                    alert('Loaded existing attendance for this date and subject');

                } else {

                    // Create new attendance sheet

                    studentData.forEach(student => {

                        addAttendanceRow(student['Roll Number'], student['Student Name'], 'Present', '');

                    });

                }

            }

            function addAttendanceRow(rollNumber, studentName, status, remarks) {

                const tbody = document.querySelector('#attendanceTable tbody');

                const row = document.createElement('tr');

                row.className = status === 'Present' ? 'status-present' : (status === 'Absent' ? 'status-absent' : 'status-late');

                row.innerHTML = `

                    <td>${rollNumber}</td>

                    <td>${studentName}</td>

                    <td>

                        <select onchange="updateRowStatus(this)" style="width: 100%; padding: 5px;">

                            <option value="Present" ${status === 'Present' ? 'selected' : ''}>Present</option>

                            <option value="Absent" ${status === 'Absent' ? 'selected' : ''}>Absent</option>

                            <option value="Late" ${status === 'Late' ? 'selected' : ''}>Late</option>

                        </select>

                    </td>

                    <td>

                        <input type="text" value="${remarks}" style="width: 100%; padding: 5px;">

                    </td>

                `;

                tbody.appendChild(row);

            }

            function updateRowStatus(selectElement) {

                const row = selectElement.closest('tr');

                const status = selectElement.value;

                // Remove all status classes

                row.classList.remove('status-present', 'status-absent', 'status-late');

                // Add appropriate class

                if (status === 'Present') {

                    row.classList.add('status-present');

                } else if (status === 'Absent') {

                    row.classList.add('status-absent');

                } else if (status === 'Late') {

                    row.classList.add('status-late');

                }

            }

            // Add this function to your teacher.html file's JavaScript section

            // Function to save attendance data in a format compatible with admin interface

            function saveAttendance() {

                const attendanceDate = document.getElementById('attendanceDate').value;

                const subject = document.getElementById('subjectSelect').value;

                if (!attendanceDate || !subject) {

                    alert('Please select a date and subject');

                    return;

                }

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

                const rows = attendanceTable.querySelectorAll('tbody tr');

                if (rows.length === 0) {

                    alert('No students loaded for attendance');

                    return;

                }

                // Get existing teacher attendance data

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

                // Remove previous records for this date and subject

                const filteredAttendance = teacherAttendance.filter(record =>

                    !(record.date === attendanceDate && record.subject === subject)

                );

                // Add new attendance records

                rows.forEach(row => {

                    const rollNumber = row.cells[0].textContent;

                    const studentName = row.cells[1].textContent;

                    const statusSelect = row.querySelector('select');

                    const remarksInput = row.querySelector('input[type="text"]');

                    if (statusSelect && remarksInput) {

                        const status = statusSelect.value;

                        const remarks = remarksInput.value;

                        filteredAttendance.push({

                            date: attendanceDate,

                            subject: subject,

                            rollNumber: rollNumber,

                            studentName: studentName,

                            status: status,

                            remarks: remarks

                        });

                    }

                });

                // Save updated attendance data

                localStorage.setItem('teacherAttendanceData', JSON.stringify(filteredAttendance));

                // Also update the shared attendance data for immediate admin access

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

                // Remove previous records for this date and subject

                const filteredAdminAttendance = adminAttendance.filter(record =>

                    !(record.date === attendanceDate && record.subject === subject)

                );

                // Add new attendance records to admin data

                rows.forEach(row => {

                    const rollNumber = row.cells[0].textContent;

                    const studentName = row.cells[1].textContent;

                    const statusSelect = row.querySelector('select');

                    const remarksInput = row.querySelector('input[type="text"]');

                    if (statusSelect && remarksInput) {

                        const status = statusSelect.value;

                        const remarks = remarksInput.value;

                        filteredAdminAttendance.push({

                            date: attendanceDate,

                            subject: subject,

                            rollNumber: rollNumber,

                            studentName: studentName,

                            status: status,

                            remarks: remarks

                        });

                    }

                });

                // Save updated admin attendance data

                localStorage.setItem('attendanceData', JSON.stringify(filteredAdminAttendance));

                alert('Attendance saved successfully');

            }

            // Function to load attendance sheet with data from admin if available

            function loadAttendanceSheet() {

                const attendanceDate = document.getElementById('attendanceDate').value;

                const subject = document.getElementById('subjectSelect').value;

                if (!attendanceDate || !subject) {

                    alert('Please select a date and subject');

                    return;

                }

                // Get student data

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

                if (students.length === 0) {

                    alert('No students found');

                    return;

                }

                // Get existing attendance data (check both teacher and admin data)

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

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

                // Combine attendance data, prioritizing teacher data

                const combinedAttendance = [...adminAttendance];

                teacherAttendance.forEach(teacherRecord => {

                    // Remove any duplicate admin records

                    const index = combinedAttendance.findIndex(adminRecord =>

                        adminRecord.date === teacherRecord.date &&

                        adminRecord.subject === teacherRecord.subject &&

                        adminRecord.rollNumber === teacherRecord.rollNumber

                    );

                    if (index !== -1) {

                        combinedAttendance[index] = teacherRecord;

                    } else {

                        combinedAttendance.push(teacherRecord);

                    }

                });

                // Populate attendance table

                const tbody = document.getElementById('attendanceTable').querySelector('tbody');

                tbody.innerHTML = '';

                students.forEach(student => {

                    const row = document.createElement('tr');

                    // Find existing attendance record for this student

                    const existingRecord = combinedAttendance.find(record =>

                        record.date === attendanceDate &&

                        record.subject === subject &&

                        record.rollNumber === student['Roll Number']

