| <!DOCTYPE html> |
| --- |
| <html lang="es"> |
| <head> |
| <meta charset="UTF-8"> |
| <meta name="viewport" content="width=device-width, initial-scale=1.0"> |
| <title>Farmacia El Ahorro - Sistema de Gestión</title> |
| <link href="<https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css>" rel="stylesheet"> |
| <link rel="stylesheet" href="<https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.0/font/bootstrap-icons.css>"> |
| <style> |
| .sidebar { |
| min-height: 100vh; |
| background-color: #f8f9fa; |
| border-right: 1px solid #dee2e6; |
| } |
| .card-counter { |
| box-shadow: 0 4px 8px rgba(0,0,0,0.1); |
| transition: all 0.3s; |
| } |
| .card-counter:hover { |
| transform: translateY(-5px); |
| box-shadow: 0 8px 16px rgba(0,0,0,0.2); |
| } |
| .navbar-brand { |
| font-weight: bold; |
| } |
| .active-nav { |
| background-color: #0d6efd; |
| color: white !important; |
| } |
| .table-responsive { |
| min-height: 400px; |
| } |
| .search-box { |
| max-width: 300px; |
| } |
| .producto-item { |
| cursor: pointer; |
| } |
| .producto-item:hover { |
| background-color: #f8f9fa; |
| } |
| .barcode-scanner { |
| position: relative; |
| width: 100%; |
| max-width: 400px; |
| margin: 0 auto; |
| } |
| .barcode-video { |
| width: 100%; |
| border: 2px solid #0d6efd; |
| border-radius: 5px; |
| } |
| .barcode-overlay { |
| position: absolute; |
| top: 50%; |
| left: 50%; |
| transform: translate(-50%, -50%); |
| width: 80%; |
| height: 2px; |
| background-color: red; |
| z-index: 10; |
| } |
| .scanner-controls { |
| margin-top: 10px; |
| text-align: center; |
| } |
| @media print { |
| body \* { |
| visibility: hidden; |
| } |
| #recibo-print, #recibo-print \* { |
| visibility: visible; |
| } |
| #recibo-print { |
| position: absolute; |
| left: 0; |
| top: 0; |
| width: 100%; |
| } |
| .no-print { |
| display: none !important; |
| } |
| } |
| .receipt { |
| font-family: 'Courier New', monospace; |
| width: 80mm; |
| margin: 0 auto; |
| padding: 10px; |
| font-size: 12px; |
| } |
| .receipt-header, .receipt-footer { |
| text-align: center; |
| margin-bottom: 10px; |
| } |
| .receipt-items { |
| width: 100%; |
| border-collapse: collapse; |
| margin: 10px 0; |
| } |
| .receipt-items th, .receipt-items td { |
| padding: 4px 0; |
| border-bottom: 1px dashed #ccc; |
| } |
| .receipt-total { |
| margin-top: 10px; |
| border-top: 2px solid #000; |
| padding-top: 10px; |
| font-weight: bold; |
| } |
| </style> |
| </head> |
| <body> |
| <!-- Navbar --> |
| <nav class="navbar navbar-expand-lg navbar-dark bg-primary"> |
| <div class="container-fluid"> |
| <a class="navbar-brand" href="#"> |
| <i class="bi bi-prescription2"></i> Farmacia El Ahorro |
| </a> |
| </div> |
| </nav> |
|  |
| <div class="container-fluid"> |
| <div class="row"> |
| <!-- Sidebar --> |
| <div class="col-md-3 col-lg-2 sidebar py-3"> |
| <div class="list-group"> |
| <a href="#" class="list-group-item list-group-item-action active-nav" onclick="showSection('dashboard')"> |
| <i class="bi bi-speedometer2"></i> Dashboard |
| </a> |
| <a href="#" class="list-group-item list-group-item-action" onclick="showSection('productos')"> |
| <i class="bi bi-capsule"></i> Productos |
| </a> |
| <a href="#" class="list-group-item list-group-item-action" onclick="showSection('clientes')"> |
| <i class="bi bi-people"></i> Clientes |
| </a> |
| <a href="#" class="list-group-item list-group-item-action" onclick="showSection('proveedores')"> |
| <i class="bi bi-truck"></i> Proveedores |
| </a> |
| <a href="#" class="list-group-item list-group-item-action" onclick="showSection('empleados')"> |
| <i class="bi bi-person-badge"></i> Empleados |
| </a> |
| <a href="#" class="list-group-item list-group-item-action" onclick="showSection('ventas')"> |
| <i class="bi bi-cash-coin"></i> Ventas |
| </a> |
| </div> |
| </div> |
|  |
| <!-- Main Content --> |
| <div class="col-md-9 col-lg-10 py-3"> |
| <!-- Dashboard Section --> |
| <div id="dashboard-section"> |
| <h2 class="mb-4">Dashboard</h2> |
| <div class="row"> |
| <div class="col-md-3 mb-3"> |
| <div class="card card-counter bg-primary text-white text-center p-3"> |
| <h3><i class="bi bi-capsule"></i></h3> |
| <h4 id="productos-count">0</h4> |
| <p>Productos</p> |
| </div> |
| </div> |
| <div class="col-md-3 mb-3"> |
| <div class="card card-counter bg-success text-white text-center p-3"> |
| <h3><i class="bi bi-people"></i></h3> |
| <h4 id="clientes-count">0</h4> |
| <p>Clientes</p> |
| </div> |
| </div> |
| <div class="col-md-3 mb-3"> |
| <div class="card card-counter bg-warning text-white text-center p-3"> |
| <h3><i class="bi bi-truck"></i></h3> |
| <h4 id="proveedores-count">0</h4> |
| <p>Proveedores</p> |
| </div> |
| </div> |
| <div class="col-md-3 mb-3"> |
| <div class="card card-counter bg-info text-white text-center p-3"> |
| <h3><i class="bi bi-person-badge"></i></h3> |
| <h4 id="empleados-count">0</h4> |
| <p>Empleados</p> |
| </div> |
| </div> |
| <div class="col-md-3 mb-3"> |
| <div class="card card-counter bg-danger text-white text-center p-3"> |
| <h3><i class="bi bi-cash-coin"></i></h3> |
| <h4 id="ventas-count">0</h4> |
| <p>Ventas Hoy</p> |
| </div> |
| </div> |
| <div class="col-md-3 mb-3"> |
| <div class="card card-counter bg-secondary text-white text-center p-3"> |
| <h3><i class="bi bi-currency-dollar"></i></h3> |
| <h4 id="ingresos-hoy">$0.00</h4> |
| <p>Ingresos Hoy</p> |
| </div> |
| </div> |
| </div> |
| </div> |
|  |
| <!-- Productos Section --> |
| <div id="productos-section" style="display: none;"> |
| <div class="d-flex justify-content-between align-items-center mb-3"> |
| <h2>Gestión de Productos</h2> |
| <button class="btn btn-primary" onclick="showProductoForm()"> |
| <i class="bi bi-plus-circle"></i> Nuevo Producto |
| </button> |
| </div> |
|  |
| <div class="mb-3"> |
| <div class="input-group search-box"> |
| <input type="text" id="search-producto" class="form-control" placeholder="Buscar producto..."> |
| <button class="btn btn-outline-secondary" onclick="buscarProductos()"> |
| <i class="bi bi-search"></i> |
| </button> |
| </div> |
| </div> |
|  |
| <div class="table-responsive"> |
| <table class="table table-striped table-hover"> |
| <thead> |
| <tr> |
| <th>ID</th> |
| <th>Nombre</th> |
| <th>Código Barras</th> |
| <th>Stock</th> |
| <th>Precio Unitario</th> |
| <th>Acciones</th> |
| </tr> |
| </thead> |
| <tbody id="productos-table"> |
| <!-- Los productos se cargarán aquí --> |
| </tbody> |
| </table> |
| </div> |
| </div> |
|  |
| <!-- Clientes Section --> |
| <div id="clientes-section" style="display: none;"> |
| <div class="d-flex justify-content-between align-items-center mb-3"> |
| <h2>Gestión de Clientes</h2> |
| <button class="btn btn-primary" onclick="showClienteForm()"> |
| <i class="bi bi-plus-circle"></i> Nuevo Cliente |
| </button> |
| </div> |
|  |
| <div class="mb-3"> |
| <div class="input-group search-box"> |
| <input type="text" id="search-cliente" class="form-control" placeholder="Buscar cliente..."> |
| <button class="btn btn-outline-secondary" onclick="buscarClientes()"> |
| <i class="bi bi-search"></i> |
| </button> |
| </div> |
| </div> |
|  |
| <div class="table-responsive"> |
| <table class="table table-striped table-hover"> |
| <thead> |
| <tr> |
| <th>ID</th> |
| <th>Nombre</th> |
| <th>Teléfono</th> |
| <th>Acciones</th> |
| </tr> |
| </thead> |
| <tbody id="clientes-table"> |
| <!-- Los clientes se cargarán aquí --> |
| </tbody> |
| </table> |
| </div> |
| </div> |
|  |
| <!-- Proveedores Section --> |
| <div id="proveedores-section" style="display: none;"> |
| <div class="d-flex justify-content-between align-items-center mb-3"> |
| <h2>Gestión de Proveedores</h2> |
| <button class="btn btn-primary" onclick="showProveedorForm()"> |
| <i class="bi bi-plus-circle"></i> Nuevo Proveedor |
| </button> |
| </div> |
|  |
| <div class="mb-3"> |
| <div class="input-group search-box"> |
| <input type="text" id="search-proveedor" class="form-control" placeholder="Buscar proveedor..."> |
