html

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Gestor de Inventario con Lector</title>

<style>

:root {

--primary: #3498db;

--secondary: #2ecc71;

--danger: #e74c3c;

--warning: #f39c12;

--dark: #2c3e50;

--light: #ecf0f1;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

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

}

body {

background-color: #f5f7fa;

color: var(--dark);

line-height: 1.6;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

header {

background: linear-gradient(135deg, var(--dark), #1a2530);

color: white;

padding: 25px 0;

margin-bottom: 30px;

box-shadow: 0 4px 12px rgba(0,0,0,0.1);

border-radius: 0 0 10px 10px;

}

.header-content {

display: flex;

justify-content: space-between;

align-items: center;

}

h1 {

font-size: 2.2rem;

display: flex;

align-items: center;

gap: 10px;

}

h1 i {

color: var(--secondary);

}

.card {

background: white;

border-radius: 10px;

box-shadow: 0 4px 15px rgba(0,0,0,0.08);

padding: 25px;

margin-bottom: 25px;

transition: transform 0.3s, box-shadow 0.3s;

}

.card:hover {

transform: translateY(-5px);

box-shadow: 0 6px 20px rgba(0,0,0,0.12);

}

.form-group {

margin-bottom: 20px;

}

label {

display: block;

margin-bottom: 8px;

font-weight: 600;

color: var(--dark);

}

input, select {

width: 100%;

padding: 12px 15px;

border: 1px solid #ddd;

border-radius: 6px;

font-size: 16px;

transition: border 0.3s, box-shadow 0.3s;

}

input:focus, select:focus {

border-color: var(--primary);

box-shadow: 0 0 0 3px rgba(52, 152, 219, 0.2);

outline: none;

}

button {

background-color: var(--primary);

color: white;

border: none;

padding: 12px 20px;

border-radius: 6px;

cursor: pointer;

font-size: 16px;

font-weight: 600;

transition: all 0.3s;

display: inline-flex;

align-items: center;

gap: 8px;

}

button:hover {

background-color: #2980b9;

transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

}

.btn-success {

background-color: var(--secondary);

}

.btn-success:hover {

background-color: #27ae60;

}

.btn-danger {

background-color: var(--danger);

}

.btn-danger:hover {

background-color: #c0392b;

}

.btn-warning {

background-color: var(--warning);

}

.btn-warning:hover {

background-color: #e67e22;

}

.actions {

display: flex;

gap: 12px;

flex-wrap: wrap;

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

box-shadow: 0 2px 8px rgba(0,0,0,0.05);

border-radius: 8px;

overflow: hidden;

}

th, td {

padding: 15px;

text-align: left;

border-bottom: 1px solid #eaeaea;

}

th {

background-color: #f8f9fa;

font-weight: 700;

color: var(--dark);

}

tr:hover {

background-color: #f9f9f9;

}

.low-stock {

color: var(--danger);

font-weight: bold;

background-color: rgba(231, 76, 60, 0.1);

padding: 5px 10px;

border-radius: 20px;

display: inline-block;

}

.good-stock {

color: var(--secondary);

font-weight: bold;

background-color: rgba(46, 204, 113, 0.1);

padding: 5px 10px;

border-radius: 20px;

display: inline-block;

}

.alert {

padding: 15px;

border-radius: 8px;

margin-bottom: 20px;

display: none;

animation: fadeIn 0.5s;

}

.alert-success {

background-color: #d4edda;

color: #155724;

border: 1px solid #c3e6cb;

}

.alert-error {

background-color: #f8d7da;

color: #721c24;

border: 1px solid #f5c6cb;

}

.stats-container {

display: grid;

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

gap: 20px;

margin-bottom: 30px;

}

.stat-card {

background: white;

border-radius: 10px;

padding: 20px;

text-align: center;

box-shadow: 0 4px 10px rgba(0,0,0,0.08);

}

.stat-value {

font-size: 2.5rem;

font-weight: 700;

margin: 10px 0;

}

.stat-label {

color: #7f8c8d;

font-weight: 600;

}

.total-value {

color: var(--primary);

}

.low-stock-count {

color: var(--danger);

}

.categories-count {

color: var(--secondary);

}

.scanner-container {

position: relative;

margin-bottom: 20px;

border-radius: 8px;

overflow: hidden;

background: #f8f9fa;

padding: 20px;

text-align: center;

}

#reader {

width: 100%;

max-width: 500px;

margin: 0 auto;

border-radius: 8px;

overflow: hidden;

