<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Academia - Control de Alumnos (Web)</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<style>

:root {

--primary: #1976d2;

--primary-dark: #1565c0;

--secondary: #f4f7fb;

--card-bg: #ffffff;

--text: #333333;

--text-light: #666666;

--border: #e0e0e0;

--success: #4caf50;

--warning: #ff9800;

--danger: #f44336;

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

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

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

}

body {

background-color: var(--secondary);

color: var(--text);

line-height: 1.6;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

header {

background-color: var(--primary);

color: white;

padding: 15px 0;

box-shadow: var(--shadow);

position: sticky;

top: 0;

z-index: 100;

}

.header-content {

max-width: 1200px;

margin: 0 auto;

padding: 0 20px;

display: flex;

justify-content: space-between;

align-items: center;

}

.logo {

font-size: 24px;

font-weight: bold;

}

.nav-tabs {

display: flex;

background-color: white;

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

padding: 0 20px;

}

.tab {

padding: 15px 25px;

cursor: pointer;

border-bottom: 3px solid transparent;

transition: all 0.3s;

font-weight: 500;

}

.tab.active {

border-bottom-color: var(--primary);

color: var(--primary);

}

.tab:hover {

background-color: rgba(25, 118, 210, 0.05);

}

.tab-content {

display: none;

padding: 20px;

background-color: white;

border-radius: 0 0 8px 8px;

box-shadow: var(--shadow);

margin-bottom: 20px;

}

.tab-content.active {

display: block;

}

.card {

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

border-radius: 8px;

box-shadow: var(--shadow);

padding: 20px;

margin-bottom: 20px;

}

.card-title {

font-size: 18px;

font-weight: 600;

margin-bottom: 10px;

color: var(--primary);

}

.card-value {

font-size: 32px;

font-weight: bold;

margin-bottom: 5px;

}

.card-subtitle {

color: var(--text-light);

font-size: 14px;

}

.stats-container {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 20px;

margin-bottom: 20px;

}

.btn {

padding: 10px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

font-weight: 500;

transition: background-color 0.3s;

display: inline-flex;

align-items: center;

gap: 8px;

}

.btn-primary {

background-color: var(--primary);

color: white;

}

.btn-primary:hover {

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

}

.btn-secondary {

background-color: #e0e0e0;

color: var(--text);

}

.btn-secondary:hover {

background-color: #d0d0d0;

}

.btn-success {

background-color: var(--success);

color: white;

}

.btn-success:hover {

background-color: #3d8b40;

}

.btn-danger {

background-color: var(--danger);

color: white;

}

.btn-danger:hover {

background-color: #d32f2f;

}

.btn-group {

display: flex;

gap: 10px;

margin: 15px 0;

}

.form-group {

margin-bottom: 15px;

}

.form-label {

display: block;

margin-bottom: 5px;

font-weight: 500;

}

.form-control {

width: 100%;

padding: 10px;

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

border-radius: 4px;

font-size: 16px;

}

.form-control:focus {

outline: none;

border-color: var(--primary);

}

.form-row {

display: flex;

gap: 15px;

}

.form-row .form-group {

flex: 1;

}

.search-container {

display: flex;

margin-bottom: 20px;

}

.search-input {

flex: 1;

padding: 10px;

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

border-radius: 4px 0 0 4px;

font-size: 16px;

}

.search-btn {

background-color: var(--primary);

color: white;

border: none;

border-radius: 0 4px 4px 0;

padding: 0 15px;

cursor: pointer;

}

.list-container {

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

border-radius: 4px;

max-height: 400px;

overflow-y: auto;

margin-bottom: 20px;

}

.list-item {

padding: 12px 15px;

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

cursor: pointer;

transition: background-color 0.2s;

}

.list-item:hover {

background-color: #f5f5f5;

}

.list-item.selected {

background-color: #e3f2fd;

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

}

.list-item:last-child {

border-bottom: none;

}

.photo-container {

display: flex;

align-items: flex-start;

gap: 20px;

margin: 20px 0;

}

.photo-preview {

width: 160px;

height: 160px;

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

border-radius: 4px;

display: flex;

align-items: center;

justify-content: center;

overflow: hidden;

background-color: #f9f9f9;

}

.photo-preview img {

max-width: 100%;

max-height: 100%;

}

.photo-actions {

display: flex;

flex-direction: column;

gap: 10px;

}

.table-container {

overflow-x: auto;

}

table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

th, td {

padding: 12px 15px;

text-align: left;

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

}

th {

background-color: #f5f5f5;

font-weight: 600;

}

tr:hover {

background-color: #f9f9f9;

}

.modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

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

z-index: 1000;

align-items: center;

justify-content: center;

}

.modal-content {

background-color: white;

border-radius: 8px;

box-shadow: var(--shadow);

width: 90%;

max-width: 500px;

max-height: 90vh;

overflow-y: auto;

}

.modal-header {

padding: 15px 20px;

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

display: flex;

justify-content: space-between;

align-items: center;

}

.modal-title {

font-size: 18px;

font-weight: 600;

}

.modal-close {

background: none;

border: none;

font-size: 24px;

cursor: pointer;

color: var(--text-light);

}

.modal-body {

padding: 20px;

}

.modal-footer {

padding: 15px 20px;

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

display: flex;

justify-content: flex-end;

gap: 10px;

}

.toast {

position: fixed;

bottom: 20px;

right: 20px;

padding: 15px 20px;

background-color: var(--success);

color: white;

border-radius: 4px;

box-shadow: var(--shadow);

display: flex;

align-items: center;

gap: 10px;

z-index: 1100;

transform: translateY(100px);

opacity: 0;

transition: all 0.3s;

