import React, { useState, useEffect } from 'react';

import { ShoppingCart, Search, Heart, User, Menu, X, Star, Shield, Truck, RefreshCw, ChevronRight, TrendingUp, Filter, Check } from 'lucide-react';

export default function ECommercePlatform() {

const [cart, setCart] = useState([]);

const [searchQuery, setSearchQuery] = useState('');

const [selectedCategory, setSelectedCategory] = useState('all');

const [showMobileMenu, setShowMobileMenu] = useState(false);

const [showCart, setShowCart] = useState(false);

const [wishlist, setWishlist] = useState([]);

const [showFilters, setShowFilters] = useState(false);

const [toast, setToast] = useState(null);

const products = [

{ id: 1, name: 'Wireless Headphones', price: 79.99, category: 'electronics', rating: 4.5, reviews: 245, stock: 15, image: '🎧', shipping: 'Free 2-day shipping', description: 'Premium noise-canceling wireless headphones' },

{ id: 2, name: 'Smart Watch', price: 199.99, category: 'electronics', rating: 4.7, reviews: 532, stock: 8, image: '⌚', shipping: 'Free shipping', description: 'Advanced fitness tracking smartwatch' },

{ id: 3, name: 'Running Shoes', price: 89.99, category: 'sports', rating: 4.3, reviews: 189, stock: 25, image: '👟', shipping: 'Free returns', description: 'Comfortable running shoes with arch support' },

{ id: 4, name: 'Laptop Backpack', price: 49.99, category: 'accessories', rating: 4.6, reviews: 312, stock: 42, image: '🎒', shipping: 'Free shipping', description: 'Durable laptop backpack with USB port' },

{ id: 5, name: 'Coffee Maker', price: 129.99, category: 'home', rating: 4.8, reviews: 678, stock: 12, image: '☕', shipping: 'Free 2-day shipping', description: 'Programmable coffee maker with grinder' },

{ id: 6, name: 'Yoga Mat', price: 34.99, category: 'sports', rating: 4.4, reviews: 421, stock: 50, image: '🧘', shipping: 'Free shipping', description: 'Non-slip eco-friendly yoga mat' },

];

const categories = [

{ id: 'all', name: 'All Products', icon: '🛍️' },

{ id: 'electronics', name: 'Electronics', icon: '💻' },

{ id: 'sports', name: 'Sports', icon: '⚽' },

{ id: 'accessories', name: 'Accessories', icon: '👜' },

{ id: 'home', name: 'Home', icon: '🏠' },

];

const filteredProducts = products.filter(p =>

(selectedCategory === 'all' || p.category === selectedCategory) &&

(searchQuery === '' || p.name.toLowerCase().includes(searchQuery.toLowerCase()))

);

const showToast = (message) => {

setToast(message);

setTimeout(() => setToast(null), 3000);

};

const addToCart = (product) => {

const existing = cart.find(item => item.id === product.id);

if (existing) {

setCart(cart.map(item =>

item.id === product.id ? { ...item, quantity: item.quantity + 1 } : item

));

} else {

setCart([...cart, { ...product, quantity: 1 }]);

}

showToast(`${product.name} added to cart!`);

};

const removeFromCart = (productId) => {

setCart(cart.filter(item => item.id !== productId));

};

const updateQuantity = (productId, change) => {

setCart(cart.map(item => {

if (item.id === productId) {

const newQuantity = item.quantity + change;

return newQuantity > 0 ? { ...item, quantity: newQuantity } : item;

}

return item;

}).filter(item => item.quantity > 0));

};

const toggleWishlist = (productId) => {

if (wishlist.includes(productId)) {

setWishlist(wishlist.filter(id => id !== productId));

showToast('Removed from wishlist');

} else {

setWishlist([...wishlist, productId]);

showToast('Added to wishlist');

}

};

const cartTotal = cart.reduce((sum, item) => sum + (item.price \* item.quantity), 0);

const cartItemCount = cart.reduce((sum, item) => sum + item.quantity, 0);

return (

<div className="min-h-screen bg-gradient-to-br from-blue-50 via-white to-purple-50">

{/\* Toast Notification \*/}

{toast && (

<div className="fixed top-4 right-4 z-50 bg-green-500 text-white px-6 py-3 rounded-lg shadow-lg animate-fade-in flex items-center gap-2">

