<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Attendease - Smart Classroom Attendance System</title>

<script src="https://cdnjs.cloudflare.com/ajax/libs/qrcode/1.5.3/qrcode.min.js"></script>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

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

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

min-height: 100vh;

color: #333;

}

.navbar {

background: rgba(255, 255, 255, 0.95);

backdrop-filter: blur(10px);

padding: 1rem 0;

position: sticky;

top: 0;

z-index: 1000;

box-shadow: 0 2px 20px rgba(0,0,0,0.1);

}

.nav-container {

max-width: 1200px;

margin: 0 auto;

padding: 0 20px;

display: flex;

justify-content: space-between;

align-items: center;

}

.logo {

font-size: 2rem;

font-weight: bold;

color: #667eea;

display: flex;

align-items: center;

gap: 10px;

}

.nav-buttons {

display: flex;

gap: 1rem;

}

.btn {

padding: 10px 20px;

border: none;

border-radius: 25px;

font-weight: 600;

cursor: pointer;

transition: all 0.3s ease;

text-decoration: none;

display: inline-block;

}

.btn-primary {

background: linear-gradient(45deg, #667eea, #764ba2);

color: white;

}

.btn-secondary {

background: transparent;

color: #667eea;

border: 2px solid #667eea;

}

.btn:hover {

transform: translateY(-2px);

box-shadow: 0 5px 15px rgba(0,0,0,0.2);

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 2rem;

}

.hero {

text-align: center;

color: white;

padding: 4rem 0;

}

.hero h1 {

font-size: 3.5rem;

margin-bottom: 1rem;

text-shadow: 2px 2px 4px rgba(0,0,0,0.3);

}

.hero p {

font-size: 1.3rem;

margin-bottom: 2rem;

opacity: 0.9;

}

.features {

display: grid;

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

gap: 2rem;

margin: 3rem 0;

}

.feature-card {

background: rgba(255, 255, 255, 0.95);

padding: 2rem;

border-radius: 15px;

box-shadow: 0 10px 30px rgba(0,0,0,0.1);

text-align: center;

transition: transform 0.3s ease;

}

.feature-card:hover {

transform: translateY(-10px);

}

.feature-icon {

font-size: 3rem;

margin-bottom: 1rem;

color: #667eea;

}

.modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: rgba(0,0,0,0.8);

z-index: 2000;

}

.modal-content {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

background: white;

padding: 2rem;

border-radius: 15px;

max-width: 600px;

width: 90%;

max-height: 90%;

overflow-y: auto;

}

.modal-header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 1rem;

padding-bottom: 1rem;

border-bottom: 1px solid #eee;

}

.close {

font-size: 2rem;

cursor: pointer;

color: #999;

}

.dashboard {

display: none;

}

.dashboard.active {

display: block;

}

.dashboard-header {

background: white;

padding: 2rem;

border-radius: 15px;

margin-bottom: 2rem;

box-shadow: 0 5px 15px rgba(0,0,0,0.1);

}

.dashboard-grid {

display: grid;

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

gap: 2rem;

}

.dashboard-card {

background: white;

padding: 2rem;

border-radius: 15px;

box-shadow: 0 5px 15px rgba(0,0,0,0.1);

}

.form-group {

margin-bottom: 1.5rem;

}

.form-group label {

display: block;

margin-bottom: 0.5rem;

color: #333;

font-weight: 600;

}

.form-group input,

.form-group select {

width: 100%;

padding: 12px;

border: 2px solid #e9ecef;

border-radius: 8px;

font-size: 1rem;

transition: border-color 0.3s ease;

}

.form-group input:focus,

.form-group select:focus {

outline: none;

border-color: #667eea;

}

.qr-container {

text-align: center;

padding: 2rem;

background: #f8f9fa;

border-radius: 10px;

margin: 1rem 0;

}

.video-container {

position: relative;

width: 100%;

max-width: 400px;

margin: 0 auto;

}

#videoElement {

width: 100%;

border-radius: 10px;

box-shadow: 0 5px 15px rgba(0,0,0,0.2);

}

.capture-overlay {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

width: 200px;

height: 200px;

border: 3px solid #667eea;

border-radius: 50%;

pointer-events: none;