| <button class="btn btn-outline-secondary" onclick="buscarProveedores()"> |
| <i class="bi bi-search"></i> |
| </button> |
| </div> |
| </div> |
|  |
| <div class="table-responsive"> |
| <table class="table table-striped table-hover"> |
| <thead> |
| <tr> |
| <th>ID</th> |
| <th>Nombre</th> |
| <th>Contacto</th> |
| <th>Acciones</th> |
| </tr> |
| </thead> |
| <tbody id="proveedores-table"> |
| <!-- Los proveedores se cargarán aquí --> |
| </tbody> |
| </table> |
| </div> |
| </div> |
|  |
| <!-- Empleados Section --> |
| <div id="empleados-section" style="display: none;"> |
| <div class="d-flex justify-content-between align-items-center mb-3"> |
| <h2>Gestión de Empleados</h2> |
| <button class="btn btn-primary" onclick="showEmpleadoForm()"> |
| <i class="bi bi-plus-circle"></i> Nuevo Empleado |
| </button> |
| </div> |
|  |
| <div class="mb-3"> |
| <div class="input-group search-box"> |
| <input type="text" id="search-empleado" class="form-control" placeholder="Buscar empleado..."> |
| <button class="btn btn-outline-secondary" onclick="buscarEmpleados()"> |
| <i class="bi bi-search"></i> |
| </button> |
| </div> |
| </div> |
|  |
| <div class="table-responsive"> |
| <table class="table table-striped table-hover"> |
| <thead> |
| <tr> |
| <th>ID</th> |
| <th>Nombre</th> |
| <th>Cargo</th> |
| <th>Salario</th> |
| <th>Acciones</th> |
| </tr> |
| </thead> |
| <tbody id="empleados-table"> |
| <!-- Los empleados se cargarán aquí --> |
| </tbody> |
| </table> |
| </div> |
| </div> |
|  |
| <!-- Ventas Section --> |
| <div id="ventas-section" style="display: none;"> |
| <div class="d-flex justify-content-between align-items-center mb-3"> |
| <h2>Gestión de Ventas</h2> |
| <button class="btn btn-primary" onclick="showVentaForm()"> |
| <i class="bi bi-plus-circle"></i> Nueva Venta |
| </button> |
| </div> |
|  |
| <div class="mb-3"> |
| <div class="input-group search-box"> |
| <input type="text" id="search-venta" class="form-control" placeholder="Buscar venta por cliente..."> |
| <button class="btn btn-outline-secondary" onclick="buscarVentas()"> |
| <i class="bi bi-search"></i> |
| </button> |
| </div> |
| </div> |
|  |
| <div class="table-responsive"> |
| <table class="table table-striped table-hover"> |
| <thead> |
| <tr> |
| <th>ID</th> |
| <th>Fecha</th> |
| <th>Cliente</th> |
| <th>NIT</th> |
| <th>Total</th> |
| <th>Acciones</th> |
| </tr> |
| </thead> |
| <tbody id="ventas-table"> |
| <!-- Las ventas se cargarán aquí --> |
| </tbody> |
| </table> |
| </div> |
| </div> |
| </div> |
| </div> |
| </div> |
|  |
| <!-- Modal para formularios --> |
| <div class="modal fade" id="formModal" tabindex="-1" aria-hidden="true"> |
| <div class="modal-dialog modal-lg"> |
| <div class="modal-content"> |
| <div class="modal-header"> |
| <h5 class="modal-title" id="modalTitle">Formulario</h5> |
| <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button> |
| </div> |
| <div class="modal-body" id="modalBody"> |
| <!-- El formulario se cargará aquí --> |
| </div> |
| </div> |
| </div> |
| </div> |
|  |
| <!-- Modal para escaner de código de barras --> |
| <div class="modal fade" id="barcodeModal" tabindex="-1" aria-hidden="true"> |
| <div class="modal-dialog"> |
| <div class="modal-content"> |
| <div class="modal-header"> |
| <h5 class="modal-title">Escaner de Código de Barras</h5> |
| <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button> |
| </div> |
| <div class="modal-body"> |
| <div class="barcode-scanner"> |
| <video id="barcode-video" class="barcode-video"></video> |
| <div class="barcode-overlay"></div> |
| </div> |
| <div class="scanner-controls"> |
| <button id="start-scan" class="btn btn-success"> |
| <i class="bi bi-camera"></i> Iniciar Escáner |
| </button> |
| <button id="stop-scan" class="btn btn-danger" style="display: none;"> |
| <i class="bi bi-stop-circle"></i> Detener Escáner |
| </button> |
| </div> |
| <div class="mt-3"> |
| <label for="manual-barcode" class="form-label">O ingresar código manualmente:</label> |
| <div class="input-group"> |
| <input type="text" id="manual-barcode" class="form-control" placeholder="Código de barras"> |
| <button class="btn btn-primary" onclick="addProductByManualBarcode()"> |
| <i class="bi bi-plus-circle"></i> Agregar |
| </button> |
| </div> |
| </div> |
| </div> |
| </div> |
| </div> |
| </div> |
|  |
| <!-- Recibo de venta (oculto para impresión) --> |
| <div id="recibo-print" style="display: none;"> |
| <div class="receipt"> |
| <div class="receipt-header"> |
| <h4>Farmacia El Ahorro</h4> |
| <p>Tel: (502) 1234-5678</p> |
| <p>Dirección: Av. Principal 123</p> |
| <p>NIT: 1234567-8</p> |
| </div> |
| <hr> |
| <div class="receipt-details"> |
| <p><strong>Recibo No:</strong> <span id="recibo-numero"></span></p> |
| <p><strong>Fecha:</strong> <span id="recibo-fecha"></span></p> |
| <p><strong>Cliente:</strong> <span id="recibo-cliente"></span></p> |
| <p><strong>NIT:</strong> <span id="recibo-nit"></span></p> |
| </div> |
| <hr> |
| <table class="receipt-items"> |
| <thead> |
| <tr> |
| <th style="width: 60%;">Producto</th> |
| <th style="width: 15%;">Cant</th> |
| <th style="width: 25%; text-align: right;">Precio</th> |
| </tr> |
| </thead> |
| <tbody id="recibo-items"> |
| <!-- Los items del recibo se insertarán aquí --> |
| </tbody> |
| </table> |
| <hr> |
| <div class="receipt-total"> |
| <p style="text-align: right;">Total: <span id="recibo-total"></span></p> |
| </div> |
| <div class="receipt-footer"> |
| <p>¡Gracias por su compra!</p> |
| <p>Vuelva pronto</p> |
| </div> |
| </div> |
| </div> |
|  |
| <!-- Toast para mensajes --> |
| <div class="toast-container position-fixed top-0 end-0 p-3"> |
| <div id="liveToast" class="toast" role="alert" aria-live="assertive" aria-atomic="true"> |
| <div class="toast-header"> |
| <strong class="me-auto" id="toastTitle">Notificación</strong> |
| <button type="button" class="btn-close" data-bs-dismiss="toast" aria-label="Close"></button> |
| </div> |
| <div class="toast-body" id="toastMessage"> |
| Mensaje aquí |
| </div> |
| </div> |
| </div> |
|  |
| <!-- Incluir QuaggaJS para lectura de código de barras --> |
| <script src="<https://cdn.jsdelivr.net/npm/quagga@0.12.1/dist/quagga.min.js>"></script> |
| <script src="<https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js>"></script> |
| <script> |
| // Variables globales |
| let db; |
| let currentSection = 'dashboard'; |
| let editingId = null; |
| const modal = new bootstrap.Modal(document.getElementById('formModal')); |
| const barcodeModal = new bootstrap.Modal(document.getElementById('barcodeModal')); |
| const toast = new bootstrap.Toast(document.getElementById('liveToast')); |
| let quaggaInitialized = false; |
|  |
| // Inicializar la base de datos al cargar la página |
| document.addEventListener('DOMContentLoaded', function() { |
| initDB(); |
| updateCounts(); |
| showSection('dashboard'); |
| setupBarcodeScanner(); |
| }); |
|  |
| // Configurar el escáner de código de barras |
| function setupBarcodeScanner() { |
| document.getElementById('start-scan').addEventListener('click', startBarcodeScanner); |
| document.getElementById('stop-scan').addEventListener('click', stopBarcodeScanner); |
| } |
|  |
| // Inicializar IndexedDB |
| function initDB() { |
| const request = indexedDB.open('FarmaciaDB', 3); // Incrementamos la versión a 3 |
|  |
| request.onerror = function(event) { |
| showToast('Error', 'No se pudo abrir la base de datos', 'danger'); |
| }; |
|  |
| request.onupgradeneeded = function(event) { |
| db = event.target.result; |
|  |
| // Crear object stores (tablas) |
| if (!db.objectStoreNames.contains('producto')) { |
| const productoStore = db.createObjectStore('producto', { keyPath: 'id', autoIncrement: true }); |
| productoStore.createIndex('nombre', 'nombre', { unique: false }); |
| productoStore.createIndex('codigo\_barras', 'codigo\_barras', { unique: true }); |
| } |
|  |
| if (!db.objectStoreNames.contains('cliente')) { |
| const clienteStore = db.createObjectStore('cliente', { keyPath: 'id', autoIncrement: true }); |
