<!DOCTYPE html>

<html lang="ar" dir="rtl">

<head>

<meta charset="UTF-8">

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

<title>متجر إلكتروني</title>

<script src="https://cdn.tailwindcss.com"></script>

<style>

@import url('https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap');

body {

font-family: 'Cairo', sans-serif;

background-color: #f7f9fc;

color: #333;

}

.container {

max-width: 1200px;

}

.product-card {

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

transition: transform 0.2s;

}

.product-card:hover {

transform: translateY(-5px);

}

#cart-modal {

transition: opacity 0.3s ease-in-out;

}

.chat-container {

position: fixed;

bottom: 20px;

left: 20px;

z-index: 100;

}

#chat-window {

height: 300px;

overflow-y: auto;

border-radius: 1rem;

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

}

</style>

</head>

<body class="bg-gray-100 flex flex-col min-h-screen">

<!-- Firebase SDKs -->

<script type="module">

import { initializeApp } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-app.js";

import { getAuth, signInAnonymously, signInWithCustomToken, onAuthStateChanged } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-auth.js";

import { getFirestore, doc, getDoc, addDoc, setDoc, updateDoc, deleteDoc, onSnapshot, collection, query, where, getDocs, Timestamp, orderBy } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-firestore.js";

// Global variables for Firebase services

let app;

let db;

let auth;

let userId;

// Firebase configuration (provided by the environment)

const firebaseConfig = JSON.parse(typeof \_\_firebase\_config !== 'undefined' ? \_\_firebase\_config : '{}');

const appId = typeof \_\_app\_id !== 'undefined' ? \_\_app\_id : 'default-app-id';

const initialAuthToken = typeof \_\_initial\_auth\_token !== 'undefined' ? \_\_initial\_auth\_token : null;

// Initialize Firebase

if (Object.keys(firebaseConfig).length > 0) {

app = initializeApp(firebaseConfig);

db = getFirestore(app);

auth = getAuth(app);

console.log("Firebase initialized.");

// Handle user authentication

if (initialAuthToken) {

signInWithCustomToken(auth, initialAuthToken)

.then((userCredential) => {

const user = userCredential.user;

userId = user.uid;

console.log("User signed in with custom token:", userId);

document.getElementById('user-id-display').innerText = `معرف المستخدم: ${userId}`;

setupRealtimeListeners();

setupChatListener();

})

.catch((error) => {

console.error("Error signing in with custom token:", error);

signInAnonymously(auth).then(() => {

userId = auth.currentUser.uid;

console.log("Signed in anonymously:", userId);

document.getElementById('user-id-display').innerText = `معرف المستخدم: ${userId}`;

setupRealtimeListeners();

setupChatListener();

}).catch(e => console.error("Anonymous sign-in failed:", e));

});

} else {

signInAnonymously(auth).then(() => {

userId = auth.currentUser.uid;

console.log("Signed in anonymously:", userId);

document.getElementById('user-id-display').innerText = `معرف المستخدم: ${userId}`;

setupRealtimeListeners();

setupChatListener();

}).catch(e => console.error("Anonymous sign-in failed:", e));

}

} else {

console.error("Firebase config is missing. Firestore functionality will be disabled.");

document.getElementById('user-id-display').innerText = `لا يوجد اتصال بقاعدة البيانات`;

}

// Dummy product data (simulating a database)

const products = [

{ id: 1, name: "هاتف ذكي", description: "أحدث هاتف ذكي مع كاميرا عالية الدقة وشاشة OLED.", price: 799, image: "https://placehold.co/400x300/a0a0a0/ffffff?text=هاتف+ذكي" },

{ id: 2, name: "سماعات لاسلكية", description: "صوت نقي ومريح للغاية للاستخدام اليومي.", price: 99, image: "https://placehold.co/400x300/a0a0a0/ffffff?text=سماعات+لاسلكية" },

{ id: 3, name: "ساعة ذكية", description: "تتبع نشاطك البدني، نبضات القلب، والإشعارات.", price: 199, image: "https://placehold.co/400x300/a0a0a0/ffffff?text=ساعة+ذكية" },

{ id: 4, name: "لابتوب فائق", description: "أداء قوي وتصميم نحيف، مثالي للعمل والدراسة.", price: 1299, image: "https://placehold.co/400x300/a0a0a0/ffffff?text=لابتوب" },

{ id: 5, name: "ماوس ألعاب", description: "دقة عالية وأزرار قابلة للبرمجة لتجربة ألعاب لا مثيل لها.", price: 49, image: "https://placehold.co/400x300/a0a0a0/ffffff?text=ماوس" },