}

.location-info {

background: #e8f4ff;

padding: 1rem;

border-radius: 8px;

margin: 1rem 0;

border-left: 4px solid #667eea;

}

.attendance-list {

background: white;

border-radius: 10px;

overflow: hidden;

box-shadow: 0 5px 15px rgba(0,0,0,0.1);

}

.attendance-header {

background: #667eea;

color: white;

padding: 1rem;

font-weight: 600;

}

.attendance-item {

padding: 1rem;

border-bottom: 1px solid #eee;

display: flex;

justify-content: space-between;

align-items: center;

}

.status-present {

color: #28a745;

font-weight: 600;

}

.status-absent {

color: #dc3545;

font-weight: 600;

}

.alert {

padding: 1rem;

border-radius: 8px;

margin: 1rem 0;

}

.alert-success {

background: #d4edda;

color: #155724;

border: 1px solid #c3e6cb;

}

.alert-error {

background: #f8d7da;

color: #721c24;

border: 1px solid #f5c6cb;

}

.alert-warning {

background: #fff3cd;

color: #856404;

border: 1px solid #ffeaa7;

}

.security-features {

background: white;

padding: 2rem;

border-radius: 15px;

margin: 2rem 0;

}

.security-item {

display: flex;

align-items: center;

gap: 1rem;

margin: 1rem 0;

padding: 1rem;

background: #f8f9fa;

border-radius: 8px;

}

.security-icon {

font-size: 2rem;

color: #28a745;

}

@media (max-width: 768px) {

.hero h1 {

font-size: 2.5rem;

}

.nav-buttons {

flex-direction: column;

gap: 0.5rem;

}

.dashboard-grid {

grid-template-columns: 1fr;

}

}

</style>

</head>

<body>

<nav class="navbar">

<div class="nav-container">

<div class="logo">

📚 Attendease

</div>

<div class="nav-buttons">

<button class="btn btn-secondary" onclick="showLogin('teacher')">Teacher Login</button>

<button class="btn btn-primary" onclick="showLogin('student')">Student Login</button>

</div>

</div>

</nav>

<div id="homepage">

<section class="hero">

<div class="container">

<h1>Attendease</h1>

<p>Smart Classroom Attendance System with QR Code & Face Detection</p>

<p>Secure, Accurate, Anti-Proxy Technology</p>

</div>

</section>

<section class="container">

<div class="features">

<div class="feature-card">

<div class="feature-icon">🔒</div>

<h3>Anti-Proxy Security</h3>

<p>Advanced face detection and location verification prevent proxy attendance</p>

</div>

<div class="feature-card">

<div class="feature-icon">📱</div>

<h3>QR Code Generation</h3>

<p>Teachers can generate time-sensitive QR codes for each class session</p>

</div>

<div class="feature-card">

<div class="feature-icon">👤</div>

<h3>Face Recognition</h3>

<p>Students verify attendance through facial recognition technology</p>

</div>

<div class="feature-card">

<div class="feature-icon">📍</div>

<h3>Location Tracking</h3>

<p>GPS verification ensures students are physically present in classroom</p>

</div>

<div class="feature-card">

<div class="feature-icon">📊</div>

<h3>Real-time Reports</h3>

<p>Instant attendance tracking and comprehensive analytics</p>

</div>

<div class="feature-card">

<div class="feature-icon">⚡</div>

<h3>Quick & Easy</h3>

<p>Streamlined process takes seconds to mark attendance</p>

</div>

</div>

<div class="security-features">

<h2 style="text-align: center; margin-bottom: 2rem; color: #667eea;">🛡️ Anti-Proxy Security Features</h2>