}

.toast.show {

transform: translateY(0);

opacity: 1;

}

.toast.error {

background-color: var(--danger);

}

.toast.warning {

background-color: var(--warning);

}

@media (max-width: 768px) {

.form-row {

flex-direction: column;

gap: 0;

}

.stats-container {

grid-template-columns: 1fr;

}

.nav-tabs {

overflow-x: auto;

white-space: nowrap;

}

.tab {

padding: 15px 20px;

}

}

</style>

</head>

<body>

<header>

<div class="header-content">

<div class="logo">Academia - Control de Alumnos</div>

<div class="user-info">

<i class="fas fa-user"></i> Administrador

</div>

</div>

</header>

<div class="nav-tabs">

<div class="tab active" data-tab="dashboard">Dashboard</div>

<div class="tab" data-tab="students">Alumnos</div>

<div class="tab" data-tab="payments">Pagos</div>

<div class="tab" data-tab="grades">Calificaciones</div>

</div>

<div class="container">

<!-- Dashboard Tab -->

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

<div class="stats-container">

<div class="card">

<div class="card-title">Total de Alumnos</div>

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

<div class="card-subtitle">Registrados en el sistema</div>

</div>

<div class="card">

<div class="card-title">Pagos del Mes</div>

<div class="card-value" id="month-payments">Q0.00</div>

<div class="card-subtitle">Ingresos del mes actual</div>

</div>

<div class="card">

<div class="card-title">Promedio Calificaciones</div>

<div class="card-value" id="avg-grades">0.0</div>

<div class="card-subtitle">Promedio general</div>

</div>

</div>

<div class="btn-group">

<button class="btn btn-primary" id="btn-new-student">

<i class="fas fa-user-plus"></i> Nuevo Alumno

</button>

<button class="btn btn-secondary" id="btn-import-csv">

<i class="fas fa-file-import"></i> Importar CSV

</button>

<button class="btn btn-secondary" id="btn-refresh-dashboard">

<i class="fas fa-sync-alt"></i> Actualizar

</button>

</div>

</div>

<!-- Students Tab -->

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

<div class="card">

<div class="card-title">Gestión de Alumnos</div>

<div class="form-row">

<div class="form-group" style="flex: 2;">

<div class="search-container">

<input type="text" class="search-input" id="student-search" placeholder="Buscar por nombre, apellido o ID">

<button class="search-btn" id="student-search-btn">

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

</button>

</div>

</div>

<div class="form-group">

<div class="btn-group">

<button class="btn btn-primary" id="btn-new-student2">

<i class="fas fa-user-plus"></i> Nuevo

</button>

<button class="btn btn-secondary" id="btn-export-csv">

<i class="fas fa-file-export"></i> Exportar

</button>

</div>

</div>

</div>

<div class="form-row">

<div class="form-group" style="flex: 1;">

<div class="list-container" id="students-list">

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

</div>

</div>

<div class="form-group" style="flex: 2;">

<div class="card">

<div class="card-title">Detalles del Alumno</div>

<div class="form-row">

<div class="form-group">

<label class="form-label">Nombre</label>

<input type="text" class="form-control" id="student-first-name">

</div>

<div class="form-group">

<label class="form-label">Apellido</label>

<input type="text" class="form-control" id="student-last-name">

</div>

</div>

<div class="form-row">

<div class="form-group">

<label class="form-label">Fecha de Nacimiento</label>

<input type="date" class="form-control" id="student-dob">

</div>

<div class="form-group">

<label class="form-label">Teléfono</label>

<input type="tel" class="form-control" id="student-phone">

</div>

</div>

<div class="form-group">

<label class="form-label">Email</label>

<input type="email" class="form-control" id="student-email">

</div>

<div class="form-group">

<label class="form-label">Notas</label>

<textarea class="form-control" id="student-notes" rows="3"></textarea>

</div>

<div class="photo-container">

<div class="photo-preview" id="student-photo-preview">

<span>Sin foto</span>

</div>

<div class="photo-actions">

<button class="btn btn-secondary" id="btn-load-photo">

<i class="fas fa-camera"></i> Cargar Foto

</button>

<button class="btn btn-secondary" id="btn-remove-photo">

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

</button>

<input type="file" id="student-photo-input" accept="image/\*" style="display:none;">

</div>

</div>

<div class="btn-group">

<button class="btn btn-primary" id="btn-save-student">

<i class="fas fa-save"></i> Guardar

</button>

<button class="btn btn-secondary" id="btn-view-payments">

<i class="fas fa-money-bill-wave"></i> Ver Pagos

</button>

<button class="btn btn-secondary" id="btn-view-grades">

<i class="fas fa-graduation-cap"></i> Ver Calificaciones

</button>

<button class="btn btn-danger" id="btn-delete-student">

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

</button>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Payments Tab -->

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

<div class="card">

<div class="card-title">Gestión de Pagos</div>

<div class="form-row">

<div class="form-group" style="flex: 2;">

<label class="form-label">Alumno</label>

<div class="search-container">

<input type="text" class="search-input" id="payment-student-search" placeholder="Buscar por nombre, apellido o ID">