| clienteStore.createIndex('nombre', 'nombre', { unique: false }); |
| } |
|  |
| if (!db.objectStoreNames.contains('proveedor')) { |
| const proveedorStore = db.createObjectStore('proveedor', { keyPath: 'id', autoIncrement: true }); |
| proveedorStore.createIndex('nombre', 'nombre', { unique: false }); |
| } |
|  |
| if (!db.objectStoreNames.contains('empleado')) { |
| const empleadoStore = db.createObjectStore('empleado', { keyPath: 'id', autoIncrement: true }); |
| empleadoStore.createIndex('nombre', 'nombre', { unique: false }); |
| } |
|  |
| // Crear object store para ventas |
| if (!db.objectStoreNames.contains('venta')) { |
| const ventaStore = db.createObjectStore('venta', { keyPath: 'id', autoIncrement: true }); |
| ventaStore.createIndex('fecha', 'fecha', { unique: false }); |
| ventaStore.createIndex('cliente', 'cliente', { unique: false }); |
| } |
| }; |
|  |
| request.onsuccess = function(event) { |
| db = event.target.result; |
| showToast('Éxito', 'Base de datos inicializada correctamente', 'success'); |
| cargarDatosIniciales(); |
| }; |
| } |
|  |
| // Cargar datos iniciales si es necesario |
| function cargarDatosIniciales() { |
| // Verificar si ya hay datos |
| const transaction = db.transaction(['producto'], 'readonly'); |
| const store = transaction.objectStore('producto'); |
| const countRequest = store.count(); |
|  |
| countRequest.onsuccess = function() { |
| if (countRequest.result === 0) { |
| // Datos de ejemplo |
| const datosEjemplo = { |
| producto: [ |
| { nombre: 'Paracetamol 500mg', codigo\_barras: '7501006554025', stock: 100, precio\_unitario: 5.50 }, |
| { nombre: 'Ibuprofeno 400mg', codigo\_barras: '7501059253028', stock: 80, precio\_unitario: 6.20 }, |
| { nombre: 'Amoxicilina 500mg', codigo\_barras: '7501055301029', stock: 50, precio\_unitario: 12.80 } |
| ], |
| cliente: [ |
| { nombre: 'Juan Pérez', telefono: '555-1234', nit: '123456-7', direccion: 'Av. Siempre Viva 742' }, |
| { nombre: 'María García', telefono: '555-5678', nit: '765432-1', direccion: 'Calle Falsa 123' } |
| ], |
| proveedor: [ |
| { nombre: 'Farmacéutica Central', contacto: 'contacto@farmaceuticacentral.com' }, |
| { nombre: 'Medicamentos S.A.', contacto: 'ventas@medicamentossa.com' } |
| ], |
| empleado: [ |
| { nombre: 'Carlos López', cargo: 'Farmacéutico', salario: 2500 }, |
| { nombre: 'Ana Martínez', cargo: 'Cajera', salario: 1500 } |
| ] |
| }; |
|  |
| // Insertar datos de ejemplo |
| for (const [tabla, datos] of Object.entries(datosEjemplo)) { |
| const transaction = db.transaction([tabla], 'readwrite'); |
| const store = transaction.objectStore(tabla); |
|  |
| datos.forEach(item => { |
| store.add(item); |
| }); |
| } |
|  |
| showToast('Info', 'Datos de ejemplo cargados', 'info'); |
| } |
| updateCounts(); |
| cargarProductos(); |
| cargarClientes(); |
| cargarProveedores(); |
| cargarEmpleados(); |
| cargarVentas(); |
| }; |
| } |
|  |
| // Mostrar u ocultar secciones |
| function showSection(section) { |
| // Ocultar todas las secciones |
| document.querySelectorAll('[id$="-section"]').forEach(sec => { |
| sec.style.display = 'none'; |
| }); |
|  |
| // Mostrar la sección seleccionada |
| document.getElementById(section + '-section').style.display = 'block'; |
| currentSection = section; |
|  |
| // Actualizar datos según la sección |
| switch(section) { |
| case 'productos': |
| cargarProductos(); |
| break; |
| case 'clientes': |
| cargarClientes(); |
| break; |
| case 'proveedores': |
| cargarProveedores(); |
| break; |
| case 'empleados': |
| cargarEmpleados(); |
| break; |
| case 'ventas': |
| cargarVentas(); |
| break; |
| case 'dashboard': |
| updateCounts(); |
| break; |
| } |
| } |
|  |
| // Actualizar contadores del dashboard |
| function updateCounts() { |
| contarRegistros('producto', 'productos-count'); |
| contarRegistros('cliente', 'clientes-count'); |
| contarRegistros('proveedor', 'proveedores-count'); |
| contarRegistros('empleado', 'empleados-count'); |
| contarRegistros('venta', 'ventas-count'); |
|  |
| // Calcular ingresos de hoy |
| calcularIngresosHoy(); |
| } |
|  |
| function calcularIngresosHoy() { |
| const transaction = db.transaction(['venta'], 'readonly'); |
| const store = transaction.objectStore('venta'); |
| const index = store.index('fecha'); |
| const hoy = new Date().toISOString().split('T')[0]; // Formato YYYY-MM-DD |
|  |
| let totalHoy = 0; |
|  |
| // No podemos buscar directamente por fecha sin un rango, así que recorremos todas |
| const request = store.getAll(); |
|  |
| request.onsuccess = function() { |
| const ventas = request.result; |
| ventas.forEach(venta => { |
| if (venta.fecha === hoy) { |
| totalHoy += venta.total; |
| } |
| }); |
|  |
| document.getElementById('ingresos-hoy').textContent = `$${totalHoy.toFixed(2)}`; |
| }; |
| } |
|  |
| function contarRegistros(tabla, elementId) { |
| const transaction = db.transaction([tabla], 'readonly'); |
| const store = transaction.objectStore(tabla); |
| const countRequest = store.count(); |
|  |
| countRequest.onsuccess = function() { |
| document.getElementById(elementId).textContent = countRequest.result; |
| }; |
| } |
|  |
| // Mostrar toast de notificación |
| function showToast(title, message, type = 'info') { |
| document.getElementById('toastTitle').textContent = title; |
| document.getElementById('toastMessage').textContent = message; |
|  |
| const toastElement = document.getElementById('liveToast'); |
| toastElement.className = 'toast'; |
|  |
| switch(type) { |
| case 'success': |
| toastElement.classList.add('bg-success', 'text-white'); |
| break; |
| case 'danger': |
| toastElement.classList.add('bg-danger', 'text-white'); |
| break; |
| case 'warning': |
| toastElement.classList.add('bg-warning', 'text-dark'); |
| break; |
| default: |
| toastElement.classList.add('bg-info', 'text-white'); |
| } |
|  |
| toast.show(); |
| } |
|  |
| // ========== FUNCIONES PARA PRODUCTOS ========== |
| function cargarProductos(busqueda = '') { |
| const transaction = db.transaction(['producto'], 'readonly'); |
| const store = transaction.objectStore('producto'); |
| const request = store.getAll(); |
|  |
| request.onsuccess = function() { |
| let productos = request.result; |
|  |
| // Filtrar si hay búsqueda |
| if (busqueda) { |
| productos = productos.filter(p => |
| p.nombre.toLowerCase().includes(busqueda.toLowerCase()) || |
| (p.codigo\_barras && p.codigo\_barras.includes(busqueda)) |
| ); |
| } |
|  |
| const tbody = document.getElementById('productos-table'); |
| tbody.innerHTML = ''; |
|  |
| productos.forEach(producto => { |
| const tr = document.createElement('tr'); |
| tr.innerHTML = ` |
| <td>${producto.id}</td> |
| <td>${producto.nombre}</td> |
| <td>${producto.codigo\_barras || 'N/A'}</td> |
| <td>${producto.stock}</td> |
| <td>$${producto.precio\_unitario.toFixed(2)}</td> |
| <td> |
| <button class="btn btn-sm btn-warning" onclick="editarProducto(${producto.id})"> |
| <i class="bi bi-pencil"></i> |
| </button> |
| <button class="btn btn-sm btn-danger" onclick="eliminarProducto(${producto.id})"> |
| <i class="bi bi-trash"></i> |
| </button> |
| </td> |
| `; |
| tbody.appendChild(tr); |
| }); |
| }; |
| } |
|  |
| function buscarProductos() { |
| const busqueda = document.getElementById('search-producto').value; |
| cargarProductos(busqueda); |
| } |
|  |
| function showProductoForm(producto = null) { |
| editingId = producto ? producto.id : null; |
|  |
| document.getElementById('modalTitle').textContent = producto ? 'Editar Producto' : 'Nuevo Producto'; |
|  |
| document.getElementById('modalBody').innerHTML = ` |
| <form onsubmit="guardarProducto(event)"> |
| <div class="mb-3"> |
| <label for="nombre" class="form-label">Nombre</label> |
| <input type="text" class="form-control" id="nombre" value="${producto ? producto.nombre : ''}" required> |
| </div> |
| <div class="mb-3"> |
| <label for="codigo\_barras" class="form-label">Código de Barras</label> |
| <input type="text" class="form-control" id="codigo\_barras" value="${producto ? producto.codigo\_barras || '' : ''}"> |