}

.scanner-controls {

display: flex;

justify-content: center;

gap: 10px;

margin-top: 15px;

}

.scanner-overlay {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

background: rgba(0,0,0,0.7);

color: white;

z-index: 10;

}

.input-methods {

display: flex;

gap: 10px;

margin-bottom: 20px;

}

.input-method-btn {

flex: 1;

text-align: center;

padding: 15px;

background: #f8f9fa;

border-radius: 8px;

cursor: pointer;

transition: all 0.3s;

border: 2px solid transparent;

}

.input-method-btn.active {

border-color: var(--primary);

background: rgba(52, 152, 219, 0.1);

}

.input-method-btn:hover {

background: #e9ecef;

}

.input-method-btn i {

font-size: 24px;

margin-bottom: 8px;

display: block;

}

.manual-input {

display: block;

}

.scanner-input {

display: none;

}

.search-container {

display: flex;

gap: 10px;

margin-bottom: 20px;

}

.search-input {

flex: 1;

}

.empty-state {

text-align: center;

padding: 40px 20px;

color: #7f8c8d;

}

.empty-state i {

font-size: 3rem;

margin-bottom: 15px;

color: #bdc3c7;

}

@keyframes fadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

@media (max-width: 768px) {

.header-content {

flex-direction: column;

text-align: center;

gap: 15px;

}

table {

display: block;

overflow-x: auto;

}

.actions {

flex-direction: column;

}

button {

width: 100%;

justify-content: center;

}

.input-methods {

flex-direction: column;

}

}

</style>

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

<script src="https://cdn.jsdelivr.net/npm/html5-qrcode@2.3.4/minified/html5-qrcode.min.js"></script>

</head>

<body>

<header>

<div class="container">

<div class="header-content">

<h1><i class="fas fa-boxes"></i> Gestor de Inventario</h1>

<div class="total-products">

<span id="total-count">0</span> productos en inventario

</div>

</div>

</div>

</header>

<div class="container">

<div class="alert" id="alert-message"></div>

<div class="stats-container">

<div class="stat-card">

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

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

</div>

<div class="stat-card">

<div class="stat-label">Productos</div>

<div class="stat-value" id="products-count">0</div>

</div>

<div class="stat-card">

<div class="stat-label">Stock Bajo</div>

<div class="stat-value low-stock-count" id="low-stock-count">0</div>

</div>

<div class="stat-card">

<div class="stat-label">Categorías</div>

<div class="stat-value categories-count" id="categories-count">0</div>

</div>

</div>

<div class="card">

<h2><i class="fas fa-plus-circle"></i> Agregar Nuevo Producto</h2>

<div class="input-methods">

<div class="input-method-btn active" data-method="manual">

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

<span>Entrada Manual</span>

</div>

<div class="input-method-btn" data-method="scanner">

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

<span>Escanear Código</span>

</div>

</div>

<div class="manual-input" id="manual-input">

<form id="product-form">

<div class="form-group">

<label for="product-code"><i class="fas fa-barcode"></i> Código del Producto</label>

<input type="text" id="product-code" required placeholder="Ej: 7501006554102 o código QR">

</div>

<div class="form-group">

<label for="product-name"><i class="fas fa-tag"></i> Nombre del Producto</label>

<input type="text" id="product-name" required placeholder="Ej: Arroz Integral">

</div>

<div class="form-group">

<label for="product-category"><i class="fas fa-folder"></i> Categoría</label>

<input type="text" id="product-category" required placeholder="Ej: Alimentos">

</div>

<div class="form-group">

<label for="product-price"><i class="fas fa-dollar-sign"></i> Precio ($)</label>

<input type="number" id="product-price" step="0.01" min="0" required placeholder="0.00">

</div>

<div class="form-group">

<label for="product-quantity"><i class="fas fa-boxes"></i> Cantidad</label>

<input type="number" id="product-quantity" min="0" required placeholder="0">

</div>

<div class="form-group">

<label for="product-minimum"><i class="fas fa-exclamation-triangle"></i> Stock Mínimo</label>

<input type="number" id="product-minimum" min="0" required placeholder="0">

</div>

<div class="actions">

<button type="submit" class="btn-success"><i class="fas fa-save"></i> Agregar Producto</button>

<button type="button" id="clear-form"><i class="fas fa-broom"></i> Limpiar</button>