<button class="search-btn" id="payment-student-search-btn">

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

</button>

</div>

</div>

<div class="form-group">

<label class="form-label">&nbsp;</label>

<div class="btn-group">

<button class="btn btn-primary" id="btn-new-payment">

<i class="fas fa-plus"></i> Nuevo Pago

</button>

<button class="btn btn-secondary" id="btn-export-payments">

<i class="fas fa-file-export"></i> Exportar

</button>

</div>

</div>

</div>

<div class="table-container">

<table id="payments-table">

<thead>

<tr>

<th>ID</th>

<th>Fecha</th>

<th>Monto</th>

<th>Método</th>

<th>Notas</th>

<th>Acciones</th>

</tr>

</thead>

<tbody>

<!-- Payments will be listed here -->

</tbody>

</table>

</div>

</div>

</div>

<!-- Grades Tab -->

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

<div class="card">

<div class="card-title">Gestión de Calificaciones</div>

<div class="form-row">

<div class="form-group" style="flex: 2;">

<label class="form-label">Alumno</label>

<div class="search-container">

<input type="text" class="search-input" id="grade-student-search" placeholder="Buscar por nombre, apellido o ID">

<button class="search-btn" id="grade-student-search-btn">

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

</button>

</div>

</div>

<div class="form-group">

<label class="form-label">&nbsp;</label>

<div class="btn-group">

<button class="btn btn-primary" id="btn-new-grade">

<i class="fas fa-plus"></i> Nueva Calificación

</button>

<button class="btn btn-secondary" id="btn-export-grades">

<i class="fas fa-file-export"></i> Exportar

</button>

</div>

</div>

</div>

<div class="table-container">

<table id="grades-table">

<thead>

<tr>

<th>ID</th>

<th>Asignatura</th>

<th>Calificación</th>

<th>Fecha</th>

<th>Notas</th>

<th>Acciones</th>

</tr>

</thead>

<tbody>

<!-- Grades will be listed here -->

</tbody>

</table>

</div>

</div>

</div>

</div>

<!-- Payment Modal -->

<div class="modal" id="payment-modal">

<div class="modal-content">

<div class="modal-header">

<div class="modal-title" id="payment-modal-title">Nuevo Pago</div>

<button class="modal-close">&times;</button>

</div>

<div class="modal-body">

<div class="form-group">

<label class="form-label">Fecha</label>

<input type="date" class="form-control" id="payment-date">

</div>

<div class="form-group">

<label class="form-label">Monto</label>

<input type="number" class="form-control" id="payment-amount" step="0.01">

</div>

<div class="form-group">

<label class="form-label">Método de Pago</label>

<select class="form-control" id="payment-method">

<option value="">Seleccionar método</option>

<option value="Efectivo">Efectivo</option>

<option value="Tarjeta">Tarjeta</option>

<option value="Transferencia">Transferencia</option>

<option value="Cheque">Cheque</option>

</select>

</div>

<div class="form-group">

<label class="form-label">Notas</label>

<textarea class="form-control" id="payment-notes" rows="3"></textarea>

</div>

</div>

<div class="modal-footer">

<button class="btn btn-secondary" id="payment-modal-cancel">Cancelar</button>

<button class="btn btn-primary" id="payment-modal-save">Guardar</button>

<button class="btn btn-danger" id="payment-modal-delete" style="display: none;">Eliminar</button>

</div>

</div>

</div>

<!-- Grade Modal -->

<div class="modal" id="grade-modal">

<div class="modal-content">

<div class="modal-header">

<div class="modal-title" id="grade-modal-title">Nueva Calificación</div>

<button class="modal-close">&times;</button>

</div>

<div class="modal-body">

<div class="form-group">

<label class="form-label">Asignatura</label>

<input type="text" class="form-control" id="grade-subject">

</div>

<div class="form-group">

<label class="form-label">Calificación (0-100)</label>

<input type="number" class="form-control" id="grade-value" step="0.1" min="0" max="100">

</div>

<div class="form-group">

<label class="form-label">Fecha</label>

<input type="date" class="form-control" id="grade-date">

</div>

<div class="form-group">

<label class="form-label">Notas</label>

<textarea class="form-control" id="grade-notes" rows="3"></textarea>

</div>

</div>

<div class="modal-footer">

<button class="btn btn-secondary" id="grade-modal-cancel">Cancelar</button>

<button class="btn btn-primary" id="grade-modal-save">Guardar</button>

<button class="btn btn-danger" id="grade-modal-delete" style="display: none;">Eliminar</button>

</div>

</div>

</div>

<!-- Toast Notification -->

<div class="toast" id="toast">

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

<span id="toast-message">Operación completada</span>

</div>

<script>

// Data storage simulation (in a real app, this would be server-side)

let students = JSON.parse(localStorage.getItem('academy\_students')) || [];

let payments = JSON.parse(localStorage.getItem('academy\_payments')) || [];

let grades = JSON.parse(localStorage.getItem('academy\_grades')) || [];

// Current selections

let selectedStudentId = null;

let selectedPaymentId = null;

let selectedGradeId = null;

// Initialize the application

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

initializeTabs();

initializeDashboard();

initializeStudentsTab();

initializePaymentsTab();

initializeGradesTab();

initializeModals();

// Load initial data

refreshDashboard();

loadStudentsList();

});

// Tab navigation

function initializeTabs() {

const tabs = document.querySelectorAll('.tab');

tabs.forEach(tab => {

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

// Remove active class from all tabs and contents

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

document.querySelectorAll('.tab-content').forEach(c => c.classList.remove('active'));

// Add active class to clicked tab and corresponding content

this.classList.add('active');

document.getElementById(this.dataset.tab).classList.add('active');

});

});