| </div> |
| <div class="mb-3"> |
| <label for="stock" class="form-label">Stock</label> |
| <input type="number" class="form-control" id="stock" value="${producto ? producto.stock : 0}" required> |
| </div> |
| <div class="mb-3"> |
| <label for="precio" class="form-label">Precio Unitario</label> |
| <input type="number" step="0.01" class="form-control" id="precio" value="${producto ? producto.precio\_unitario : 0.0}" required> |
| </div> |
| <div class="modal-footer"> |
| <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button> |
| <button type="submit" class="btn btn-primary">Guardar</button> |
| </div> |
| </form> |
| `; |
|  |
| modal.show(); |
| } |
|  |
| function guardarProducto(event) { |
| event.preventDefault(); |
|  |
| const nombre = document.getElementById('nombre').value; |
| const codigo\_barras = document.getElementById('codigo\_barras').value; |
| const stock = parseInt(document.getElementById('stock').value); |
| const precio = parseFloat(document.getElementById('precio').value); |
|  |
| if (!nombre) { |
| showToast('Error', 'El nombre es obligatorio', 'danger'); |
| return; |
| } |
|  |
| const producto = { nombre, codigo\_barras, stock, precio\_unitario: precio }; |
|  |
| const transaction = db.transaction(['producto'], 'readwrite'); |
| const store = transaction.objectStore('producto'); |
|  |
| let request; |
| if (editingId) { |
| producto.id = editingId; |
| request = store.put(producto); |
| } else { |
| request = store.add(producto); |
| } |
|  |
| request.onsuccess = function() { |
| modal.hide(); |
| showToast('Éxito', `Producto ${editingId ? 'actualizado' : 'agregado'} correctamente`, 'success'); |
| cargarProductos(); |
| updateCounts(); |
| editingId = null; |
| }; |
|  |
| request.onerror = function() { |
| showToast('Error', 'No se pudo guardar el producto', 'danger'); |
| }; |
| } |
|  |
| function editarProducto(id) { |
| const transaction = db.transaction(['producto'], 'readonly'); |
| const store = transaction.objectStore('producto'); |
| const request = store.get(id); |
|  |
| request.onsuccess = function() { |
| showProductoForm(request.result); |
| }; |
| } |
|  |
| function eliminarProducto(id) { |
| if (confirm('¿Está seguro de que desea eliminar este producto?')) { |
| const transaction = db.transaction(['producto'], 'readwrite'); |
| const store = transaction.objectStore('producto'); |
| const request = store.delete(id); |
|  |
| request.onsuccess = function() { |
| showToast('Éxito', 'Producto eliminado correctamente', 'success'); |
| cargarProductos(); |
| updateCounts(); |
| }; |
| } |
| } |
|  |
| // ========== FUNCIONES PARA CLIENTES ========== |
| function cargarClientes(busqueda = '') { |
| const transaction = db.transaction(['cliente'], 'readonly'); |
| const store = transaction.objectStore('cliente'); |
| const request = store.getAll(); |
|  |
| request.onsuccess = function() { |
| let clientes = request.result; |
|  |
| if (busqueda) { |
| clientes = clientes.filter(c => |
| c.nombre.toLowerCase().includes(busqueda.toLowerCase()) |
| ); |
| } |
|  |
| const tbody = document.getElementById('clientes-table'); |
| tbody.innerHTML = ''; |
|  |
| clientes.forEach(cliente => { |
| const tr = document.createElement('tr'); |
| tr.innerHTML = ` |
| <td>${cliente.id}</td> |
| <td>${cliente.nombre}</td> |
| <td>${cliente.telefono || ''}</td> |
| <td> |
| <button class="btn btn-sm btn-warning" onclick="editarCliente(${cliente.id})"> |
| <i class="bi bi-pencil"></i> |
| </button> |
| <button class="btn btn-sm btn-danger" onclick="eliminarCliente(${cliente.id})"> |
| <i class="bi bi-trash"></i> |
| </button> |
| </td> |
| `; |
| tbody.appendChild(tr); |
| }); |
| }; |
| } |
|  |
| function buscarClientes() { |
| const busqueda = document.getElementById('search-cliente').value; |
| cargarClientes(busqueda); |
| } |
|  |
| function showClienteForm(cliente = null) { |
| editingId = cliente ? cliente.id : null; |
|  |
| document.getElementById('modalTitle').textContent = cliente ? 'Editar Cliente' : 'Nuevo Cliente'; |
|  |
| document.getElementById('modalBody').innerHTML = ` |
| <form onsubmit="guardarCliente(event)"> |
| <div class="mb-3"> |
| <label for="cliente-nombre" class="form-label">Nombre</label> |
| <input type="text" class="form-control" id="cliente-nombre" value="${cliente ? cliente.nombre : ''}" required> |
| </div> |
| <div class="mb-3"> |
| <label for="cliente-telefono" class="form-label">Teléfono</label> |
| <input type="text" class="form-control" id="cliente-telefono" value="${cliente ? cliente.telefono || '' : ''}"> |
| </div> |
| <div class="mb-3"> |
| <label for="cliente-nit" class="form-label">NIT</label> |
| <input type="text" class="form-control" id="cliente-nit" value="${cliente ? cliente.nit || '' : ''}"> |
| </div> |
| <div class="mb-3"> |
| <label for="cliente-direccion" class="form-label">Dirección</label> |
| <textarea class="form-control" id="cliente-direccion">${cliente ? cliente.direccion || '' : ''}</textarea> |
| </div> |
| <div class="modal-footer"> |
| <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button> |
| <button type="submit" class="btn btn-primary">Guardar</button> |
| </div> |
| </form> |
| `; |
|  |
| modal.show(); |
| } |
|  |
| function guardarCliente(event) { |
| event.preventDefault(); |
|  |
| const nombre = document.getElementById('cliente-nombre').value; |
| const telefono = document.getElementById('cliente-telefono').value; |
| const nit = document.getElementById('cliente-nit').value; |
| const direccion = document.getElementById('cliente-direccion').value; |
|  |
| if (!nombre) { |
| showToast('Error', 'El nombre es obligatorio', 'danger'); |
| return; |
| } |
|  |
| const cliente = { nombre, telefono, nit, direccion }; |
|  |
| const transaction = db.transaction(['cliente'], 'readwrite'); |
| const store = transaction.objectStore('cliente'); |
|  |
| let request; |
| if (editingId) { |
| cliente.id = editingId; |
| request = store.put(cliente); |
| } else { |
| request = store.add(cliente); |
| } |
|  |
| request.onsuccess = function() { |
| modal.hide(); |
| showToast('Éxito', `Cliente ${editingId ? 'actualizado' : 'agregado'} correctamente`, 'success'); |
| cargarClientes(); |
| updateCounts(); |
| editingId = null; |
| }; |
| } |
|  |
| function editarCliente(id) { |
| const transaction = db.transaction(['cliente'], 'readonly'); |
| const store = transaction.objectStore('cliente'); |
| const request = store.get(id); |
|  |
| request.onsuccess = function() { |
| showClienteForm(request.result); |
| }; |
| } |
|  |
| function eliminarCliente(id) { |
| if (confirm('¿Está seguro de que desea eliminar este cliente?')) { |
| const transaction = db.transaction(['cliente'], 'readwrite'); |
| const store = transaction.objectStore('cliente'); |
| const request = store.delete(id); |
|  |
| request.onsuccess = function() { |
| showToast('Éxito', 'Cliente eliminado correctamente', 'success'); |
| cargarClientes(); |
| updateCounts(); |
| }; |
| } |
| } |
|  |
| // ========== FUNCIONES PARA PROVEEDORES ========== |
| function cargarProveedores(busqueda = '') { |
| const transaction = db.transaction(['proveedor'], 'readonly'); |
| const store = transaction.objectStore('proveedor'); |
| const request = store.getAll(); |
|  |
| request.onsuccess = function() { |
| let proveedores = request.result; |
|  |
| if (busqueda) { |
| proveedores = proveedores.filter(p => |
| p.nombre.toLowerCase().includes(busqueda.toLowerCase()) |
| ); |
| } |
|  |
| const tbody = document.getElementById('proveedores-table'); |
| tbody.innerHTML = ''; |
|  |
| proveedores.forEach(proveedor => { |
| const tr = document.createElement('tr'); |
| tr.innerHTML = ` |
| <td>${proveedor.id}</td> |
| <td>${proveedor.nombre}</td> |
| <td>${proveedor.contacto || ''}</td> |
| <td> |
| <button class="btn btn-sm btn-warning" onclick="editarProveedor(${proveedor.id})"> |
| <i class="bi bi-pencil"></i> |
| </button> |
| <button class="btn btn-sm btn-danger" onclick="eliminarProveedor(${proveedor.id})"> |
