<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Registro de Asistencia - Sistema de Gestión</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;

background: linear-gradient(135deg, #8B0000 0%, #4A0000 100%);

min-height: 100vh;

display: flex;

}

.sidebar {

width: 250px;

background: #f5f5f5;

padding: 20px 0;

border-right: 1px solid #ddd;

}

.sidebar h2 {

background: #8B0000;

color: white;

padding: 20px;

margin: 0 0 20px 0;

font-size: 18px;

font-weight: 600;

}

.sidebar-subtitle {

color: #666;

font-size: 12px;

margin-top: 5px;

}

.menu-item {

display: flex;

align-items: center;

padding: 12px 20px;

color: #666;

text-decoration: none;

border-left: 3px solid transparent;

transition: all 0.3s ease;

}

.menu-item:hover {

background: #f0f0f0;

border-left-color: #8B0000;

color: #333;

}

.menu-item.active {

background: #ffeaea;

border-left-color: #8B0000;

color: #8B0000;

}

.menu-item i {

margin-right: 10px;

width: 16px;

}

.main-content {

flex: 1;

padding: 20px;

}

.header {

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

backdrop-filter: blur(10px);

border-radius: 15px;

padding: 20px;

margin-bottom: 30px;

display: flex;

justify-content: space-between;

align-items: center;

color: white;

}

.header h1 {

font-size: 28px;

font-weight: 600;

}

.user-info {

display: flex;

align-items: center;

gap: 15px;

}

.user-avatar {

width: 40px;

height: 40px;

border-radius: 50%;

background: #fff;

display: flex;

align-items: center;

justify-content: center;

font-weight: bold;

color: #8B0000;

}

.logout-btn {

background: rgba(255, 255, 255, 0.2);

border: none;

color: white;

padding: 8px 16px;

border-radius: 8px;

cursor: pointer;

transition: all 0.3s ease;

}

.logout-btn:hover {

background: rgba(255, 255, 255, 0.3);

}

.content-card {

background: white;

border-radius: 15px;

padding: 30px;

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

margin-bottom: 20px;

}

.search-section {

margin-bottom: 30px;

}

.search-title {

font-size: 20px;

font-weight: 600;

color: #333;

margin-bottom: 20px;

display: flex;

align-items: center;

gap: 10px;

}

.search-container {

display: flex;

gap: 15px;

align-items: center;

margin-bottom: 20px;

}

.search-input {

flex: 1;

padding: 12px 16px;

border: 2px solid #e0e0e0;

border-radius: 8px;

font-size: 16px;

transition: border-color 0.3s ease;

}

.search-input:focus {

outline: none;

border-color: #8B0000;

}

.search-status {

display: flex;

align-items: center;

gap: 10px;

min-width: 200px;

padding: 12px;

font-size: 14px;

color: #666;

}

.results-container {

display: none;

}

.results-container.show {

display: block;

animation: fadeIn 0.3s ease;

}

@keyframes fadeIn {

from { opacity: 0; transform: translateY(-10px); }

to { opacity: 1; transform: translateY(0); }

}

.results-grid {

display: grid;

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

gap: 15px;

margin-top: 20px;

}

.user-card {

background: white;

border: 2px solid #e0e0e0;

border-radius: 12px;

padding: 20px;

transition: all 0.3s ease;

box-shadow: 0 2px 8px rgba(0,0,0,0.1);

}

.user-card:hover {

border-color: #8B0000;

transform: translateY(-2px);

box-shadow: 0 4px 15px rgba(139, 0, 0, 0.15);

}

.user-header {

margin-bottom: 15px;

padding-bottom: 10px;

border-bottom: 1px solid #f0f0f0;

}

.user-name {

font-size: 18px;

font-weight: 600;

color: #333;

margin-bottom: 5px;

}

.user-details {

display: flex;

justify-content: space-between;

align-items: center;

font-size: 14px;

color: #666;

}

.user-dni {

background: #f8f9fa;

padding: 3px 8px;

border-radius: 4px;

font-family: monospace;

font-weight: 500;

}

.user-role {

background: #e3f2fd;

color: #1976d2;

padding: 3px 8px;

border-radius: 4px;

font-size: 12px;

font-weight: 500;