                    );

                    const status = existingRecord ? existingRecord.status : '';

                    const remarks = existingRecord ? existingRecord.remarks : '';

                    row.innerHTML = `

                        <td>${student['Roll Number']}</td>

                        <td>${student['Student Name']}</td>

                        <td>

                            <select class="status-select" onchange="updateRowStatus(this)">

                                <option value="Present" ${status === 'Present' ? 'selected' : ''}>Present</option>

                                <option value="Absent" ${status === 'Absent' ? 'selected' : ''}>Absent</option>

                                <option value="Late" ${status === 'Late' ? 'selected' : ''}>Late</option>

                            </select>

                        </td>

                        <td>

                            <input type="text" value="${remarks}" placeholder="Add remarks">

                        </td>

                    `;

                    // Set row class based on status

                    if (status === 'Present') {

                        row.classList.add('status-present');

                    } else if (status === 'Absent') {

                        row.classList.add('status-absent');

                    } else if (status === 'Late') {

                        row.classList.add('status-late');

                    }

                    tbody.appendChild(row);

                });

            }

            // Function to navigate to admin interface

            function goToAdminInterface() {

                window.location.href = 'admin.html';

            }

            // Function to update row status class

            function updateRowStatus(select) {

                const row = select.closest('tr');

                // Remove existing status classes

                row.classList.remove('status-present', 'status-absent', 'status-late');

                // Add appropriate class based on selected status

                if (select.value === 'Present') {

                    row.classList.add('status-present');

                } else if (select.value === 'Absent') {

                    row.classList.add('status-absent');

                } else if (select.value === 'Late') {

                    row.classList.add('status-late');

                }

            }

            function exportDailyAttendance() {

                const date = document.getElementById('attendanceDate').value;

                const subject = document.getElementById('subjectSelect').value;

                if (!date || !subject) {

                    alert('Please select both date and subject');

                    return;

                }

                // Create a new workbook

                const wb = XLSX.utils.book\_new();

                // Get table data

                const table = document.getElementById('attendanceTable');

                const ws = XLSX.utils.table\_to\_sheet(table);

                // Add worksheet to workbook

                XLSX.utils.book\_append\_sheet(wb, ws, "Attendance");

                // Save the file

                XLSX.writeFile(wb, `Attendance\_${subject}\_${date}.xlsx`);

            }

            function generateMonthlyReport() {

                const month = document.getElementById('monthPicker').value;

                const subject = document.getElementById('monthlySubjectSelect').value;

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

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

                // Filter attendance records for the selected month

                let filteredAttendance = attendanceData.filter(record =>

                    record.date.startsWith(month)

                );

                // Further filter by subject if selected

                if (subject) {

                    filteredAttendance = filteredAttendance.filter(record =>

                        record.subject === subject

                    );

                }

                // Calculate attendance statistics for each student

                const studentStats = {};

                filteredAttendance.forEach(record => {

                    if (!studentStats[record.rollNumber]) {

                        studentStats[record.rollNumber] = {

                            studentName: record.studentName,

                            totalClasses: 0,

                            present: 0,

                            absent: 0

                        };

                    }

                    studentStats[record.rollNumber].totalClasses++;

                    if (record.status === 'Present') {

                        studentStats[record.rollNumber].present++;

                    } else if (record.status === 'Absent') {

                        studentStats[record.rollNumber].absent++;

                    } else if (record.status === 'Late') {

                        // Count late as present for percentage calculation

                        studentStats[record.rollNumber].present++;

                    }

                });

                // Display report

                const tbody = document.querySelector('#monthlyReportTable tbody');

                tbody.innerHTML = '';

                // Add all students, even those without attendance records

                studentData.forEach(student => {

                    const rollNumber = student['Roll Number'];

                    const stats = studentStats[rollNumber] || {

                        studentName: student['Student Name'],

                        totalClasses: 0,

                        present: 0,

                        absent: 0

                    };

                    const attendancePercentage = stats.totalClasses > 0

                        ? ((stats.present / stats.totalClasses) \* 100).toFixed(1)

                        : '0.0';

                    const row = document.createElement('tr');

                    row.innerHTML = `

                        <td>${rollNumber}</td>

                        <td>${stats.studentName}</td>

                        <td>${stats.totalClasses}</td>

                        <td>${stats.present}</td>

                        <td>${stats.absent}</td>

                        <td>${attendancePercentage}%</td>

                    `;

                    tbody.appendChild(row);

                });

            }

            function exportMonthlyReport() {

                const month = document.getElementById('monthPicker').value;

                const subject = document.getElementById('monthlySubjectSelect').value;

                // Create a new workbook

                const wb = XLSX.utils.book\_new();

                // Get table data

                const table = document.getElementById('monthlyReportTable');

                const ws = XLSX.utils.table\_to\_sheet(table);

                // Add worksheet to workbook

                XLSX.utils.book\_append\_sheet(wb, ws, "Monthly Report");

                // Generate filename

                const subjectText = subject ? `\_${subject}` : '';

                const filename = `Monthly\_Attendance\_${month}${subjectText}.xlsx`;

                // Save the file

                XLSX.writeFile(wb, filename);

            }

    </script>

    </div>

    <script src="https://unpkg.com/xlsx/dist/xlsx.full.min.js"></script>

    <script>

        function exportToExcel() {

            // Create a new workbook

            const wb = XLSX.utils.book\_new();

            // Get table data

            const table = document.getElementById('studentTable');

            const ws = XLSX.utils.table\_to\_sheet(table);

            // Add worksheet to workbook

            XLSX.utils.book\_append\_sheet(wb, ws, "Students");

            // Save the file

            XLSX.writeFile(wb, "student\_records.xlsx");

        }

        function toggleAddForm() {

            const form = document.getElementById('addStudentForm');

            form.style.display = form.style.display === 'none' ? 'block' : 'none';

        }

    </script>

</body>

</html>