{ id: 6, name: "لوحة مفاتيح ميكانيكية", description: "تجربة كتابة مريحة وسريعة مع إضاءة خلفية RGB.", price: 129, image: "https://placehold.co/400x300/a0a0a0/ffffff?text=لوحة+مفاتيح" },

{ id: 7, name: "كاميرا احترافية", description: "التقط صورًا مذهلة بجودة عالية ومقاطع فيديو بدقة 4K.", price: 1599, image: "https://placehold.co/400x300/a0a0a0/ffffff?text=كاميرا" },

{ id: 8, name: "شاشة كمبيوتر", description: "شاشة كبيرة ومنحنية لتجربة مشاهدة غامرة.", price: 399, image: "https://placehold.co/400x300/a0a0a0/ffffff?text=شاشة" }

];

// Global cart items variable

let cartItems = [];

// Function to render products

function renderProducts(filteredProducts) {

const productList = document.getElementById('product-list');

productList.innerHTML = '';

const productsToRender = filteredProducts || products;

productsToRender.forEach(product => {

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

productCard.className = 'product-card bg-white p-6 rounded-xl shadow-lg flex flex-col items-center text-center';

productCard.innerHTML = `

<img src="${product.image}" alt="${product.name}" class="w-full h-48 object-cover rounded-lg mb-4">

<h3 class="text-xl font-bold mb-2">${product.name}</h3>

<p class="text-sm text-gray-600 mb-4">${product.description}</p>

<div class="flex items-center justify-between w-full">

<span class="text-2xl font-extrabold text-blue-600">${product.price} ر.س</span>

<button onclick="addToCart(${product.id})" class="add-to-cart-btn bg-blue-600 text-white rounded-full px-4 py-2 hover:bg-blue-700 transition duration-300">أضف إلى السلة</button>

</div>

`;

productList.appendChild(productCard);

});

}

// Function to filter products based on search query

function searchProducts(event) {

const query = event.target.value.toLowerCase();

const filtered = products.filter(product => product.name.toLowerCase().includes(query) || product.description.toLowerCase().includes(query));

renderProducts(filtered);

}

// Function to add a product to the Firestore cart

async function addToCart(productId) {

if (!db || !userId) {

console.error("Firestore is not initialized.");

showMessage("عذراً، لا يمكن إضافة المنتج للسلة الآن.");

return;

}

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

if (!product) return;

const cartRef = collection(db, `artifacts/${appId}/public/data/carts`);

const cartDocRef = doc(cartRef, userId);

try {

const docSnap = await getDoc(cartDocRef);

const currentCart = docSnap.exists() ? docSnap.data().items || [] : [];

const existingItem = currentCart.find(item => item.id === productId);

if (existingItem) {

existingItem.quantity += 1;

} else {

currentCart.push({ ...product, quantity: 1 });

}

await setDoc(cartDocRef, { items: currentCart });

showMessage("تمت إضافة المنتج إلى السلة بنجاح!");

} catch (e) {

console.error("Error adding document to cart: ", e);

showMessage("حدث خطأ أثناء إضافة المنتج.");

}

}

// Function to remove an item from the cart

async function removeItem(productId) {

if (!db || !userId) return;

const cartRef = collection(db, `artifacts/${appId}/public/data/carts`);

const cartDocRef = doc(cartRef, userId);

try {

const docSnap = await getDoc(cartDocRef);

if (docSnap.exists()) {

const currentCart = docSnap.data().items || [];

const updatedCart = currentCart.filter(item => item.id !== productId);

await setDoc(cartDocRef, { items: updatedCart });

showMessage("تم حذف المنتج من السلة.");

}

} catch (e) {

console.error("Error removing document from cart: ", e);

showMessage("حدث خطأ أثناء حذف المنتج.");

}

}

// Function to clear the cart

async function clearCart() {

if (!db || !userId) return;

const cartRef = collection(db, `artifacts/${appId}/public/data/carts`);

const cartDocRef = doc(cartRef, userId);

try {

await setDoc(cartDocRef, { items: [] });

showMessage("تم إفراغ سلة التسوق.");

} catch (e) {

console.error("Error clearing cart: ", e);

}

}

// Function to simulate checkout

function checkout() {

if (cartItems.length === 0) {

showMessage("سلة التسوق فارغة!");

return;

}

showMessage("تم إتمام عملية الدفع بنجاح! شكراً لك.");

clearCart();

closeCartModal();

}

// Function to show a message in a custom modal (not alert())

function showMessage(message) {

const messageBox = document.getElementById('message-box');

messageBox.textContent = message;

messageBox.classList.remove('hidden');

setTimeout(() => {

messageBox.classList.add('hidden');

}, 3000);

}

// Function to render cart items

function renderCart(items) {

const cartItemsList = document.getElementById('cart-items');

const cartTotal = document.getElementById('cart-total');

cartItemsList.innerHTML = '';

let total = 0;

if (items.length === 0) {

cartItemsList.innerHTML = `<li class="text-center text-gray-500">سلة التسوق فارغة.</li>`;