| <i class="bi bi-trash"></i> |
| </button> |
| </td> |
| `; |
| tbody.appendChild(tr); |
| }); |
| }; |
| } |
|  |
| function buscarProveedores() { |
| const busqueda = document.getElementById('search-proveedor').value; |
| cargarProveedores(busqueda); |
| } |
|  |
| function showProveedorForm(proveedor = null) { |
| editingId = proveedor ? proveedor.id : null; |
|  |
| document.getElementById('modalTitle').textContent = proveedor ? 'Editar Proveedor' : 'Nuevo Proveedor'; |
|  |
| document.getElementById('modalBody').innerHTML = ` |
| <form onsubmit="guardarProveedor(event)"> |
| <div class="mb-3"> |
| <label for="proveedor-nombre" class="form-label">Nombre</label> |
| <input type="text" class="form-control" id="proveedor-nombre" value="${proveedor ? proveedor.nombre : ''}" required> |
| </div> |
| <div class="mb-3"> |
| <label for="proveedor-contacto" class="form-label">Contacto</label> |
| <input type="text" class="form-control" id="proveedor-contacto" value="${proveedor ? proveedor.contacto || '' : ''}"> |
| </div> |
| <div class="modal-footer"> |
| <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button> |
| <button type="submit" class="btn btn-primary">Guardar</button> |
| </div> |
| </form> |
| `; |
|  |
| modal.show(); |
| } |
|  |
| function guardarProveedor(event) { |
| event.preventDefault(); |
|  |
| const nombre = document.getElementById('proveedor-nombre').value; |
| const contacto = document.getElementById('proveedor-contacto').value; |
|  |
| if (!nombre) { |
| showToast('Error', 'El nombre es obligatorio', 'danger'); |
| return; |
| } |
|  |
| const proveedor = { nombre, contacto }; |
|  |
| const transaction = db.transaction(['proveedor'], 'readwrite'); |
| const store = transaction.objectStore('proveedor'); |
|  |
| let request; |
| if (editingId) { |
| proveedor.id = editingId; |
| request = store.put(proveedor); |
| } else { |
| request = store.add(proveedor); |
| } |
|  |
| request.onsuccess = function() { |
| modal.hide(); |
| showToast('Éxito', `Proveedor ${editingId ? 'actualizado' : 'agregado'} correctamente`, 'success'); |
| cargarProveedores(); |
| updateCounts(); |
| editingId = null; |
| }; |
| } |
|  |
| function editarProveedor(id) { |
| const transaction = db.transaction(['proveedor'], 'readonly'); |
| const store = transaction.objectStore('proveedor'); |
| const request = store.get(id); |
|  |
| request.onsuccess = function() { |
| showProveedorForm(request.result); |
| }; |
| } |
|  |
| function eliminarProveedor(id) { |
| if (confirm('¿Está seguro de que desea eliminar este proveedor?')) { |
| const transaction = db.transaction(['proveedor'], 'readwrite'); |
| const store = transaction.objectStore('proveedor'); |
| const request = store.delete(id); |
|  |
| request.onsuccess = function() { |
| showToast('Éxito', 'Proveedor eliminado correctamente', 'success'); |
| cargarProveedores(); |
| updateCounts(); |
| }; |
| } |
| } |
|  |
| // ========== FUNCIONES PARA EMPLEADOS ========== |
| function cargarEmpleados(busqueda = '') { |
| const transaction = db.transaction(['empleado'], 'readonly'); |
| const store = transaction.objectStore('empleado'); |
| const request = store.getAll(); |
|  |
| request.onsuccess = function() { |
| let empleados = request.result; |
|  |
| if (busqueda) { |
| empleados = empleados.filter(e => |
| e.nombre.toLowerCase().includes(busqueda.toLowerCase()) |
| ); |
| } |
|  |
| const tbody = document.getElementById('empleados-table'); |
| tbody.innerHTML = ''; |
|  |
| empleados.forEach(empleado => { |
| const tr = document.createElement('tr'); |
| tr.innerHTML = ` |
| <td>${empleado.id}</td> |
| <td>${empleado.nombre}</td> |
| <td>${empleado.cargo || ''}</td> |
| <td>$${empleado.salario ? empleado.salario.toFixed(2) : '0.00'}</td> |
| <td> |
| <button class="btn btn-sm btn-warning" onclick="editarEmpleado(${empleado.id})"> |
| <i class="bi bi-pencil"></i> |
| </button> |
| <button class="btn btn-sm btn-danger" onclick="eliminarEmpleado(${empleado.id})"> |
| <i class="bi bi-trash"></i> |
| </button> |
| </td> |
| `; |
| tbody.appendChild(tr); |
| }); |
| }; |
| } |
|  |
| function buscarEmpleados() { |
| const busqueda = document.getElementById('search-empleado').value; |
| cargarEmpleados(busqueda); |
| } |
|  |
| function showEmpleadoForm(empleado = null) { |
| editingId = empleado ? empleado.id : null; |
|  |
| document.getElementById('modalTitle').textContent = empleado ? 'Editar Empleado' : 'Nuevo Empleado'; |
|  |
| document.getElementById('modalBody').innerHTML = ` |
| <form onsubmit="guardarEmpleado(event)"> |
| <div class="mb-3"> |
| <label for="empleado-nombre" class="form-label">Nombre</label> |
| <input type="text" class="form-control" id="empleado-nombre" value="${empleado ? empleado.nombre : ''}" required> |
| </div> |
| <div class="mb-3"> |
| <label for="empleado-cargo" class="form-label">Cargo</label> |
| <input type="text" class="form-control" id="empleado-cargo" value="${empleado ? empleado.cargo || '' : ''}"> |
| </div> |
| <div class="mb-3"> |
| <label for="empleado-salario" class="form-label">Salario</label> |
| <input type="number" step="0.01" class="form-control" id="empleado-salario" value="${empleado ? empleado.salario || 0 : 0}"> |
| </div> |
| <div class="modal-footer"> |
| <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button> |
| <button type="submit" class="btn btn-primary">Guardar</button> |
| </div> |
| </form> |
| `; |
|  |
| modal.show(); |
| } |
|  |
| function guardarEmpleado(event) { |
| event.preventDefault(); |
|  |
| const nombre = document.getElementById('empleado-nombre').value; |
| const cargo = document.getElementById('empleado-cargo').value; |
| const salario = parseFloat(document.getElementById('empleado-salario').value) || 0; |
|  |
| if (!nombre) { |
| showToast('Error', 'El nombre es obligatorio', 'danger'); |
| return; |
| } |
|  |
| const empleado = { nombre, cargo, salario }; |
|  |
| const transaction = db.transaction(['empleado'], 'readwrite'); |
| const store = transaction.objectStore('empleado'); |
|  |
| let request; |
| if (editingId) { |
| empleado.id = editingId; |
| request = store.put(empleado); |
| } else { |
| request = store.add(empleado); |
| } |
|  |
| request.onsuccess = function() { |
| modal.hide(); |
| showToast('Éxito', `Empleado ${editingId ? 'actualizado' : 'agregado'} correctamente`, 'success'); |
| cargarEmpleados(); |
| updateCounts(); |
| editingId = null; |
| }; |
| } |
|  |
| function editarEmpleado(id) { |
| const transaction = db.transaction(['empleado'], 'readonly'); |
| const store = transaction.objectStore('empleado'); |
| const request = store.get(id); |
|  |
| request.onsuccess = function() { |
| showEmpleadoForm(request.result); |
| }; |
| } |
|  |
| function eliminarEmpleado(id) { |
| if (confirm('¿Está seguro de que desea eliminar este empleado?')) { |
| const transaction = db.transaction(['empleado'], 'readwrite'); |
| const store = transaction.objectStore('empleado'); |
| const request = store.delete(id); |
|  |
| request.onsuccess = function() { |
| showToast('Éxito', 'Empleado eliminado correctamente', 'success'); |
| cargarEmpleados(); |
| updateCounts(); |
| }; |
| } |
| } |
|  |
| // ========== FUNCIONES PARA VENTAS ========== |
| function cargarVentas(busqueda = '') { |
| const transaction = db.transaction(['venta', 'cliente'], 'readonly'); |
| const ventaStore = transaction.objectStore('venta'); |
| const clienteStore = transaction.objectStore('cliente'); |
| const request = ventaStore.getAll(); |
|  |
| request.onsuccess = function() { |
| let ventas = request.result; |
|  |
| if (busqueda) { |
| ventas = ventas.filter(v => |
| (v.cliente\_nombre && v.cliente\_nombre.toLowerCase().includes(busqueda.toLowerCase())) || |
| (v.cliente && v.cliente.toLowerCase().includes(busqueda.toLowerCase())) |
| ); |
| } |
|  |
| const tbody = document.getElementById('ventas-table'); |
| tbody.innerHTML = ''; |
|  |
| // Obtener nombres de clientes para cada venta |
| const promises = ventas.map(venta => { |
| return new Promise(resolve => { |