</div>

</form>

</div>

<div class="scanner-input" id="scanner-input">

<div class="scanner-container">

<div id="reader"></div>

<div class="scanner-controls">

<button id="start-scanner" class="btn-success"><i class="fas fa-camera"></i> Iniciar Escáner</button>

<button id="stop-scanner" class="btn-danger" disabled><i class="fas fa-stop"></i> Detener Escáner</button>

</div>

</div>

<div class="scanned-result">

<h3>Datos Escaneados</h3>

<div class="form-group">

<label for="scanned-code">Código Escaneado</label>

<input type="text" id="scanned-code" readonly>

</div>

<button id="use-scanned-code" class="btn-success"><i class="fas fa-check"></i> Usar Código Escaneado</button>

</div>

</div>

</div>

<div class="card">

<h2><i class="fas fa-clipboard-list"></i> Inventario Actual</h2>

<div class="search-container">

<div class="search-input">

<input type="text" id="search-input" placeholder="Buscar productos por nombre, categoría o código...">

</div>

<button id="export-btn" class="btn-warning"><i class="fas fa-file-export"></i> Exportar</button>

</div>

<table id="inventory-table">

<thead>

<tr>

<th>Código</th>

<th>Nombre</th>

<th>Categoría</th>

<th>Precio</th>

<th>Cantidad</th>

<th>Stock Mínimo</th>

<th>Estado</th>

<th>Acciones</th>

</tr>

</thead>

<tbody id="inventory-body">

<!-- Los productos se cargarán aquí dinámicamente -->

</tbody>

</table>

<div id="empty-state" class="empty-state" style="display: none;">

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

<h3>No hay productos en el inventario</h3>

<p>Comienza agregando tu primer producto usando el formulario superior.</p>

</div>

</div>

</div>

<script>

// Clase para gestionar el inventario

class InventoryManager {

constructor() {

this.products = JSON.parse(localStorage.getItem('inventory')) || [];

this.currentEditId = null;

this.html5QrCode = null;

this.isScanning = false;

// Inicializar la aplicación

this.init();

}

init() {

// Mostrar productos existentes

this.renderProducts();

this.updateStats();

// Configurar event listeners

document.getElementById('product-form').addEventListener('submit', (e) => {

e.preventDefault();

this.handleFormSubmit();

});

document.getElementById('clear-form').addEventListener('click', () => {

this.clearForm();

});

document.getElementById('search-input').addEventListener('input', (e) => {

this.filterProducts(e.target.value);

});

document.getElementById('export-btn').addEventListener('click', () => {

this.exportData();

});

// Métodos de entrada

document.querySelectorAll('.input-method-btn').forEach(btn => {

btn.addEventListener('click', (e) => {

this.switchInputMethod(e.currentTarget.dataset.method);

});

});

// Control del escáner

document.getElementById('start-scanner').addEventListener('click', () => {

this.startScanner();

});

document.getElementById('stop-scanner').addEventListener('click', () => {

this.stopScanner();

});

document.getElementById('use-scanned-code').addEventListener('click', () => {

this.useScannedCode();

});

// Cargar datos de ejemplo si no hay productos

if (this.products.length === 0) {

this.loadSampleData();

}

}

switchInputMethod(method) {

// Actualizar botones activos

document.querySelectorAll('.input-method-btn').forEach(btn => {

btn.classList.remove('active');

});

document.querySelector(`[data-method="${method}"]`).classList.add('active');

// Mostrar/ocultar secciones

if (method === 'manual') {

document.getElementById('manual-input').style.display = 'block';

document.getElementById('scanner-input').style.display = 'none';

this.stopScanner();

} else {

document.getElementById('manual-input').style.display = 'none';

document.getElementById('scanner-input').style.display = 'block';

}

}

startScanner() {

if (this.isScanning) return;

this.html5QrCode = new Html5Qrcode("reader");

const config = {

fps: 10,

qrbox: { width: 250, height: 250 }

};

this.html5QrCode.start(

{ facingMode: "environment" },

config,

(decodedText) => {

// Código escaneado con éxito

this.onScanSuccess(decodedText);

},

(errorMessage) => {

// Error ignorado para escaneo continuo

}

).then(() => {

this.isScanning = true;

document.getElementById('start-scanner').disabled = true;

document.getElementById('stop-scanner').disabled = false;

this.showAlert('Escáner activado. Enfoca un código QR o de barras.', 'success');

}).catch(err => {

this.showAlert('Error al iniciar el escáner: ' + err, 'error');

});