<Check className="w-5 h-5" />

{toast}

</div>

)}

{/\* Header \*/}

<header className="bg-white shadow-md sticky top-0 z-40">

<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

<div className="flex justify-between items-center py-4">

<div className="flex items-center gap-8">

<h1 className="text-2xl font-bold bg-gradient-to-r from-blue-600 to-purple-600 bg-clip-text text-transparent">

ShopSmart

</h1>

<nav className="hidden md:flex gap-6">

{categories.slice(0, 4).map(cat => (

<button

key={cat.id}

onClick={() => setSelectedCategory(cat.id)}

className={`text-sm font-medium transition-colors ${

selectedCategory === cat.id ? 'text-blue-600' : 'text-gray-600 hover:text-blue-600'

}`}

>

{cat.name}

</button>

))}

</nav>

</div>

<div className="flex items-center gap-4">

<div className="hidden sm:flex items-center bg-gray-100 rounded-full px-4 py-2 max-w-md">

<Search className="w-4 h-4 text-gray-400 mr-2" />

<input

type="text"

placeholder="Search products..."

value={searchQuery}

onChange={(e) => setSearchQuery(e.target.value)}

className="bg-transparent outline-none text-sm w-64"

/>

</div>

<button className="p-2 hover:bg-gray-100 rounded-full relative">

<Heart className="w-6 h-6 text-gray-600" />

{wishlist.length > 0 && (

<span className="absolute -top-1 -right-1 bg-red-500 text-white text-xs rounded-full w-5 h-5 flex items-center justify-center">

{wishlist.length}

</span>

)}

</button>

<button

onClick={() => setShowCart(!showCart)}

className="p-2 hover:bg-gray-100 rounded-full relative"

>

<ShoppingCart className="w-6 h-6 text-gray-600" />

{cartItemCount > 0 && (

<span className="absolute -top-1 -right-1 bg-blue-600 text-white text-xs rounded-full w-5 h-5 flex items-center justify-center">

{cartItemCount}

</span>

)}

</button>

<button className="p-2 hover:bg-gray-100 rounded-full">

<User className="w-6 h-6 text-gray-600" />

</button>

<button

onClick={() => setShowMobileMenu(!showMobileMenu)}

className="md:hidden p-2 hover:bg-gray-100 rounded-full"

>

{showMobileMenu ? <X className="w-6 h-6" /> : <Menu className="w-6 h-6" />}

</button>

</div>

</div>

{/\* Mobile Search \*/}

<div className="sm:hidden pb-4">

<div className="flex items-center bg-gray-100 rounded-full px-4 py-2">

<Search className="w-4 h-4 text-gray-400 mr-2" />

<input

type="text"

placeholder="Search products..."

value={searchQuery}

onChange={(e) => setSearchQuery(e.target.value)}

className="bg-transparent outline-none text-sm w-full"

/>

</div>

</div>

</div>

</header>

{/\* Trust Badges \*/}

<div className="bg-gradient-to-r from-blue-600 to-purple-600 text-white py-3">

<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

<div className="flex flex-wrap justify-center gap-6 text-sm">

<div className="flex items-center gap-2">

<Shield className="w-4 h-4" />

<span>Secure Payment</span>

</div>

<div className="flex items-center gap-2">

<Truck className="w-4 h-4" />

<span>Free Shipping Over $50</span>

</div>

<div className="flex items-center gap-2">

<RefreshCw className="w-4 h-4" />

<span>Easy 30-Day Returns</span>

</div>

<div className="flex items-center gap-2">

<Star className="w-4 h-4" />

<span>4.8/5 Customer Rating</span>

</div>

</div>

</div>

</div>

{/\* Main Content \*/}

<main className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-8">

{/\* Category Pills \*/}

<div className="flex gap-2 mb-6 overflow-x-auto pb-2">

{categories.map(cat => (

<button

key={cat.id}

onClick={() => setSelectedCategory(cat.id)}

className={`px-4 py-2 rounded-full text-sm font-medium whitespace-nowrap transition-all ${

selectedCategory === cat.id

? 'bg-blue-600 text-white shadow-lg scale-105'

: 'bg-white text-gray-700 hover:bg-gray-100'

}`}

>

<span className="mr-1">{cat.icon}</span>

{cat.name}

</button>

))}

</div>

{/\* Products Grid \*/}

<div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">

{filteredProducts.map(product => (

<div key={product.id} className="bg-white rounded-xl shadow-md hover:shadow-xl transition-all duration-300 overflow-hidden group">