}

.classroom-badge {

background: #f3e5f5;

color: #7b1fa2;

padding: 3px 8px;

border-radius: 4px;

font-size: 12px;

font-weight: 500;

margin-top: 8px;

display: inline-block;

}

.attendance-buttons {

display: flex;

gap: 8px;

margin-top: 15px;

}

.attendance-btn {

flex: 1;

padding: 10px 15px;

border: none;

border-radius: 6px;

font-size: 14px;

font-weight: 600;

cursor: pointer;

transition: all 0.3s ease;

text-align: center;

}

.btn-current-time {

background: #28a745;

color: white;

}

.btn-current-time:hover {

background: #218838;

transform: translateY(-2px);

}

.btn-class-time {

background: #007bff;

color: white;

}

.btn-class-time:hover {

background: #0056b3;

transform: translateY(-2px);

}

.alert {

padding: 15px;

border-radius: 8px;

margin-bottom: 20px;

display: none;

}

.alert.show {

display: block;

animation: slideIn 0.3s ease;

}

@keyframes slideIn {

from { opacity: 0; transform: translateX(-20px); }

to { opacity: 1; transform: translateX(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;

}

.no-user-message {

text-align: center;

color: #666;

font-style: italic;

padding: 20px;

}

.classroom-info {

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

color: white;

padding: 15px;

border-radius: 8px;

margin-bottom: 15px;

}

.classroom-title {

font-size: 14px;

opacity: 0.9;

margin-bottom: 5px;

}

.classroom-name {

font-size: 18px;

font-weight: 600;

}

.schedule-info {

background: linear-gradient(135deg, #f093fb 0%, #f5576c 100%);

color: white;

padding: 15px;

border-radius: 8px;

margin-bottom: 15px;

}

.loading {

display: inline-block;

width: 20px;

height: 20px;

border: 3px solid #f3f3f3;

border-top: 3px solid #8B0000;

border-radius: 50%;

animation: spin 1s linear infinite;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

</style>

</head>

<body>

<div class="sidebar">

<h2>

Dashboard

<div class="sidebar-subtitle">Sistema de Gestión</div>

</h2>

<a href="#" class="menu-item">

<i>📊</i> Resumen de asistencias

</a>

<a href="#" class="menu-item">

<i>📅</i> Bimestre

</a>

<a href="#" class="menu-item">

<i>👤</i> Nuevo usuario

</a>

<a href="#" class="menu-item">

<i>🎓</i> Matrícula

</a>

<a href="#" class="menu-item">

<i>📝</i> Justificaciones

</a>

<a href="#" class="menu-item active">

<i>✋</i> Registrar asistencia

</a>

</div>

<div class="main-content">

<div class="header">

<h1>Registro de Asistencia sin Carnet</h1>

<div class="user-info">

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

<div>

<div>Juan Jhony Agapito Guevara</div>

<div style="font-size: 12px; opacity: 0.8;">Administrador</div>

</div>

<button class="logout-btn">🚪 Cerrar Sesión</button>

</div>

</div>

<div class="content-card">

<div class="alert" id="alertContainer"></div>

<div class="search-section">

<div class="search-title">

<span>🔍</span>

Buscar Usuario para Registrar Asistencia

</div>

<div class="search-container">

<input type="text"

id="searchInput"

class="search-input"

placeholder="Ingrese DNI, nombre o apellido del usuario (mínimo 2 caracteres)...">

<div class="search-status" id="searchStatus">

<span id="searchLoader" class="loading" style="display: none;"></span>

<span id="searchStatusText"></span>

</div>

</div>

</div>

<div class="results-container" id="resultsContainer">

<div class="results-grid" id="resultsGrid">

<!-- Los resultados se mostrarán aquí -->

</div>

</div>

<div class="no-user-message" id="noUserMessage">

Ingrese al menos 2 caracteres para buscar un usuario

</div>

</div>

</div>

<script>

// Simulación de datos - En producción estos vendrían de la base de datos

const mockUsers = [

{

id: 1,

dni: '12345678',

nombres: 'Juan Carlos',

apellidos: 'Pérez García',

codigo: 'EST001',

email: 'juan.perez@email.com',

telefono: '987654321',

rol\_id: 1,

rol\_nombre: 'Estudiante',

aula\_id: 1,

aula\_nombre: 'Aula A1',

estado: 'Activo'