}

stopScanner() {

if (!this.isScanning || !this.html5QrCode) return;

this.html5QrCode.stop().then(() => {

this.html5QrCode.clear();

this.isScanning = false;

document.getElementById('start-scanner').disabled = false;

document.getElementById('stop-scanner').disabled = true;

this.showAlert('Escáner detenido', 'success');

}).catch(err => {

this.showAlert('Error al detener el escáner: ' + err, 'error');

});

}

onScanSuccess(decodedText) {

document.getElementById('scanned-code').value = decodedText;

this.showAlert('Código escaneado: ' + decodedText, 'success');

// Detener el escáner automáticamente después de un escaneo exitoso

this.stopScanner();

}

useScannedCode() {

const scannedCode = document.getElementById('scanned-code').value;

if (!scannedCode) {

this.showAlert('No hay código escaneado para usar', 'error');

return;

}

// Verificar si el código ya existe

const existingProduct = this.products.find(p => p.code === scannedCode);

if (existingProduct) {

this.showAlert('Este código ya está registrado para: ' + existingProduct.name, 'error');

return;

}

// Cambiar a entrada manual y llenar el campo de código

this.switchInputMethod('manual');

document.getElementById('product-code').value = scannedCode;

document.getElementById('product-name').focus();

this.showAlert('Código escaneado aplicado al formulario', 'success');

}

handleFormSubmit() {

const code = document.getElementById('product-code').value.trim();

const name = document.getElementById('product-name').value.trim();

const category = document.getElementById('product-category').value.trim();

const price = parseFloat(document.getElementById('product-price').value);

const quantity = parseInt(document.getElementById('product-quantity').value);

const minimum = parseInt(document.getElementById('product-minimum').value);

// Validaciones

if (!code || !name || !category) {

this.showAlert('Por favor, completa todos los campos', 'error');

return;

}

if (price < 0 || quantity < 0 || minimum < 0) {

this.showAlert('Los valores numéricos no pueden ser negativos', 'error');

return;

}

// Verificar si el código ya existe (excepto en modo edición)

if (!this.currentEditId) {

const existingProduct = this.products.find(p => p.code === code);

if (existingProduct) {

this.showAlert('Este código ya está registrado para: ' + existingProduct.name, 'error');

return;

}

}

if (this.currentEditId) {

// Editar producto existente

this.editProduct(this.currentEditId, { code, name, category, price, quantity, minimum });

this.showAlert('Producto actualizado correctamente', 'success');

} else {

// Agregar nuevo producto

this.addProduct({ code, name, category, price, quantity, minimum });

this.showAlert('Producto agregado correctamente', 'success');

}

this.clearForm();

this.renderProducts();

this.updateStats();

}

addProduct(product) {

const newProduct = {

id: Date.now().toString(),

...product

};

this.products.push(newProduct);

this.saveToLocalStorage();

}

editProduct(id, updatedProduct) {

const index = this.products.findIndex(p => p.id === id);

if (index !== -1) {

this.products[index] = { ...this.products[index], ...updatedProduct };

this.saveToLocalStorage();

}

this.currentEditId = null;

document.querySelector('button[type="submit"]').innerHTML = '<i class="fas fa-save"></i> Agregar Producto';

}

deleteProduct(id) {

if (confirm('¿Estás seguro de que quieres eliminar este producto?')) {

this.products = this.products.filter(p => p.id !== id);

this.saveToLocalStorage();

this.renderProducts();

this.updateStats();

this.showAlert('Producto eliminado correctamente', 'success');

}

}

setEditMode(id) {

const product = this.products.find(p => p.id === id);

if (product) {

document.getElementById('product-code').value = product.code;

document.getElementById('product-name').value = product.name;

document.getElementById('product-category').value = product.category;

document.getElementById('product-price').value = product.price;

document.getElementById('product-quantity').value = product.quantity;

document.getElementById('product-minimum').value = product.minimum;

this.currentEditId = id;

document.querySelector('button[type="submit"]').innerHTML = '<i class="fas fa-edit"></i> Actualizar Producto';

// Cambiar a entrada manual si no está activa

this.switchInputMethod('manual');

// Desplazar hacia el formulario

document.getElementById('product-form').scrollIntoView({ behavior: 'smooth' });

}

}

clearForm() {

document.getElementById('product-form').reset();

document.getElementById('scanned-code').value = '';

this.currentEditId = null;

document.querySelector('button[type="submit"]').innerHTML = '<i class="fas fa-save"></i> Agregar Producto';