// Button to switch to students tab

document.getElementById('btn-new-student').addEventListener('click', function() {

document.querySelectorAll('.tab').forEach(t => t.classList.remove('active'));

document.querySelectorAll('.tab-content').forEach(c => c.classList.remove('active'));

document.querySelector('.tab[data-tab="students"]').classList.add('active');

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

// Reset student form

resetStudentForm();

});

document.getElementById('btn-new-student2').addEventListener('click', resetStudentForm);

}

// Dashboard functions

function initializeDashboard() {

document.getElementById('btn-refresh-dashboard').addEventListener('click', refreshDashboard);

}

function refreshDashboard() {

// Update student count

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

// Calculate monthly payments

const now = new Date();

const currentMonth = now.getMonth();

const currentYear = now.getFullYear();

const monthlyPayments = payments.filter(p => {

const paymentDate = new Date(p.date);

return paymentDate.getMonth() === currentMonth &&

paymentDate.getFullYear() === currentYear;

});

const monthlyTotal = monthlyPayments.reduce((sum, p) => sum + parseFloat(p.amount), 0);

document.getElementById('month-payments').textContent = `Q${monthlyTotal.toFixed(2)}`;

// Calculate average grade

if (grades.length > 0) {

const avgGrade = grades.reduce((sum, g) => sum + parseFloat(g.grade), 0) / grades.length;

document.getElementById('avg-grades').textContent = avgGrade.toFixed(2);

} else {

document.getElementById('avg-grades').textContent = '0.0';

}

}

// Students tab functions

function initializeStudentsTab() {

// Search functionality

document.getElementById('student-search-btn').addEventListener('click', loadStudentsList);

document.getElementById('student-search').addEventListener('keyup', function(e) {

if (e.key === 'Enter') loadStudentsList();

});

// Student form buttons

document.getElementById('btn-save-student').addEventListener('click', saveStudent);

document.getElementById('btn-delete-student').addEventListener('click', deleteStudent);

document.getElementById('btn-view-payments').addEventListener('click', viewStudentPayments);

document.getElementById('btn-view-grades').addEventListener('click', viewStudentGrades);

// Photo buttons

document.getElementById('btn-load-photo').addEventListener('click', function() {

document.getElementById('student-photo-input').click();

});

document.getElementById('student-photo-input').addEventListener('change', loadStudentPhoto);

document.getElementById('btn-remove-photo').addEventListener('click', removeStudentPhoto);

// Export button

document.getElementById('btn-export-csv').addEventListener('click', exportStudentsCSV);

}

function loadStudentsList() {

const searchTerm = document.getElementById('student-search').value.toLowerCase();

const listContainer = document.getElementById('students-list');

listContainer.innerHTML = '';

let filteredStudents = students;

if (searchTerm) {

if (!isNaN(searchTerm)) {

// Search by ID

filteredStudents = students.filter(s => s.id == searchTerm);

} else {

// Search by name

filteredStudents = students.filter(s =>

s.firstName.toLowerCase().includes(searchTerm) ||

s.lastName.toLowerCase().includes(searchTerm)

);

}

}

if (filteredStudents.length === 0) {

listContainer.innerHTML = '<div class="list-item">No se encontraron alumnos</div>';

return;

}

filteredStudents.forEach(student => {

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

listItem.className = 'list-item';

if (student.id === selectedStudentId) {

listItem.classList.add('selected');

}

listItem.textContent = `${student.id} - ${student.lastName}, ${student.firstName}`;

listItem.dataset.id = student.id;

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

// Remove selected class from all items

document.querySelectorAll('#students-list .list-item').forEach(item => {

item.classList.remove('selected');

});

// Add selected class to clicked item

this.classList.add('selected');

// Load student details

loadStudentDetails(parseInt(this.dataset.id));

});

listContainer.appendChild(listItem);

});

}

function loadStudentDetails(studentId) {

selectedStudentId = studentId;

const student = students.find(s => s.id === studentId);

if (!student) return;

document.getElementById('student-first-name').value = student.firstName;

document.getElementById('student-last-name').value = student.lastName;

document.getElementById('student-dob').value = student.dob || '';

document.getElementById('student-phone').value = student.phone || '';

document.getElementById('student-email').value = student.email || '';

document.getElementById('student-notes').value = student.notes || '';

// Display photo

const photoPreview = document.getElementById('student-photo-preview');

if (student.photo) {

photoPreview.innerHTML = `<img src="${student.photo}" alt="Foto del alumno">`;

} else {

photoPreview.innerHTML = '<span>Sin foto</span>';

}

}

function resetStudentForm() {

selectedStudentId = null;

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

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

document.getElementById('student-dob').value = '';

document.getElementById('student-phone').value = '';

document.getElementById('student-email').value = '';

document.getElementById('student-notes').value = '';

document.getElementById('student-photo-preview').innerHTML = '<span>Sin foto</span>';

// Clear selection in list

document.querySelectorAll('#students-list .list-item').forEach(item => {

item.classList.remove('selected');

});

}

function saveStudent() {

const firstName = document.getElementById('student-first-name').value.trim();

const lastName = document.getElementById('student-last-name').value.trim();

if (!firstName || !lastName) {

showToast('Nombre y apellido son obligatorios', 'error');

return;

}

const studentData = {

firstName,

lastName,

dob: document.getElementById('student-dob').value || null,

phone: document.getElementById('student-phone').value.trim() || null,

email: document.getElementById('student-email').value.trim() || null,

notes: document.getElementById('student-notes').value.trim() || null,

photo: null

};

if (selectedStudentId) {

// Update existing student

const index = students.findIndex(s => s.id === selectedStudentId);

if (index !== -1) {

studentData.id = selectedStudentId;

studentData.photo = students[index].photo; // Preserve existing photo

students[index] = studentData;