<div class="security-item">

<div class="security-icon">✅</div>

<div>

<h4>Live Face Detection</h4>

<p>Real-time facial recognition prevents photo-based fraud</p>

</div>

</div>

<div class="security-item">

<div class="security-icon">✅</div>

<div>

<h4>GPS Location Verification</h4>

<p>Ensures students are physically present in the designated classroom</p>

</div>

</div>

<div class="security-item">

<div class="security-icon">✅</div>

<div>

<h4>Time-Limited QR Codes</h4>

<p>QR codes expire after set duration to prevent sharing</p>

</div>

</div>

<div class="security-item">

<div class="security-icon">✅</div>

<div>

<h4>Device Fingerprinting</h4>

<p>Tracks device characteristics to prevent multiple submissions</p>

</div>

</div>

</div>

</section>

</div>

<!-- Teacher Dashboard -->

<div id="teacherDashboard" class="dashboard container">

<div class="dashboard-header">

<h1>👨‍🏫 Teacher Dashboard</h1>

<p>Welcome back! Manage your classes and generate QR codes for attendance.</p>

<button class="btn btn-secondary" onclick="logout()">Logout</button>

</div>

<div class="dashboard-grid">

<div class="dashboard-card">

<h3>📱 Generate QR Code</h3>

<form id="qrForm">

<div class="form-group">

<label for="className">Class Name</label>

<input type="text" id="className" required>

</div>

<div class="form-group">

<label for="subject">Subject</label>

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

</div>

<div class="form-group">

<label for="duration">QR Code Duration (minutes)</label>

<select id="duration">

<option value="5">5 minutes</option>

<option value="10">10 minutes</option>

<option value="15">15 minutes</option>

<option value="30">30 minutes</option>

</select>

</div>

<div class="form-group">

<label for="classroom">Classroom Location</label>

<input type="text" id="classroom" required>

</div>

<button type="submit" class="btn btn-primary" style="width: 100%;">Generate QR Code</button>

</form>

<div id="qrResult" class="qr-container" style="display: none;">

<h4>QR Code for Attendance</h4>

<div id="qrcode"></div>

<p id="qrInfo"></p>

<div id="qrTimer" class="alert alert-warning"></div>

</div>

</div>

<div class="dashboard-card">

<h3>📊 Today's Attendance</h3>

<div id="attendanceList" class="attendance-list">

<div class="attendance-header">Current Session</div>

<div id="attendanceItems">

<div class="attendance-item">

<span>No active session</span>

</div>

</div>

</div>

</div>

<div class="dashboard-card">

<h3>📈 Class Statistics</h3>

<div id="classStats">

<div style="display: flex; justify-content: space-between; margin: 1rem 0; padding: 1rem; background: #f8f9fa; border-radius: 8px;">

<span>Total Students:</span>

<span id="totalStudents">0</span>

</div>

<div style="display: flex; justify-content: space-between; margin: 1rem 0; padding: 1rem; background: #f8f9fa; border-radius: 8px;">

<span>Present Today:</span>

<span id="presentToday" class="status-present">0</span>

</div>

<div style="display: flex; justify-content: space-between; margin: 1rem 0; padding: 1rem; background: #f8f9fa; border-radius: 8px;">

<span>Attendance Rate:</span>

<span id="attendanceRate">0%</span>

</div>

</div>

</div>

</div>

</div>

<!-- Student Dashboard -->

<div id="studentDashboard" class="dashboard container">

<div class="dashboard-header">

<h1>👨‍🎓 Student Dashboard</h1>

<p>Scan QR code and verify your face for attendance.</p>

<button class="btn btn-secondary" onclick="logout()">Logout</button>

</div>

<div class="dashboard-grid">

<div class="dashboard-card">

<h3>📱 Scan QR Code</h3>

<form id="studentAttendanceForm">

<div class="form-group">

<label for="qrInput">Enter QR Code or Session ID</label>

<input type="text" id="qrInput" placeholder="Scan QR or enter session ID" required>

</div>

<button type="button" class="btn btn-primary" onclick="startAttendanceProcess()" style="width: 100%;">Start Attendance Process</button>

</form>

</div>

<div class="dashboard-card">

<h3>📷 Face Verification</h3>

<div id="faceVerification" style="display: none;">

<div class="video-container">

<video id="videoElement" autoplay></video>

<div class="capture-overlay"></div>

</div>

<button type="button" class="btn btn-primary" onclick="captureAttendance()" style="width: 100%; margin-top: 1rem;">Capture Attendance</button>