},

{

id: 2,

dni: '87654321',

nombres: 'María Elena',

apellidos: 'González López',

codigo: 'EST002',

email: 'maria.gonzalez@email.com',

telefono: '123456789',

rol\_id: 1,

rol\_nombre: 'Estudiante',

aula\_id: 2,

aula\_nombre: 'Aula B1',

estado: 'Activo'

},

{

id: 3,

dni: '11223344',

nombres: 'Pedro Antonio',

apellidos: 'Rodríguez Silva',

codigo: 'DOC001',

email: 'pedro.rodriguez@email.com',

telefono: '555666777',

rol\_id: 2,

rol\_nombre: 'Profesor',

aula\_id: null,

aula\_nombre: null,

estado: 'Activo'

},

{

id: 4,

dni: '22334455',

nombres: 'Ana María',

apellidos: 'Pérez Sánchez',

codigo: 'EST003',

email: 'ana.perez@email.com',

telefono: '111222333',

rol\_id: 1,

rol\_nombre: 'Estudiante',

aula\_id: 1,

aula\_nombre: 'Aula A1',

estado: 'Activo'

},

{

id: 5,

dni: '33445566',

nombres: 'Carlos Eduardo',

apellidos: 'Pérez Martínez',

codigo: 'EST004',

email: 'carlos.perez@email.com',

telefono: '444555666',

rol\_id: 1,

rol\_nombre: 'Estudiante',

aula\_id: 2,

aula\_nombre: 'Aula B1',

estado: 'Activo'

},

{

id: 6,

dni: '44556677',

nombres: 'Luisa Fernanda',

apellidos: 'García Pérez',

codigo: 'EST005',

email: 'luisa.garcia@email.com',

telefono: '777888999',

rol\_id: 1,

rol\_nombre: 'Estudiante',

aula\_id: 1,

aula\_nombre: 'Aula A1',

estado: 'Activo'

},

{

id: 7,

dni: '55667788',

nombres: 'Miguel Ángel',

apellidos: 'González García',

codigo: 'EST006',

email: 'miguel.gonzalez@email.com',

telefono: '666777888',

rol\_id: 1,

rol\_nombre: 'Estudiante',

aula\_id: 3,

aula\_nombre: 'Aula C1',

estado: 'Activo'

},

{

id: 8,

dni: '66778899',

nombres: 'Patricia Isabel',

apellidos: 'García Rodríguez',

codigo: 'EST007',

email: 'patricia.garcia@email.com',

telefono: '999000111',

rol\_id: 1,

rol\_nombre: 'Estudiante',

aula\_id: 3,

aula\_nombre: 'Aula C1',

estado: 'Activo'

}

];

const mockClassrooms = [

{

id: 1,

nombre: 'Aula A1',

hora\_inicio: '08:00:00',

hora\_fin: '17:00:00'

},

{

id: 2,

nombre: 'Aula B1',

hora\_inicio: '09:00:00',

hora\_fin: '18:00:00'

},

{

id: 3,

nombre: 'Aula C1',

hora\_inicio: '07:30:00',

hora\_fin: '16:30:00'

}

];

let searchTimeout = null;

const currentUserRole = 3; // Simular usuario con permisos (rol 3, 4, 5)

function hasPermission() {

return [3, 4, 5].includes(currentUserRole);

}

function showAlert(message, type = 'success') {

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

alertContainer.className = `alert alert-${type} show`;

alertContainer.textContent = message;

setTimeout(() => {

alertContainer.classList.remove('show');

}, 5000);

}

function updateSearchStatus(message, showLoader = false) {

const statusText = document.getElementById('searchStatusText');

const loader = document.getElementById('searchLoader');

statusText.textContent = message;

loader.style.display = showLoader ? 'inline-block' : 'none';

}

function searchUser(searchTerm) {

if (!hasPermission()) {

showAlert('No tienes permisos para registrar asistencias', 'error');

return;

}

if (!searchTerm || searchTerm.length < 2) {

hideResults();

updateSearchStatus('Ingrese al menos 2 caracteres para buscar');

return;

}

updateSearchStatus('Buscando...', true);

// Simular búsqueda con delay

setTimeout(() => {

const users = mockUsers.filter(u =>

u.dni.includes(searchTerm) ||

u.nombres.toLowerCase().includes(searchTerm.toLowerCase()) ||

u.apellidos.toLowerCase().includes(searchTerm.toLowerCase())