}

} else {

// Create new student

studentData.id = students.length > 0 ? Math.max(...students.map(s => s.id)) + 1 : 1;

students.push(studentData);

selectedStudentId = studentData.id;

}

// Save to localStorage (simulating database)

localStorage.setItem('academy\_students', JSON.stringify(students));

showToast('Alumno guardado correctamente');

loadStudentsList();

refreshDashboard();

}

function deleteStudent() {

if (!selectedStudentId) {

showToast('Seleccione un alumno para eliminar', 'warning');

return;

}

if (!confirm('¿Está seguro de que desea eliminar este alumno y todos sus registros?')) {

return;

}

// Delete student's payments and grades

payments = payments.filter(p => p.studentId !== selectedStudentId);

grades = grades.filter(g => g.studentId !== selectedStudentId);

// Delete student

students = students.filter(s => s.id !== selectedStudentId);

// Save changes

localStorage.setItem('academy\_students', JSON.stringify(students));

localStorage.setItem('academy\_payments', JSON.stringify(payments));

localStorage.setItem('academy\_grades', JSON.stringify(grades));

showToast('Alumno eliminado correctamente');

resetStudentForm();

loadStudentsList();

refreshDashboard();

}

function viewStudentPayments() {

if (!selectedStudentId) {

showToast('Seleccione un alumno primero', 'warning');

return;

}

// Switch to payments tab

document.querySelectorAll('.tab').forEach(t => t.classList.remove('active'));

document.querySelectorAll('.tab-content').forEach(c => c.classList.remove('active'));

document.querySelector('.tab[data-tab="payments"]').classList.add('active');

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

// Load student payments

const student = students.find(s => s.id === selectedStudentId);

if (student) {

document.getElementById('payment-student-search').value = `${student.id} - ${student.lastName}, ${student.firstName}`;

loadPaymentsForStudent(selectedStudentId);

}

}

function viewStudentGrades() {

if (!selectedStudentId) {

showToast('Seleccione un alumno primero', 'warning');

return;

}

// Switch to grades tab

document.querySelectorAll('.tab').forEach(t => t.classList.remove('active'));

document.querySelectorAll('.tab-content').forEach(c => c.classList.remove('active'));

document.querySelector('.tab[data-tab="grades"]').classList.add('active');

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

// Load student grades

const student = students.find(s => s.id === selectedStudentId);

if (student) {

document.getElementById('grade-student-search').value = `${student.id} - ${student.lastName}, ${student.firstName}`;

loadGradesForStudent(selectedStudentId);

}

}

function loadStudentPhoto(e) {

const file = e.target.files[0];

if (!file) return;

// Check if file is an image

if (!file.type.match('image.\*')) {

showToast('Por favor seleccione un archivo de imagen válido', 'error');

return;

}

const reader = new FileReader();

reader.onload = function(event) {

const photoData = event.target.result;

document.getElementById('student-photo-preview').innerHTML = `<img src="${photoData}" alt="Foto del alumno">`;

// Save photo to student data

if (selectedStudentId) {

const student = students.find(s => s.id === selectedStudentId);

if (student) {

student.photo = photoData;

localStorage.setItem('academy\_students', JSON.stringify(students));

showToast('Foto cargada correctamente');

}

} else {

showToast('Guarde el alumno primero para asociar la foto', 'warning');

}

};

reader.readAsDataURL(file);

}

function removeStudentPhoto() {

if (!selectedStudentId) {

showToast('Seleccione un alumno primero', 'warning');

return;

}

const student = students.find(s => s.id === selectedStudentId);

if (student) {

student.photo = null;

localStorage.setItem('academy\_students', JSON.stringify(students));

document.getElementById('student-photo-preview').innerHTML = '<span>Sin foto</span>';

showToast('Foto eliminada');

}

}

function exportStudentsCSV() {

if (students.length === 0) {

showToast('No hay alumnos para exportar', 'warning');

return;

}

// Create CSV content

const headers = ['ID', 'Nombre', 'Apellido', 'Fecha Nacimiento', 'Teléfono', 'Email', 'Notas'];

let csvContent = headers.join(',') + '\n';

students.forEach(student => {

const row = [

student.id,

`"${student.firstName}"`,

`"${student.lastName}"`,

student.dob || '',

student.phone || '',

student.email || '',

`"${student.notes || ''}"`

];

csvContent += row.join(',') + '\n';

});

// Create download link

const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });

const url = URL.createObjectURL(blob);

const link = document.createElement('a');

link.setAttribute('href', url);

link.setAttribute('download', 'alumnos.csv');

link.style.visibility = 'hidden';

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

showToast('Archivo CSV exportado correctamente');

}

// Payments tab functions

function initializePaymentsTab() {

// Search functionality

document.getElementById('payment-student-search-btn').addEventListener('click', searchPaymentStudent);

document.getElementById('payment-student-search').addEventListener('keyup', function(e) {

if (e.key === 'Enter') searchPaymentStudent();

});

// New payment button

document.getElementById('btn-new-payment').addEventListener('click', newPayment);

// Export button

document.getElementById('btn-export-payments').addEventListener('click', exportPaymentsCSV);

}

function searchPaymentStudent() {

const searchTerm = document.getElementById('payment-student-search').value.toLowerCase();

if (!searchTerm) {

loadAllPayments();

return;

}

if (!isNaN(searchTerm)) {

// Search by ID

const studentId = parseInt(searchTerm);

loadPaymentsForStudent(studentId);