<div id="locationInfo" class="location-info" style="display: none;"></div>

</div>

</div>

<div class="dashboard-card">

<h3>📅 My Attendance</h3>

<div id="myAttendance">

<div style="text-align: center; padding: 2rem; color: #666;">

<p>Your attendance history will appear here</p>

</div>

</div>

</div>

</div>

</div>

<!-- Login Modal -->

<div id="loginModal" class="modal">

<div class="modal-content">

<div class="modal-header">

<h2 id="loginTitle">Login</h2>

<span class="close" onclick="closeModal()">&times;</span>

</div>

<form id="loginForm">

<div class="form-group">

<label for="username">Username</label>

<input type="text" id="username" required>

</div>

<div class="form-group">

<label for="password">Password</label>

<input type="password" id="password" required>

</div>

<button type="submit" class="btn btn-primary" style="width: 100%;">Login</button>

</form>

<div style="margin-top: 1rem; text-align: center; color: #666; font-size: 0.9rem;">

<p><strong>Demo Credentials:</strong></p>

<p>Teacher: admin/admin</p>

<p>Student: student/student</p>

</div>

</div>

</div>

<script>

// Global variables

let currentUser = null;

let currentRole = null;

let attendanceSession = null;

let qrTimer = null;

let videoStream = null;

// Sample data

const students = [

{ id: 1, name: "John Doe", rollNo: "2023001" },

{ id: 2, name: "Jane Smith", rollNo: "2023002" },

{ id: 3, name: "Mike Johnson", rollNo: "2023003" },

{ id: 4, name: "Sarah Wilson", rollNo: "2023004" }

];

let attendanceData = [];

let myAttendanceData = [];

// Authentication

function showLogin(role) {

currentRole = role;

document.getElementById('loginTitle').textContent = `${role.charAt(0).toUpperCase() + role.slice(1)} Login`;

document.getElementById('loginModal').style.display = 'block';

}

function closeModal() {

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

}

document.getElementById('loginForm').addEventListener('submit', function(e) {

e.preventDefault();

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

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

// Simple demo authentication

if ((currentRole === 'teacher' && username === 'admin' && password === 'admin') ||

(currentRole === 'student' && username === 'student' && password === 'student')) {

currentUser = username;

closeModal();

showDashboard(currentRole);

document.getElementById('loginForm').reset();

} else {

alert('Invalid credentials. Use demo credentials shown below the form.');

}

});

function showDashboard(role) {

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

if (role === 'teacher') {

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

updateTeacherStats();

} else {

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

loadMyAttendance();

}

}

function logout() {

currentUser = null;

currentRole = null;

attendanceSession = null;

// Stop video stream if active

if (videoStream) {

videoStream.getTracks().forEach(track => track.stop());

videoStream = null;

}

// Clear timers

if (qrTimer) {

clearInterval(qrTimer);

qrTimer = null;

}

document.getElementById('teacherDashboard').classList.remove('active');

document.getElementById('studentDashboard').classList.remove('active');

document.getElementById('homepage').style.display = 'block';

}

// Teacher functionality

document.getElementById('qrForm').addEventListener('submit', function(e) {

e.preventDefault();

generateQRCode();

});

function generateQRCode() {

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

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

const duration = parseInt(document.getElementById('duration').value);

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

// Create session

attendanceSession = {

id: 'SESSION\_' + Date.now(),

className: className,

subject: subject,

duration: duration,

classroom: classroom,

startTime: new Date(),

endTime: new Date(Date.now() + duration \* 60000),

attendees: []

};

// Generate QR Code

const qrData = JSON.stringify({

sessionId: attendanceSession.id,

className: className,

subject: subject,

classroom: classroom,

timestamp: Date.now(),

validUntil: Date.now() + duration \* 60000

});

document.getElementById('qrcode').innerHTML = '';

QRCode.toCanvas(document.getElementById('qrcode'), qrData, { width: 200 }, function (error) {

if (error) console.error(error);

console.log('QR Code generated successfully!');

});

document.getElementById('qrInfo').innerHTML = `

<strong>Session:</strong> ${className} - ${subject}<br>

<strong>Location:</strong> ${classroom}<br>

<strong>Session ID:</strong> ${attendanceSession.id}

`;

document.getElementById('qrResult').style.display = 'block';

// Start countdown timer

startQRTimer(duration);

// Update attendance list

updateAttendanceList();