);

if (users.length > 0) {

displayUsers(users);

updateSearchStatus(`${users.length} usuario(s) encontrado(s)`);

} else {

hideResults();

updateSearchStatus('No se encontraron usuarios');

if (searchTerm.length >= 3) {

showAlert('No se encontró ningún usuario con ese criterio de búsqueda', 'warning');

}

}

}, 800);

}

function displayUsers(users) {

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

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

resultsGrid.innerHTML = '';

users.forEach(user => {

const userCard = createUserCard(user);

resultsGrid.appendChild(userCard);

});

resultsContainer.classList.add('show');

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

}

function createUserCard(user) {

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

card.className = 'user-card';

const classroom = mockClassrooms.find(c => c.id === user.aula\_id);

card.innerHTML = `

<div class="user-header">

<div class="user-name">${user.apellidos}, ${user.nombres}</div>

<div class="user-details">

<span class="user-dni">DNI: ${user.dni}</span>

<span class="user-role">${user.rol\_nombre}</span>

</div>

${user.aula\_nombre ? `<div class="classroom-badge">📚 ${user.aula\_nombre}</div>` : ''}

</div>

<div class="attendance-buttons">

<button class="attendance-btn btn-current-time" onclick="registerCurrentTime(${user.id})">

🕐 Hora Actual

</button>

<button class="attendance-btn btn-class-time" onclick="registerClassTime(${user.id})" ${user.aula\_id ? '' : 'disabled'}>

📚 Hora de Clase

</button>

</div>

`;

return card;

}

function hideResults() {

document.getElementById('resultsContainer').classList.remove('show');

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

}

function handleSearchInput() {

const searchTerm = document.getElementById('searchInput').value.trim();

// Limpiar timeout anterior

if (searchTimeout) {

clearTimeout(searchTimeout);

}

// Establecer nuevo timeout para evitar búsquedas excesivas

searchTimeout = setTimeout(() => {

searchUser(searchTerm);

}, 300);

}

function registerCurrentTime(userId) {

const user = mockUsers.find(u => u.id === userId);

if (!user) {

showAlert('Error: Usuario no encontrado', 'error');

return;

}

const currentTime = new Date();

const timeString = currentTime.toLocaleTimeString('es-PE', {

hour12: false,

hour: '2-digit',

minute: '2-digit',

second: '2-digit'

});

// Simular registro en base de datos

console.log('Registrando asistencia:', {

usuario\_id: user.id,

fecha: currentTime.toISOString().split('T')[0],

hora\_registrada: timeString,

tipo\_registro: 'hora\_actual',

registrado\_por: 'Usuario Admin'

});

showAlert(`✅ Asistencia registrada a las ${timeString} para ${user.nombres} ${user.apellidos}`, 'success');

}

function registerClassTime(userId) {

const user = mockUsers.find(u => u.id === userId);

if (!user) {

showAlert('Error: Usuario no encontrado', 'error');

return;

}

if (!user.aula\_id) {

showAlert('Este usuario no tiene un aula asignada', 'warning');

return;

}

const classroom = mockClassrooms.find(c => c.id === user.aula\_id);

if (!classroom) {

showAlert('No se encontró configuración de horario para el aula', 'error');

return;

}

const classStartTime = classroom.hora\_inicio;

const currentDate = new Date();

// Simular registro en base de datos

console.log('Registrando asistencia con hora de clase:', {

usuario\_id: user.id,

fecha: currentDate.toISOString().split('T')[0],

hora\_registrada: classStartTime,

tipo\_registro: 'hora\_clase',

aula\_id: user.aula\_id,

registrado\_por: 'Usuario Admin'

});

showAlert(`✅ Asistencia registrada a las ${classStartTime} (hora de clase) para ${user.nombres} ${user.apellidos}`, 'success');

}

// Configurar búsqueda automática

document.getElementById('searchInput').addEventListener('input', handleSearchInput);

// Verificar permisos al cargar

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

if (!hasPermission()) {

showAlert('Su rol no tiene permisos para registrar asistencias', 'error');

document.getElementById('searchInput').disabled = true;

updateSearchStatus('Sin permisos para registrar asistencias');

} else {

updateSearchStatus('Listo para buscar usuarios');

}

});

</script>

</body>

</html>