} else {

// Search by name

const matchingStudents = students.filter(s =>

s.firstName.toLowerCase().includes(searchTerm) ||

s.lastName.toLowerCase().includes(searchTerm)

);

if (matchingStudents.length === 1) {

loadPaymentsForStudent(matchingStudents[0].id);

} else if (matchingStudents.length > 1) {

showToast('Múltiples alumnos encontrados. Use ID para búsqueda específica.', 'warning');

loadAllPayments();

} else {

showToast('No se encontraron alumnos', 'warning');

loadAllPayments();

}

}

}

function loadAllPayments() {

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

tableBody.innerHTML = '';

if (payments.length === 0) {

tableBody.innerHTML = '<tr><td colspan="6">No hay pagos registrados</td></tr>';

return;

}

payments.forEach(payment => {

const student = students.find(s => s.id === payment.studentId);

const studentName = student ? `${student.lastName}, ${student.firstName}` : 'Alumno no encontrado';

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

row.innerHTML = `

<td>${payment.id}</td>

<td>${formatDate(payment.date)}</td>

<td>Q${parseFloat(payment.amount).toFixed(2)}</td>

<td>${payment.method}</td>

<td>${payment.notes || ''}</td>

<td>

<button class="btn btn-secondary btn-sm edit-payment" data-id="${payment.id}">

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

</button>

<button class="btn btn-danger btn-sm delete-payment" data-id="${payment.id}">

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

</button>

</td>

`;

tableBody.appendChild(row);

});

// Add event listeners to action buttons

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

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

editPayment(parseInt(this.dataset.id));

});

});

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

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

deletePayment(parseInt(this.dataset.id));

});

});

}

function loadPaymentsForStudent(studentId) {

const studentPayments = payments.filter(p => p.studentId === studentId);

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

tableBody.innerHTML = '';

if (studentPayments.length === 0) {

tableBody.innerHTML = '<tr><td colspan="6">No hay pagos registrados para este alumno</td></tr>';

return;

}

studentPayments.forEach(payment => {

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

row.innerHTML = `

<td>${payment.id}</td>

<td>${formatDate(payment.date)}</td>

<td>Q${parseFloat(payment.amount).toFixed(2)}</td>

<td>${payment.method}</td>

<td>${payment.notes || ''}</td>

<td>

<button class="btn btn-secondary btn-sm edit-payment" data-id="${payment.id}">

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

</button>

<button class="btn btn-danger btn-sm delete-payment" data-id="${payment.id}">

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

</button>

</td>

`;

tableBody.appendChild(row);

});

// Add event listeners to action buttons

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

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

editPayment(parseInt(this.dataset.id));

});

});

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

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

deletePayment(parseInt(this.dataset.id));

});

});

}

function newPayment() {

const searchTerm = document.getElementById('payment-student-search').value;

if (!searchTerm) {

showToast('Seleccione un alumno primero', 'warning');

return;

}

selectedPaymentId = null;

document.getElementById('payment-modal-title').textContent = 'Nuevo Pago';

document.getElementById('payment-date').value = new Date().toISOString().split('T')[0];

document.getElementById('payment-amount').value = '';

document.getElementById('payment-method').value = '';

document.getElementById('payment-notes').value = '';

document.getElementById('payment-modal-delete').style.display = 'none';

document.getElementById('payment-modal').style.display = 'flex';

}

function editPayment(paymentId) {

const payment = payments.find(p => p.id === paymentId);

if (!payment) return;

selectedPaymentId = paymentId;

document.getElementById('payment-modal-title').textContent = 'Editar Pago';

document.getElementById('payment-date').value = payment.date;

document.getElementById('payment-amount').value = payment.amount;

document.getElementById('payment-method').value = payment.method;

document.getElementById('payment-notes').value = payment.notes || '';

document.getElementById('payment-modal-delete').style.display = 'inline-block';

document.getElementById('payment-modal').style.display = 'flex';

}

function deletePayment(paymentId) {

if (!confirm('¿Está seguro de que desea eliminar este pago?')) {

return;

}

payments = payments.filter(p => p.id !== paymentId);

localStorage.setItem('academy\_payments', JSON.stringify(payments));

showToast('Pago eliminado correctamente');

searchPaymentStudent(); // Refresh the list

refreshDashboard();

}

function exportPaymentsCSV() {

if (payments.length === 0) {

showToast('No hay pagos para exportar', 'warning');

return;

}

// Create CSV content

const headers = ['ID', 'ID Alumno', 'Alumno', 'Fecha', 'Monto', 'Método', 'Notas'];

let csvContent = headers.join(',') + '\n';

payments.forEach(payment => {

const student = students.find(s => s.id === payment.studentId);

const studentName = student ? `${student.lastName}, ${student.firstName}` : 'Alumno no encontrado';

const row = [

payment.id,

payment.studentId,

`"${studentName}"`,

payment.date,

payment.amount,

payment.method,

`"${payment.notes || ''}"`

];

csvContent += row.join(',') + '\n';

});

// Create download link

const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });

const url = URL.createObjectURL(blob);

const link = document.createElement('a');

link.setAttribute('href', url);

link.setAttribute('download', 'pagos.csv');

link.style.visibility = 'hidden';

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

showToast('Archivo CSV exportado correctamente');

}

// Grades tab functions

function initializeGradesTab() {

// Search functionality

document.getElementById('grade-student-search-btn').addEventListener('click', searchGradeStudent);

document.getElementById('grade-student-search').addEventListener('keyup', function(e) {

if (e.key === 'Enter') searchGradeStudent();