} else {

items.forEach(item => {

const cartItem = document.createElement('li');

cartItem.className = 'flex items-center justify-between py-4 border-b border-gray-200';

cartItem.innerHTML = `

<div class="flex items-center">

<img src="${item.image}" alt="${item.name}" class="w-16 h-16 rounded-md object-cover mr-4">

<div>

<h4 class="font-bold">${item.name}</h4>

<p class="text-sm text-gray-500">الكمية: ${item.quantity}</p>

</div>

</div>

<div class="flex items-center">

<span class="text-lg font-bold ml-4">${item.price \* item.quantity} ر.س</span>

<button onclick="removeItem(${item.id})" class="text-red-500 hover:text-red-700 transition duration-300">

<svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M19 7l-.867 12.142A2 2 0 0116.013 21H7.987a2 2 0 01-1.92-1.858L5 7m5 4v6m4-6v6m1-10V4a1 1 0 00-1-1h-4a1 1 0 00-1 1v3M4 7h16" />

</svg>

</button>

</div>

`;

cartItemsList.appendChild(cartItem);

total += item.price \* item.quantity;

});

}

cartTotal.textContent = `${total} ر.س`;

}

// Chat functionality

async function sendMessage() {

if (!db || !userId) return;

const chatInput = document.getElementById('chat-input');

const messageText = chatInput.value.trim();

if (messageText === "") return;

const messagesRef = collection(db, `artifacts/${appId}/public/data/chat`);

try {

await addDoc(messagesRef, {

userId: userId,

message: messageText,

timestamp: Timestamp.now()

});

chatInput.value = "";

} catch (e) {

console.error("Error adding message: ", e);

}

}

function setupChatListener() {

if (!db) return;

const chatMessages = document.getElementById('chat-messages');

const messagesRef = collection(db, `artifacts/${appId}/public/data/chat`);

const q = query(messagesRef, orderBy("timestamp"));

onSnapshot(q, (snapshot) => {

chatMessages.innerHTML = '';

snapshot.forEach(doc => {

const data = doc.data();

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

const isCurrentUser = data.userId === userId;

messageElement.className = `p-2 rounded-lg my-2 mx-4 max-w-[80%] break-words ${isCurrentUser ? 'bg-blue-500 text-white ml-auto' : 'bg-gray-200 text-gray-800 mr-auto'}`;

messageElement.textContent = data.message;

chatMessages.appendChild(messageElement);

});

chatMessages.scrollTop = chatMessages.scrollHeight; // Auto-scroll to latest message

});

}

// Real-time listener for cart updates

function setupRealtimeListeners() {

if (!db || !userId) return;

const cartRef = collection(db, `artifacts/${appId}/public/data/carts`);

const cartDocRef = doc(cartRef, userId);

onSnapshot(cartDocRef, (docSnap) => {

cartItems = docSnap.exists() ? docSnap.data().items || [] : [];

renderCart(cartItems);

updateCartCount(cartItems.length);

});

}

function updateCartCount(count) {

const cartCountElement = document.getElementById('cart-count');

cartCountElement.textContent = count;

if (count > 0) {

cartCountElement.classList.remove('hidden');

} else {

cartCountElement.classList.add('hidden');

}

}

function openCartModal() {

document.getElementById('cart-modal').classList.remove('opacity-0', 'pointer-events-none');

}

function closeCartModal() {

document.getElementById('cart-modal').classList.add('opacity-0', 'pointer-events-none');

}

// Expose functions to the global scope

window.addToCart = addToCart;

window.removeItem = removeItem;

window.checkout = checkout;

window.openCartModal = openCartModal;

window.closeCartModal = closeCartModal;

window.searchProducts = searchProducts;

window.sendMessage = sendMessage;

// Initial rendering on load

window.onload = () => {

renderProducts();

document.getElementById('cart-icon').addEventListener('click', openCartModal);

document.getElementById('close-cart-btn').addEventListener('click', closeCartModal);

document.getElementById('checkout-btn').addEventListener('click', checkout);

};

</script>

<!-- Message Box -->

<div id="message-box" class="hidden fixed bottom-5 left-1/2 -translate-x-1/2 bg-green-500 text-white px-6 py-3 rounded-full shadow-lg z-50 transition duration-300">

تمت العملية بنجاح!