| if (venta.cliente\_id) { |
| const clienteRequest = clienteStore.get(venta.cliente\_id); |
| clienteRequest.onsuccess = function() { |
| venta.cliente\_nombre = clienteRequest.result ? clienteRequest.result.nombre : 'Cliente no encontrado'; |
| resolve(venta); |
| }; |
| clienteRequest.onerror = function() { |
| venta.cliente\_nombre = venta.cliente || 'Cliente no especificado'; |
| resolve(venta); |
| }; |
| } else { |
| venta.cliente\_nombre = venta.cliente || 'Cliente no especificado'; |
| resolve(venta); |
| } |
| }); |
| }); |
|  |
| Promise.all(promises).then(ventasConNombres => { |
| ventasConNombres.forEach(venta => { |
| const tr = document.createElement('tr'); |
| tr.innerHTML = ` |
| <td>${venta.id}</td> |
| <td>${venta.fecha}</td> |
| <td>${venta.cliente\_nombre}</td> |
| <td>${venta.nit || ''}</td> |
| <td>$${venta.total.toFixed(2)}</td> |
| <td> |
| <button class="btn btn-sm btn-info" onclick="verDetalleVenta(${venta.id})"> |
| <i class="bi bi-eye"></i> |
| </button> |
| <button class="btn btn-sm btn-success" onclick="imprimirRecibo(${venta.id})"> |
| <i class="bi bi-printer"></i> |
| </button> |
| <button class="btn btn-sm btn-danger" onclick="eliminarVenta(${venta.id})"> |
| <i class="bi bi-trash"></i> |
| </button> |
| </td> |
| `; |
| tbody.appendChild(tr); |
| }); |
| }); |
| }; |
| } |
|  |
| function buscarVentas() { |
| const busqueda = document.getElementById('search-venta').value; |
| cargarVentas(busqueda); |
| } |
|  |
| function showVentaForm(venta = null) { |
| editingId = venta ? venta.id : null; |
|  |
| document.getElementById('modalTitle').textContent = venta ? 'Editar Venta' : 'Nueva Venta'; |
|  |
| // Cargar clientes para el selector |
| const transaction = db.transaction(['cliente'], 'readonly'); |
| const store = transaction.objectStore('cliente'); |
| const request = store.getAll(); |
|  |
| request.onsuccess = function() { |
| const clientes = request.result; |
| let clienteOptions = '<option value="">Seleccionar cliente...</option>'; |
| clientes.forEach(cliente => { |
| const selected = venta && venta.cliente\_id === cliente.id ? 'selected' : ''; |
| clienteOptions += `<option value="${cliente.id}" ${selected} data-nit="${cliente.nit || ''}" data-direccion="${cliente.direccion || ''}" data-telefono="${cliente.telefono || ''}">${cliente.nombre}</option>`; |
| }); |
|  |
| // Cargar productos para el selector |
| const productoTransaction = db.transaction(['producto'], 'readonly'); |
| const productoStore = productoTransaction.objectStore('producto'); |
| const productoRequest = productoStore.getAll(); |
|  |
| productoRequest.onsuccess = function() { |
| const productos = productoRequest.result; |
| let productoOptions = '<option value="">Seleccionar producto...</option>'; |
| productos.forEach(producto => { |
| productoOptions += `<option value="${producto.id}" data-precio="${producto.precio\_unitario}" data-codigo="${producto.codigo\_barras || ''}">${producto.nombre} - $${producto.precio\_unitario.toFixed(2)}</option>`; |
| }); |
|  |
| // Construir el formulario |
| let productosHTML = ''; |
| if (venta && venta.productos) { |
| venta.productos.forEach((prod, index) => { |
| productosHTML += ` |
| <div class="row producto-item mb-2" data-id="${prod.id || ''}"> |
| <div class="col-md-5"> |
| <select class="form-select producto-select" onchange="actualizarPrecio(this)" required> |
| <option value="">Seleccionar producto...</option> |
| ${productoOptions.replace(`value="${prod.id}"`, `value="${prod.id}" selected`)} |
| </select> |
| </div> |
| <div class="col-md-2"> |
| <input type="number" class="form-control cantidad" min="1" value="${prod.cantidad || 1}" onchange="calcularSubtotal(this)" required> |
| </div> |
| <div class="col-md-2"> |
| <input type="number" step="0.01" class="form-control precio" value="${prod.precio || 0}" readonly> |
| </div> |
| <div class="col-md-2"> |
| <input type="number" step="0.01" class="form-control subtotal" value="${prod.subtotal || 0}" readonly> |
| </div> |
| <div class="col-md-1"> |
| <button type="button" class="btn btn-danger btn-sm" onclick="eliminarProductoVenta(this)">X</button> |
| </div> |
| </div> |
| `; |
| }); |
| } else { |
| productosHTML = ` |
| <div class="row producto-item mb-2"> |
| <div class="col-md-5"> |
| <select class="form-select producto-select" onchange="actualizarPrecio(this)" required> |
| ${productoOptions} |
| </select> |
| </div> |
| <div class="col-md-2"> |
| <input type="number" class="form-control cantidad" min="1" value="1" onchange="calcularSubtotal(this)" required> |
| </div> |
| <div class="col-md-2"> |
| <input type="number" step="0.01" class="form-control precio" value="0" readonly> |
| </div> |
| <div class="col-md-2"> |
| <input type="number" step="0.01" class="form-control subtotal" value="0" readonly> |
| </div> |
| <div class="col-md-1"> |
| <button type="button" class="btn btn-danger btn-sm" onclick="eliminarProductoVenta(this)">X</button> |
| </div> |
| </div> |
| `; |
| } |
|  |
| document.getElementById('modalBody').innerHTML = ` |
| <form onsubmit="guardarVenta(event)"> |
| <div class="mb-3"> |
| <label for="venta-fecha" class="form-label">Fecha</label> |
| <input type="date" class="form-control" id="venta-fecha" value="${venta ? venta.fecha : new Date().toISOString().split('T')[0]}" required> |
| </div> |
| <div class="mb-3"> |
| <label for="venta-cliente" class="form-label">Cliente</label> |
| <select class="form-select" id="venta-cliente" onchange="actualizarDatosCliente()"> |
| ${clienteOptions} |
| </select> |
| </div> |
| <div class="mb-3"> |
| <label for="venta-nit" class="form-label">NIT</label> |
| <input type="text" class="form-control" id="venta-nit" value="${venta ? venta.nit || '' : ''}"> |
| </div> |
| <div class="mb-3"> |
| <label for="venta-direccion" class="form-label">Dirección</label> |
| <input type="text" class="form-control" id="venta-direccion" value="${venta ? venta.direccion || '' : ''}"> |
| </div> |
| <div class="mb-3"> |
| <label for="venta-telefono" class="form-label">Teléfono</label> |
| <input type="text" class="form-control" id="venta-telefono" value="${venta ? venta.telefono || '' : ''}"> |
| </div> |
|  |
| <h5>Productos |
| <button type="button" class="btn btn-sm btn-outline-primary ms-2" onclick="abrirLectorCodigoBarras()"> |
| <i class="bi bi-upc-scan"></i> Escanear código |
| </button> |
| </h5> |
| <div id="productos-container"> |
| ${productosHTML} |
| </div> |
|  |
| <button type="button" class="btn btn-secondary btn-sm mt-2 mb-3" onclick="agregarProductoVenta()"> |
| <i class="bi bi-plus"></i> Agregar Producto |
| </button> |
|  |
| <div class="row mt-3"> |
| <div class="col-md-6"> |
| <div class="form-group"> |
| <label for="venta-subtotal" class="form-label">Subtotal</label> |
| <input type="number" step="0.01" class="form-control" id="venta-subtotal" value="${venta ? venta.subtotal || 0 : 0}" readonly> |
| </div> |
| </div> |
| <div class="col-md-6"> |
| <div class="form-group"> |
| <label for="venta-total" class="form-label">Total</label> |
| <input type="number" step="0.01" class="form-control" id="venta-total" value="${venta ? venta.total || 0 : 0}" readonly> |
| </div> |
| </div> |
| </div> |
|  |
| <div class="modal-footer"> |
| <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button> |
| <button type="submit" class="btn btn-primary">Guardar</button> |
| </div> |
| </form> |
| `; |
|  |
| modal.show(); |
| calcularTotalVenta(); |
|  |
| // Si estamos editando, actualizar precios de productos existentes |
| if (venta && venta.productos) { |
| setTimeout(() => { |
| document.querySelectorAll('.producto-select').forEach(select => { |
| if (select.value) { |
| actualizarPrecio(select); |
| } |
| }); |
| }, 100); |
| } |
| }; |
| }; |
| } |
|  |
| function actualizarDatosCliente() { |
| const clienteSelect = document.getElementById('venta-cliente'); |
| const selectedOption = clienteSelect.options[clienteSelect.selectedIndex]; |
|  |
| if (selectedOption && selectedOption.value) { |