});

// New grade button

document.getElementById('btn-new-grade').addEventListener('click', newGrade);

// Export button

document.getElementById('btn-export-grades').addEventListener('click', exportGradesCSV);

}

function searchGradeStudent() {

const searchTerm = document.getElementById('grade-student-search').value.toLowerCase();

if (!searchTerm) {

loadAllGrades();

return;

}

if (!isNaN(searchTerm)) {

// Search by ID

const studentId = parseInt(searchTerm);

loadGradesForStudent(studentId);

} else {

// Search by name

const matchingStudents = students.filter(s =>

s.firstName.toLowerCase().includes(searchTerm) ||

s.lastName.toLowerCase().includes(searchTerm)

);

if (matchingStudents.length === 1) {

loadGradesForStudent(matchingStudents[0].id);

} else if (matchingStudents.length > 1) {

showToast('Múltiples alumnos encontrados. Use ID para búsqueda específica.', 'warning');

loadAllGrades();

} else {

showToast('No se encontraron alumnos', 'warning');

loadAllGrades();

}

}

}

function loadAllGrades() {

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

tableBody.innerHTML = '';

if (grades.length === 0) {

tableBody.innerHTML = '<tr><td colspan="6">No hay calificaciones registradas</td></tr>';

return;

}

grades.forEach(grade => {

const student = students.find(s => s.id === grade.studentId);

const studentName = student ? `${student.lastName}, ${student.firstName}` : 'Alumno no encontrado';

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

row.innerHTML = `

<td>${grade.id}</td>

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

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

<td>${formatDate(grade.date)}</td>

<td>${grade.notes || ''}</td>

<td>

<button class="btn btn-secondary btn-sm edit-grade" data-id="${grade.id}">

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

</button>

<button class="btn btn-danger btn-sm delete-grade" data-id="${grade.id}">

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

</button>

</td>

`;

tableBody.appendChild(row);

});

// Add event listeners to action buttons

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

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

editGrade(parseInt(this.dataset.id));

});

});

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

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

deleteGrade(parseInt(this.dataset.id));

});

});

}

function loadGradesForStudent(studentId) {

const studentGrades = grades.filter(g => g.studentId === studentId);

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

tableBody.innerHTML = '';

if (studentGrades.length === 0) {

tableBody.innerHTML = '<tr><td colspan="6">No hay calificaciones registradas para este alumno</td></tr>';

return;

}

studentGrades.forEach(grade => {

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

row.innerHTML = `

<td>${grade.id}</td>

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

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

<td>${formatDate(grade.date)}</td>

<td>${grade.notes || ''}</td>

<td>

<button class="btn btn-secondary btn-sm edit-grade" data-id="${grade.id}">

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

</button>

<button class="btn btn-danger btn-sm delete-grade" data-id="${grade.id}">

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

</button>

</td>

`;

tableBody.appendChild(row);

});

// Add event listeners to action buttons

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

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

editGrade(parseInt(this.dataset.id));

});

});

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

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

deleteGrade(parseInt(this.dataset.id));

});

});

}

function newGrade() {

const searchTerm = document.getElementById('grade-student-search').value;

if (!searchTerm) {

showToast('Seleccione un alumno primero', 'warning');

return;

}

selectedGradeId = null;

document.getElementById('grade-modal-title').textContent = 'Nueva Calificación';

document.getElementById('grade-subject').value = '';

document.getElementById('grade-value').value = '';

document.getElementById('grade-date').value = new Date().toISOString().split('T')[0];

document.getElementById('grade-notes').value = '';

document.getElementById('grade-modal-delete').style.display = 'none';

document.getElementById('grade-modal').style.display = 'flex';

}

function editGrade(gradeId) {

const grade = grades.find(g => g.id === gradeId);

if (!grade) return;

selectedGradeId = gradeId;

document.getElementById('grade-modal-title').textContent = 'Editar Calificación';

document.getElementById('grade-subject').value = grade.subject;

document.getElementById('grade-value').value = grade.grade;

document.getElementById('grade-date').value = grade.date;

document.getElementById('grade-notes').value = grade.notes || '';

document.getElementById('grade-modal-delete').style.display = 'inline-block';

document.getElementById('grade-modal').style.display = 'flex';

}

function deleteGrade(gradeId) {

if (!confirm('¿Está seguro de que desea eliminar esta calificación?')) {

return;

}

grades = grades.filter(g => g.id !== gradeId);

localStorage.setItem('academy\_grades', JSON.stringify(grades));

showToast('Calificación eliminada correctamente');

searchGradeStudent(); // Refresh the list

refreshDashboard();

}

function exportGradesCSV() {

if (grades.length === 0) {

showToast('No hay calificaciones para exportar', 'warning');

return;

}

// Create CSV content

const headers = ['ID', 'ID Alumno', 'Alumno', 'Asignatura', 'Calificación', 'Fecha', 'Notas'];

let csvContent = headers.join(',') + '\n';

grades.forEach(grade => {

const student = students.find(s => s.id === grade.studentId);

const studentName = student ? `${student.lastName}, ${student.firstName}` : 'Alumno no encontrado';

const row = [

grade.id,

grade.studentId,

`"${studentName}"`,

grade.subject,

grade.grade,

grade.date,

`"${grade.notes || ''}"`

];

csvContent += row.join(',') + '\n';

});

// Create download link

const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });

const url = URL.createObjectURL(blob);

const link = document.createElement('a');

link.setAttribute('href', url);

link.setAttribute('download', 'calificaciones.csv');

link.style.visibility = 'hidden';