}

function startQRTimer(duration) {

let timeLeft = duration \* 60; // Convert to seconds

const timerElement = document.getElementById('qrTimer');

qrTimer = setInterval(function() {

const minutes = Math.floor(timeLeft / 60);

const seconds = timeLeft % 60;

timerElement.innerHTML = `⏱️ QR Code expires in: ${minutes}:${seconds.toString().padStart(2, '0')}`;

if (timeLeft <= 0) {

clearInterval(qrTimer);

timerElement.innerHTML = '❌ QR Code has expired';

timerElement.className = 'alert alert-error';

attendanceSession = null;

}

timeLeft--;

}, 1000);

}

function updateAttendanceList() {

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

if (!attendanceSession || attendanceSession.attendees.length === 0) {

attendanceItems.innerHTML = '<div class="attendance-item"><span>No attendance recorded yet</span></div>';

return;

}

attendanceItems.innerHTML = attendanceSession.attendees.map(attendee => `

<div class="attendance-item">

<div>

<strong>${attendee.name}</strong><br>

<small>${attendee.rollNo} • ${new Date(attendee.timestamp).toLocaleTimeString()}</small>

</div>

<div class="status-present">✓ Present</div>

</div>

`).join('');

}

function updateTeacherStats() {

document.getElementById('totalStudents').textContent = students.length;

document.getElementById('presentToday').textContent = attendanceSession ? attendanceSession.attendees.length : 0;

const rate = attendanceSession && students.length > 0 ?

Math.round((attendanceSession.attendees.length / students.length) \* 100) : 0;

document.getElementById('attendanceRate').textContent = rate + '%';

}

// Student functionality

function startAttendanceProcess() {

const qrInput = document.getElementById('qrInput').value.trim();

if (!qrInput) {

alert('Please enter QR code or session ID');

return;

}

// Validate session

if (!attendanceSession ||

(qrInput !== attendanceSession.id && !qrInput.includes(attendanceSession.id))) {

alert('Invalid or expired QR code/session ID');

return;

}

// Check if session is still valid

if (new Date() > attendanceSession.endTime) {

alert('This attendance session has expired');

return;

}

// Start face verification

document.getElementById('faceVerification').style.display = 'block';

startCamera();

getCurrentLocation();

}

function startCamera() {

navigator.mediaDevices.getUserMedia({

video: {

width: 400,

height: 300,

facingMode: 'user'

}

})

.then(function(stream) {

videoStream = stream;

document.getElementById('videoElement').srcObject = stream;

})

.catch(function(err) {

console.error('Error accessing camera:', err);

alert('Camera access required for face verification');

});

}

function getCurrentLocation() {

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(

function(position) {

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

locationInfo.innerHTML = `

<strong>📍 Location Verified</strong><br>

Latitude: ${position.coords.latitude.toFixed(6)}<br>

Longitude: ${position.coords.longitude.toFixed(6)}<br>

<em>Location matches classroom requirements</em>

`;

locationInfo.style.display = 'block';

},

function(error) {

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

locationInfo.innerHTML = `

<strong>⚠️ Location Required</strong><br>

Please enable location services for attendance verification

`;

locationInfo.style.display = 'block';

locationInfo.className = 'alert alert-warning';

}

);

}

}

function captureAttendance() {

if (!videoStream) {

alert('Camera not ready. Please wait and try again.');

return;

}

// Simulate face detection and anti-proxy checks

const video = document.getElementById('videoElement');

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

canvas.width = video.videoWidth;

canvas.height = video.videoHeight;

const ctx = canvas.getContext('2d');

ctx.drawImage(video, 0, 0);

// Simulate face recognition process

setTimeout(() => {

// Anti-proxy checks simulation

const checks = [

'Live face detection: ✅ Passed',

'Photo/video detection: ✅ No fraud detected',

'Location verification: ✅ Within classroom radius',

'Device fingerprint: ✅ Unique device confirmed',

'Time validation: ✅ Within session time'

];

let checkResults = checks.join('\n');

// Simulate successful attendance

const student = students[Math.floor(Math.random() \* students.length)];

const attendanceRecord = {

...student,

timestamp: new Date().toISOString(),

location: 'Classroom A-101',

sessionI