| document.getElementById('venta-nit').value = selectedOption.getAttribute('data-nit') || ''; |
| document.getElementById('venta-direccion').value = selectedOption.getAttribute('data-direccion') || ''; |
| document.getElementById('venta-telefono').value = selectedOption.getAttribute('data-telefono') || ''; |
| } |
| } |
|  |
| function abrirLectorCodigoBarras() { |
| barcodeModal.show(); |
| } |
|  |
| function startBarcodeScanner() { |
| if (!quaggaInitialized) { |
| Quagga.init({ |
| inputStream: { |
| name: "Live", |
| type: "LiveStream", |
| target: document.querySelector('#barcode-video'), |
| constraints: { |
| width: 400, |
| height: 300, |
| facingMode: "environment" // Usar la cámara trasera |
| } |
| }, |
| decoder: { |
| readers: ["code\_128\_reader", "ean\_reader", "ean\_8\_reader", "code\_39\_reader", "upc\_reader"] |
| } |
| }, function(err) { |
| if (err) { |
| console.error(err); |
| showToast('Error', 'No se pudo inicializar el escáner', 'danger'); |
| return; |
| } |
| quaggaInitialized = true; |
| Quagga.start(); |
| document.getElementById('start-scan').style.display = 'none'; |
| document.getElementById('stop-scan').style.display = 'inline-block'; |
| }); |
|  |
| Quagga.onDetected(function(result) { |
| const code = result.codeResult.code; |
| document.getElementById('manual-barcode').value = code; |
| addProductByBarcode(code); |
| stopBarcodeScanner(); |
| barcodeModal.hide(); |
| }); |
| } else { |
| Quagga.start(); |
| document.getElementById('start-scan').style.display = 'none'; |
| document.getElementById('stop-scan').style.display = 'inline-block'; |
| } |
| } |
|  |
| function stopBarcodeScanner() { |
| if (quaggaInitialized) { |
| Quagga.stop(); |
| document.getElementById('start-scan').style.display = 'inline-block'; |
| document.getElementById('stop-scan').style.display = 'none'; |
| } |
| } |
|  |
| function addProductByBarcode(barcode) { |
| const transaction = db.transaction(['producto'], 'readonly'); |
| const store = transaction.objectStore('producto'); |
| const index = store.index('codigo\_barras'); |
| const request = index.get(barcode); |
|  |
| request.onsuccess = function() { |
| const producto = request.result; |
| if (producto) { |
| // Buscar si el producto ya está en la lista |
| let encontrado = false; |
| document.querySelectorAll('.producto-select').forEach(select => { |
| if (select.value == producto.id) { |
| // Incrementar la cantidad |
| const cantidadInput = select.closest('.producto-item').querySelector('.cantidad'); |
| cantidadInput.value = parseInt(cantidadInput.value) + 1; |
| calcularSubtotal(cantidadInput); |
| encontrado = true; |
| } |
| }); |
|  |
| if (!encontrado) { |
| // Agregar nuevo producto |
| agregarProductoVenta(); |
| const lastProduct = document.querySelectorAll('.producto-item'); |
| const lastSelect = lastProduct[lastProduct.length - 1].querySelector('.producto-select'); |
| lastSelect.value = producto.id; |
| actualizarPrecio(lastSelect); |
| } |
| } else { |
| showToast('Error', 'Producto no encontrado', 'danger'); |
| } |
| }; |
| } |
|  |
| function addProductByManualBarcode() { |
| const barcode = document.getElementById('manual-barcode').value; |
| if (barcode) { |
| addProductByBarcode(barcode); |
| } |
| } |
|  |
| function agregarProductoVenta() { |
| const productoContainer = document.getElementById('productos-container'); |
| const nuevoProducto = document.createElement('div'); |
| nuevoProducto.className = 'row producto-item mb-2'; |
| nuevoProducto.innerHTML = ` |
| <div class="col-md-5"> |
| <select class="form-select producto-select" onchange="actualizarPrecio(this)" required> |
| <option value="">Seleccionar producto...</option> |
| ${document.querySelector('.producto-select').innerHTML.replace(/<option value="">Seleccionar producto...<\/option>/, '')} |
| </select> |
| </div> |
| <div class="col-md-2"> |
| <input type="number" class="form-control cantidad" min="1" value="1" onchange="calcularSubtotal(this)" required> |
| </div> |
| <div class="col-md-2"> |
| <input type="number" step="0.01" class="form-control precio" value="0" readonly> |
| </div> |
| <div class="col-md-2"> |
| <input type="number" step="0.01" class="form-control subtotal" value="0" readonly> |
| </div> |
| <div class="col-md-1"> |
| <button type="button" class="btn btn-danger btn-sm" onclick="eliminarProductoVenta(this)">X</button> |
| </div> |
| `; |
| productoContainer.appendChild(nuevoProducto); |
| } |
|  |
| function eliminarProductoVenta(button) { |
| // No permitir eliminar si solo queda un producto |
| if (document.querySelectorAll('.producto-item').length > 1) { |
| button.closest('.producto-item').remove(); |
| calcularTotalVenta(); |
| } else { |
| showToast('Advertencia', 'Debe haber al menos un producto en la venta', 'warning'); |
| } |
| } |
|  |
| function actualizarPrecio(select) { |
| const precioInput = select.closest('.producto-item').querySelector('.precio'); |
| const selectedOption = select.options[select.selectedIndex]; |
| const precio = selectedOption.getAttribute('data-precio') || 0; |
| precioInput.value = precio; |
| calcularSubtotal(select); |
| } |
|  |
| function calcularSubtotal(input) { |
| const item = input.closest('.producto-item'); |
| const cantidad = item.querySelector('.cantidad').value || 0; |
| const precio = item.querySelector('.precio').value || 0; |
| const subtotal = cantidad \* precio; |
| item.querySelector('.subtotal').value = subtotal.toFixed(2); |
| calcularTotalVenta(); |
| } |
|  |
| function calcularTotalVenta() { |
| let subtotal = 0; |
| document.querySelectorAll('.subtotal').forEach(input => { |
| subtotal += parseFloat(input.value) || 0; |
| }); |
|  |
| document.getElementById('venta-subtotal').value = subtotal.toFixed(2); |
| document.getElementById('venta-total').value = subtotal.toFixed(2); // Podría agregar impuestos aquí |
| } |
|  |
| function guardarVenta(event) { |
| event.preventDefault(); |
|  |
| const fecha = document.getElementById('venta-fecha').value; |
| const cliente\_id = document.getElementById('venta-cliente').value; |
| const nit = document.getElementById('venta-nit').value; |
| const direccion = document.getElementById('venta-direccion').value; |
| const telefono = document.getElementById('venta-telefono').value; |
| const subtotal = parseFloat(document.getElementById('venta-subtotal').value); |
| const total = parseFloat(document.getElementById('venta-total').value); |
|  |
| // Obtener productos |
| const productos = []; |
| document.querySelectorAll('.producto-item').forEach(item => { |
| const productoSelect = item.querySelector('.producto-select'); |
| const producto\_id = productoSelect.value; |
| const producto\_nombre = productoSelect.options[productoSelect.selectedIndex].text; |
| const cantidad = parseInt(item.querySelector('.cantidad').value); |
| const precio = parseFloat(item.querySelector('.precio').value); |
| const producto\_subtotal = parseFloat(item.querySelector('.subtotal').value); |
|  |
| if (producto\_id && cantidad && precio) { |
| productos.push({ |
| id: producto\_id, |
| nombre: producto\_nombre, |
| cantidad: cantidad, |
| precio: precio, |
| subtotal: producto\_subtotal |
| }); |
| } |
| }); |
|  |
| if (productos.length === 0) { |
| showToast('Error', 'Debe agregar al menos un producto', 'danger'); |
| return; |
| } |
|  |
| // Obtener nombre del cliente si está seleccionado |
| let cliente\_nombre = ''; |
| if (cliente\_id) { |
| const clienteSelect = document.getElementById('venta-cliente'); |
| cliente\_nombre = clienteSelect.options[clienteSelect.selectedIndex].text; |
| } |
|  |
| const venta = { |
| fecha: fecha, |
| cliente\_id: cliente\_id || null, |
| cliente: cliente\_nombre, |
| nit: nit, |
| direccion: direccion, |
| telefono: telefono, |
| productos: productos, |
| subtotal: subtotal, |
| total: total |
| }; |
|  |
| const transaction = db.transaction(['venta'], 'readwrite'); |
| const store = transaction.objectStore('venta'); |
|  |
| let request; |
| if (editingId) { |
| venta.id = editingId; |