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

showToast('Archivo CSV exportado correctamente');

}

// Modal functions

function initializeModals() {

// Payment modal

document.querySelector('#payment-modal .modal-close').addEventListener('click', closePaymentModal);

document.getElementById('payment-modal-cancel').addEventListener('click', closePaymentModal);

document.getElementById('payment-modal-save').addEventListener('click', savePayment);

document.getElementById('payment-modal-delete').addEventListener('click', deletePaymentFromModal);

// Grade modal

document.querySelector('#grade-modal .modal-close').addEventListener('click', closeGradeModal);

document.getElementById('grade-modal-cancel').addEventListener('click', closeGradeModal);

document.getElementById('grade-modal-save').addEventListener('click', saveGrade);

document.getElementById('grade-modal-delete').addEventListener('click', deleteGradeFromModal);

// Close modals when clicking outside

window.addEventListener('click', function(e) {

if (e.target === document.getElementById('payment-modal')) {

closePaymentModal();

}

if (e.target === document.getElementById('grade-modal')) {

closeGradeModal();

}

});

}

function closePaymentModal() {

document.getElementById('payment-modal').style.display = 'none';

}

function closeGradeModal() {

document.getElementById('grade-modal').style.display = 'none';

}

function savePayment() {

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

const amount = parseFloat(document.getElementById('payment-amount').value);

const method = document.getElementById('payment-method').value;

if (!date || isNaN(amount) || !method) {

showToast('Fecha, monto y método son obligatorios', 'error');

return;

}

// Get student ID from search field

const searchTerm = document.getElementById('payment-student-search').value;

let studentId = null;

if (searchTerm && searchTerm.includes('-')) {

studentId = parseInt(searchTerm.split('-')[0].trim());

}

if (!studentId) {

showToast('Seleccione un alumno válido', 'error');

return;

}

const paymentData = {

date,

amount,

method,

notes: document.getElementById('payment-notes').value.trim() || null,

studentId: studentId

};

if (selectedPaymentId) {

// Update existing payment

const index = payments.findIndex(p => p.id === selectedPaymentId);

if (index !== -1) {

paymentData.id = selectedPaymentId;

payments[index] = paymentData;

}

} else {

// Create new payment

paymentData.id = payments.length > 0 ? Math.max(...payments.map(p => p.id)) + 1 : 1;

payments.push(paymentData);

}

// Save to localStorage

localStorage.setItem('academy\_payments', JSON.stringify(payments));

showToast('Pago guardado correctamente');

closePaymentModal();

loadPaymentsForStudent(studentId); // Refresh the list

refreshDashboard();

}

function deletePaymentFromModal() {

if (!selectedPaymentId) return;

if (!confirm('¿Está seguro de que desea eliminar este pago?')) {

return;

}

payments = payments.filter(p => p.id !== selectedPaymentId);

localStorage.setItem('academy\_payments', JSON.stringify(payments));

showToast('Pago eliminado correctamente');

closePaymentModal();

searchPaymentStudent(); // Refresh the list

refreshDashboard();

}

function saveGrade() {

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

const gradeValue = parseFloat(document.getElementById('grade-value').value);

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

if (!subject || isNaN(gradeValue) || !date) {

showToast('Asignatura, calificación y fecha son obligatorios', 'error');

return;

}

if (gradeValue < 0 || gradeValue > 100) {

showToast('La calificación debe estar entre 0 y 100', 'error');

return;

}

// Get student ID from search field

const searchTerm = document.getElementById('grade-student-search').value;

let studentId = null;

if (searchTerm && searchTerm.includes('-')) {

studentId = parseInt(searchTerm.split('-')[0].trim());

}

if (!studentId) {

showToast('Seleccione un alumno válido', 'error');

return;

}

const gradeData = {

subject,

grade: gradeValue,

date,

notes: document.getElementById('grade-notes').value.trim() || null,

studentId: studentId

};

if (selectedGradeId) {

// Update existing grade

const index = grades.findIndex(g => g.id === selectedGradeId);

if (index !== -1) {

gradeData.id = selectedGradeId;

grades[index] = gradeData;

}

} else {

// Create new grade

gradeData.id = grades.length > 0 ? Math.max(...grades.map(g => g.id)) + 1 : 1;

grades.push(gradeData);

}

// Save to localStorage

localStorage.setItem('academy\_grades', JSON.stringify(grades));

showToast('Calificación guardada correctamente');

closeGradeModal();

loadGradesForStudent(studentId); // Refresh the list

refreshDashboard();

}

function deleteGradeFromModal() {

if (!selectedGradeId) return;

if (!confirm('¿Está seguro de que desea eliminar esta calificación?')) {

return;

}

grades = grades.filter(g => g.id !== selectedGradeId);

localStorage.setItem('academy\_grades', JSON.stringify(grades));

showToast('Calificación eliminada correctamente');

closeGradeModal();

searchGradeStudent(); // Refresh the list

refreshDashboard();

}

// Utility functions

function formatDate(dateString) {

if (!dateString) return '';

const date = new Date(dateString);

return date.toLocaleDateString('es-ES');

}

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

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

const toastMessage = document.getElementById('toast-message');

toastMessage.textContent = message;

toast.className = 'toast';

if (type === 'error') {

toast.classList.add('error');

} else if (type === 'warning') {

toast.classList.add('warning');

}

toast.classList.add('show');

setTimeout(() => {

toast.classList.remove('show');

}, 3000);

}

</script>

</body>

</html>