}

filterProducts(searchTerm) {

const filteredProducts = this.products.filter(product =>

product.name.toLowerCase().includes(searchTerm.toLowerCase()) ||

product.category.toLowerCase().includes(searchTerm.toLowerCase()) ||

product.code.toLowerCase().includes(searchTerm.toLowerCase())

);

this.renderProducts(filteredProducts);

}

renderProducts(productsToRender = null) {

const products = productsToRender || this.products;

const tbody = document.getElementById('inventory-body');

const emptyState = document.getElementById('empty-state');

if (products.length === 0) {

tbody.innerHTML = '';

emptyState.style.display = 'block';

return;

}

emptyState.style.display = 'none';

tbody.innerHTML = products.map(product => {

const isLowStock = product.quantity <= product.minimum;

const statusClass = isLowStock ? 'low-stock' : 'good-stock';

const statusText = isLowStock ? 'Stock Bajo' : 'Disponible';

const statusIcon = isLowStock ? 'fa-exclamation-triangle' : 'fa-check-circle';

return `

<tr>

<td>${product.code}</td>

<td>${product.name}</td>

<td>${product.category}</td>

<td>$${product.price.toFixed(2)}</td>

<td>${product.quantity}</td>

<td>${product.minimum}</td>

<td><span class="${statusClass}"><i class="fas ${statusIcon}"></i> ${statusText}</span></td>

<td>

<button class="btn-success" onclick="inventory.setEditMode('${product.id}')"><i class="fas fa-edit"></i> Editar</button>

<button class="btn-danger" onclick="inventory.deleteProduct('${product.id}')"><i class="fas fa-trash"></i> Eliminar</button>

</td>

</tr>

`;

}).join('');

}

updateStats() {

// Actualizar contador de productos

document.getElementById('total-count').textContent = this.products.length;

document.getElementById('products-count').textContent = this.products.length;

// Calcular valor total del inventario

const totalValue = this.products.reduce((sum, product) =>

sum + (product.price \* product.quantity), 0);

document.getElementById('total-value').textContent = `$${totalValue.toFixed(2)}`;

// Contar productos con stock bajo

const lowStockCount = this.products.filter(p => p.quantity <= p.minimum).length;

document.getElementById('low-stock-count').textContent = lowStockCount;

// Contar categorías únicas

const uniqueCategories = new Set(this.products.map(p => p.category));

document.getElementById('categories-count').textContent = uniqueCategories.size;

}

saveToLocalStorage() {

localStorage.setItem('inventory', JSON.stringify(this.products));

}

showAlert(message, type) {

const alert = document.getElementById('alert-message');

alert.textContent = message;

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

alert.style.display = 'block';

setTimeout(() => {

alert.style.display = 'none';

}, 4000);

}

exportData() {

if (this.products.length === 0) {

this.showAlert('No hay datos para exportar', 'error');

return;

}

let csvContent = "Código,Nombre,Categoría,Precio,Cantidad,Stock Mínimo,Estado\n";

this.products.forEach(product => {

const estado = product.quantity <= product.minimum ? "Stock Bajo" : "Disponible";

csvContent += `"${product.code}","${product.name}","${product.category}",${product.price},${product.quantity},${product.minimum},"${estado}"\n`;

});

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", "inventario.csv");

link.style.visibility = 'hidden';

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

this.showAlert('Datos exportados correctamente', 'success');

}

loadSampleData() {

const sampleProducts = [

{ id: '1', code: '7501006554102', name: 'Arroz Integral', category: 'Alimentos', price: 2.50, quantity: 25, minimum: 10 },

{ id: '2', code: '7501055301025', name: 'Aceite de Oliva', category: 'Alimentos', price: 8.75, quantity: 15, minimum: 5 },

{ id: '3', code: '7501008003018', name: 'Detergente Líquido', category: 'Limpieza', price: 5.20, quantity: 8, minimum: 10 },

{ id: '4', code: '7501008702015', name: 'Shampoo', category: 'Cuidado Personal', price: 4.50, quantity: 12, minimum: 5 },

{ id: '5', code: '7501006304012', name: 'Café Molido', category: 'Alimentos', price: 6.80, quantity: 4, minimum: 8 }

];

this.products = sampleProducts;

this.saveToLocalStorage();

this.renderProducts();

this.updateStats();

}

}

// Inicializar la aplicación cuando se carga la página

const inventory = new InventoryManager();

</script>

</body>

</html>