| request = store.put(venta); |
| } else { |
| request = store.add(venta); |
| } |
|  |
| request.onsuccess = function() { |
| modal.hide(); |
| showToast('Éxito', `Venta ${editingId ? 'actualizada' : 'registrada'} correctamente`, 'success'); |
| cargarVentas(); |
| updateCounts(); |
| editingId = null; |
|  |
| // Imprimir recibo automáticamente para nuevas ventas |
| if (!editingId) { |
| imprimirRecibo(request.result); // Pasar el ID de la venta recién creada |
| } |
| }; |
|  |
| request.onerror = function() { |
| showToast('Error', 'No se pudo guardar la venta', 'danger'); |
| }; |
| } |
|  |
| function verDetalleVenta(id) { |
| const transaction = db.transaction(['venta', 'cliente'], 'readonly'); |
| const ventaStore = transaction.objectStore('venta'); |
| const clienteStore = transaction.objectStore('cliente'); |
| const request = ventaStore.get(id); |
|  |
| request.onsuccess = function() { |
| const venta = request.result; |
|  |
| // Obtener nombre del cliente si existe |
| let clienteNombre = venta.cliente || 'Cliente no especificado'; |
| if (venta.cliente\_id) { |
| const clienteRequest = clienteStore.get(venta.cliente\_id); |
| clienteRequest.onsuccess = function() { |
| if (clienteRequest.result) { |
| clienteNombre = clienteRequest.result.nombre; |
| } |
| mostrarDetalleVenta(venta, clienteNombre); |
| }; |
| clienteRequest.onerror = function() { |
| clienteNombre = venta.cliente || 'Cliente no especificado'; |
| mostrarDetalleVenta(venta, clienteNombre); |
| }; |
| } else { |
| mostrarDetalleVenta(venta, clienteNombre); |
| } |
| }; |
| } |
|  |
| function mostrarDetalleVenta(venta, clienteNombre) { |
| let productosHTML = ''; |
| if (venta.productos && venta.productos.length > 0) { |
| venta.productos.forEach(prod => { |
| productosHTML += ` |
| <tr> |
| <td>${prod.nombre}</td> |
| <td>${prod.cantidad}</td> |
| <td>$${prod.precio.toFixed(2)}</td> |
| <td>$${prod.subtotal.toFixed(2)}</td> |
| </tr> |
| `; |
| }); |
| } |
|  |
| document.getElementById('modalTitle').textContent = `Detalle de Venta #${venta.id}`; |
| document.getElementById('modalBody').innerHTML = ` |
| <div class="mb-3"> |
| <strong>Fecha:</strong> ${venta.fecha} |
| </div> |
| <div class="mb-3"> |
| <strong>Cliente:</strong> ${clienteNombre} |
| </div> |
| <div class="mb-3"> |
| <strong>NIT:</strong> ${venta.nit || 'No especificado'} |
| </div> |
| <div class="mb-3"> |
| <strong>Dirección:</strong> ${venta.direccion || 'No especificada'} |
| </div> |
| <div class="mb-3"> |
| <strong>Teléfono:</strong> ${venta.telefono || 'No especificado'} |
| </div> |
|  |
| <h5>Productos</h5> |
| <table class="table table-sm"> |
| <thead> |
| <tr> |
| <th>Producto</th> |
| <th>Cantidad</th> |
| <th>Precio Unitario</th> |
| <th>Subtotal</th> |
| </tr> |
| </thead> |
| <tbody> |
| ${productosHTML} |
| </tbody> |
| </table> |
|  |
| <div class="row mt-3"> |
| <div class="col-md-6"> |
| <strong>Subtotal:</strong> $${venta.subtotal.toFixed(2)} |
| </div> |
| <div class="col-md-6"> |
| <strong>Total:</strong> $${venta.total.toFixed(2)} |
| </div> |
| </div> |
|  |
| <div class="modal-footer"> |
| <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button> |
| <button type="button" class="btn btn-primary" onclick="imprimirRecibo(${venta.id})"> |
| <i class="bi bi-printer"></i> Imprimir Recibo |
| </button> |
| </div> |
| `; |
|  |
| modal.show(); |
| } |
|  |
| function imprimirRecibo(ventaId) { |
| const transaction = db.transaction(['venta', 'cliente'], 'readonly'); |
| const ventaStore = transaction.objectStore('venta'); |
| const clienteStore = transaction.objectStore('cliente'); |
| const request = ventaStore.get(ventaId); |
|  |
| request.onsuccess = function() { |
| const venta = request.result; |
|  |
| // Obtener nombre del cliente si existe |
| let clienteNombre = venta.cliente || 'Cliente no especificado'; |
| if (venta.cliente\_id) { |
| const clienteRequest = clienteStore.get(venta.cliente\_id); |
| clienteRequest.onsuccess = function() { |
| if (clienteRequest.result) { |
| clienteNombre = clienteRequest.result.nombre; |
| } |
| generarRecibo(venta, clienteNombre); |
| }; |
| clienteRequest.onerror = function() { |
| clienteNombre = venta.cliente || 'Cliente no especificado'; |
| generarRecibo(venta, clienteNombre); |
| }; |
| } else { |
| generarRecibo(venta, clienteNombre); |
| } |
| }; |
| } |
|  |
| function generarRecibo(venta, clienteNombre) { |
| // Actualizar los datos del recibo |
| document.getElementById('recibo-numero').textContent = venta.id; |
| document.getElementById('recibo-fecha').textContent = venta.fecha; |
| document.getElementById('recibo-cliente').textContent = clienteNombre; |
| document.getElementById('recibo-nit').textContent = venta.nit || 'CF'; |
| document.getElementById('recibo-total').textContent = `$${venta.total.toFixed(2)}`; |
|  |
| // Generar items del recibo |
| const itemsContainer = document.getElementById('recibo-items'); |
| itemsContainer.innerHTML = ''; |
|  |
| if (venta.productos && venta.productos.length > 0) { |
| venta.productos.forEach(prod => { |
| const tr = document.createElement('tr'); |
| tr.innerHTML = ` |
| <td>${prod.nombre}</td> |
| <td>${prod.cantidad}</td> |
| <td style="text-align: right;">$${prod.precio.toFixed(2)}</td> |
| `; |
| itemsContainer.appendChild(tr); |
| }); |
| } |
|  |
| // Crear una ventana de impresión |
| const ventanaImpresion = window.open('', '\_blank'); |
| ventanaImpresion.document.write(` |
| <!DOCTYPE html> |
| <html> |
| <head> |
| <title>Recibo de Venta - Farmacia El Ahorro</title> |
| <style> |
| body { |
| font-family: 'Courier New', monospace; |
| width: 80mm; |
| margin: 0 auto; |
| padding: 10px; |
| font-size: 12px; |
| } |
| .receipt-header, .receipt-footer { |
| text-align: center; |
| margin-bottom: 10px; |
| } |
| .receipt-items { |
| width: 100%; |
| border-collapse: collapse; |
| margin: 10px 0; |
| } |
| .receipt-items th, .receipt-items td { |
| padding: 4px 0; |
| border-bottom: 1px dashed #ccc; |
| } |
| .receipt-total { |
| margin-top: 10px; |
| border-top: 2px solid #000; |
| padding-top: 10px; |
| font-weight: bold; |
| } |
| @media print { |
| body { |
| width: 80mm; |
| margin: 0; |
| padding: 10px; |
| } |
| } |
| </style> |
| </head> |
| <body onload="window.print(); window.close();"> |
| <div class="receipt-header"> |
| <h4>Farmacia El Ahorro</h4> |
| <p>Tel: (502) 1234-5678</p> |
| <p>Dirección: Av. Principal 123</p> |
| <p>NIT: 1234567-8</p> |
| </div> |
| <hr> |
| <div class="receipt-details"> |
| <p><strong>Recibo No:</strong> ${venta.id}</p> |
| <p><strong>Fecha:</strong> ${venta.fecha}</p> |
| <p><strong>Cliente:</strong> ${clienteNombre}</p> |
| <p><strong>NIT:</strong> ${venta.nit || 'CF'}</p> |
| </div> |
| <hr> |
| <table class="receipt-items"> |
| <thead> |
| <tr> |
| <th style="width: 60%;">Producto</th> |
| <th style="width: 15%;">Cant</th> |
| <th style="width: 25%; text-align: right;">Precio</th> |
| </tr> |
| </thead> |
| <tbody> |
| ${venta.productos.map(prod => ` |
| <tr> |
| <td>${prod.nombre}</td> |
| <td>${prod.cantidad}</td> |
| <td style="text-align: right;">$${prod.precio.toFixed(2)}</td> |
| </tr> |
| `).join('')} |
| </tbody> |
| </table> |
| <hr> |
| <div class="receipt-total"> |
| <p style="text-align: right;">Total: $${venta.total.toFixed(2)}</p> |
| </div> |
| <div class="receipt-footer"> |
| <p>¡Gracias por su compra!</p> |
| <p>Vuelva pronto</p> |
| </div> |
| </body> |
| </html> |
| `); |
| ventanaImpresion.document.close(); |
|  |
| showToast('Éxito', 'Recibo impreso correctamente', 'success'); |
| } |
|  |
| function eliminarVenta(id) { |
| if (confirm('¿Está seguro de que desea eliminar esta venta?')) { |
| const transaction = db.transaction(['venta'], 'readwrite'); |
| const store = transaction.objectStore('venta'); |
| const request = store.delete(id); |
|  |
| request.onsuccess = function() { |
| showToast('Éxito', 'Venta eliminada correctamente', 'success'); |
| cargarVentas(); |
| updateCounts(); |
| }; |
| } |
| } |
| </script> |
| </body> |
| </html> |