</div>

<!-- Header -->

<header class="bg-white shadow-md sticky top-0 z-50">

<div class="container mx-auto px-6 py-4 flex items-center justify-between">

<a href="#" class="text-3xl font-bold text-blue-600">متجرنا</a>

<div class="relative flex-grow mx-4">

<input type="text" id="search-input" oninput="searchProducts(event)" placeholder="ابحث عن منتجات..." class="w-full px-4 py-2 rounded-full border border-gray-300 focus:outline-none focus:ring-2 focus:ring-blue-500 transition duration-300">

<span class="absolute right-4 top-1/2 -translate-y-1/2 text-gray-400">

<svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" viewBox="0 0 20 20" fill="currentColor">

<path fill-rule="evenodd" d="M8 4a4 4 0 100 8 4 4 0 000-8zM2 8a6 6 0110 0 6 6 0 01-10 0z" clip-rule="evenodd" />

<path fill-rule="evenodd" d="M12.95 12.95a1 1 0 011.414 0l3.854 3.854a1 1 0 01-1.414 1.414l-3.854-3.854a1 1 0 010-1.414z" clip-rule="evenodd" />

</svg>

</span>

</div>

<div class="relative cursor-pointer" id="cart-icon">

<svg xmlns="http://www.w3.org/2000/svg" class="h-8 w-8 text-blue-600" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M3 3h2l.4 2M7 13h10l1.4-7H5.4M7 13L5.4 5H19M7 13L7 17H17M10 17H14" />

</svg>

<span id="cart-count" class="absolute -top-2 -right-2 bg-red-500 text-white text-xs font-bold rounded-full h-5 w-5 flex items-center justify-center hidden">0</span>

</div>

</div>

</header>

<!-- Main Content -->

<main class="flex-grow container mx-auto px-6 py-8">

<h2 class="text-3xl font-bold text-center mb-8">اكتشف منتجاتنا المميزة</h2>

<div id="user-id-display" class="text-center text-sm text-gray-500 mb-4"></div>

<div id="product-list" class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-8">

<!-- Products will be rendered here by JavaScript -->

</div>

</main>

<!-- Chat Window -->

<div class="chat-container">

<button id="chat-toggle" class="bg-green-500 text-white p-4 rounded-full shadow-lg">

<svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M8 10h.01M12 10h.01M16 10h.01M9 16l-3 3v-2a1 1 0 011-1h6a1 1 0 011 1v2l-3-3m-6 0h12a2 2 0 002-2V6a2 2 0 00-2-2H6a2 2 0 00-2 2v6a2 2 0 002 2z" />

</svg>

</button>

<div id="chat-window" class="hidden bg-white w-80 rounded-xl flex flex-col">

<div class="p-4 bg-gray-200 text-gray-800 rounded-t-xl text-center font-bold">محادثة مع الدعم</div>

<div id="chat-messages" class="flex-grow p-4 space-y-2">

<!-- Chat messages will be rendered here -->

</div>

<div class="p-4 border-t border-gray-200 flex">

<input type="text" id="chat-input" placeholder="اكتب رسالتك..." class="flex-grow rounded-full px-4 py-2 bg-gray-100 border focus:outline-none">

<button onclick="sendMessage()" class="bg-blue-600 text-white rounded-full px-4 py-2 ml-2 hover:bg-blue-700 transition duration-300">

<svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12 19l9 2-9-18-9 18 9-2zm0 0v-8" />

</svg>

</button>

</div>

</div>

</div>

<script>

document.getElementById('chat-toggle').addEventListener('click', () => {

const chatWindow = document.getElementById('chat-window');

chatWindow.classList.toggle('hidden');

});

document.getElementById('chat-input').addEventListener('keypress', (event) => {

if (event.key === 'Enter') {

sendMessage();

}

});

</script>

<!-- Cart Modal -->

<div id="cart-modal" class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 opacity-0 pointer-events-none transition-all">

<div class="bg-white rounded-2xl shadow-xl w-11/12 md:w-2/3 lg:w-1/2 p-8 relative">

<button id="close-cart-btn" class="absolute top-4 right-4 text-gray-400 hover:text-gray-600 transition duration-300">

<svg xmlns="http://www.w3.org/2000/svg" class="h-8 w-8" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M6 18L18 6M6 6l12 12" />

</svg>

</button>

<h3 class="text-3xl font-bold mb-6 text-center">سلة التسوق</h3>

<ul id="cart-items" class="divide-y divide-gray-200 max-h-80 overflow-y-auto">

<!-- Cart items will be rendered here -->

</ul>

<div class="mt-8 pt-4 border-t-2 border-dashed border-gray-300 flex justify-between items-center font-bold text-2xl">

<span>الإجمالي:</span>

<span id="cart-total">0 ر.س</span>

</div>

<button id="checkout-btn" class="w-full bg-green-500 text-white text-xl font-bold py-4 rounded-full mt-6 hover:bg-green-600 transition duration-300">

إتمام عملية الشراء

</button>

</div>

</div>

</body>

</html>