<div className="relative">

<div className="h-48 bg-gradient-to-br from-blue-100 to-purple-100 flex items-center justify-center text-6xl">

{product.image}

</div>

<button

onClick={() => toggleWishlist(product.id)}

className="absolute top-3 right-3 p-2 bg-white rounded-full shadow-md hover:scale-110 transition-transform"

>

<Heart

className={`w-5 h-5 ${wishlist.includes(product.id) ? 'fill-red-500 text-red-500' : 'text-gray-400'}`}

/>

</button>

{product.stock < 10 && (

<div className="absolute top-3 left-3 bg-red-500 text-white text-xs px-2 py-1 rounded-full">

Only {product.stock} left!

</div>

)}

</div>

<div className="p-5">

<h3 className="font-semibold text-lg mb-1">{product.name}</h3>

<p className="text-sm text-gray-600 mb-2">{product.description}</p>

<div className="flex items-center gap-1 mb-2">

<div className="flex">

{[...Array(5)].map((\_, i) => (

<Star

key={i}

className={`w-4 h-4 ${i < Math.floor(product.rating) ? 'fill-yellow-400 text-yellow-400' : 'text-gray-300'}`}

/>

))}

</div>

<span className="text-sm text-gray-600">({product.reviews})</span>

</div>

<div className="flex items-center gap-2 mb-3 text-sm text-green-600">

<Truck className="w-4 h-4" />

<span>{product.shipping}</span>

</div>

<div className="flex items-center justify-between">

<span className="text-2xl font-bold text-blue-600">${product.price}</span>

<button

onClick={() => addToCart(product)}

className="bg-gradient-to-r from-blue-600 to-purple-600 text-white px-6 py-2 rounded-full hover:shadow-lg transition-all hover:scale-105"

>

Add to Cart

</button>

</div>

</div>

</div>

))}

</div>

{filteredProducts.length === 0 && (

<div className="text-center py-20">

<p className="text-gray-500 text-lg">No products found matching your search.</p>

</div>

)}

</main>

{/\* Cart Sidebar \*/}

{showCart && (

<>

<div

className="fixed inset-0 bg-black bg-opacity-50 z-40"

onClick={() => setShowCart(false)}

/>

<div className="fixed right-0 top-0 h-full w-full sm:w-96 bg-white shadow-2xl z-50 overflow-y-auto">

<div className="p-6">

<div className="flex justify-between items-center mb-6">

<h2 className="text-2xl font-bold">Shopping Cart</h2>

<button onClick={() => setShowCart(false)} className="p-2 hover:bg-gray-100 rounded-full">

<X className="w-6 h-6" />

</button>

</div>

{cart.length === 0 ? (

<div className="text-center py-20">

<ShoppingCart className="w-16 h-16 text-gray-300 mx-auto mb-4" />

<p className="text-gray-500">Your cart is empty</p>

</div>

) : (

<>

<div className="space-y-4 mb-6">

{cart.map(item => (

<div key={item.id} className="flex gap-4 bg-gray-50 p-4 rounded-lg">

<div className="w-16 h-16 bg-gradient-to-br from-blue-100 to-purple-100 rounded-lg flex items-center justify-center text-2xl">

{item.image}

</div>

<div className="flex-1">

<h3 className="font-semibold mb-1">{item.name}</h3>

<p className="text-blue-600 font-bold">${item.price}</p>

<div className="flex items-center gap-2 mt-2">

<button

onClick={() => updateQuantity(item.id, -1)}

className="w-6 h-6 bg-gray-200 rounded-full flex items-center justify-center hover:bg-gray-300"

>

-

</button>

<span className="w-8 text-center">{item.quantity}</span>

<button

onClick={() => updateQuantity(item.id, 1)}

className="w-6 h-6 bg-gray-200 rounded-full flex items-center justify-center hover:bg-gray-300"

>

+

</button>

<button

onClick={() => removeFromCart(item.id)}

className="ml-auto text-red-500 text-sm hover:underline"

>

Remove

</button>

</div>

</div>

</div>

))}

</div>

<div className="border-t pt-4">

<div className="flex justify-between mb-2 text-sm">

<span>Subtotal</span>

<span>${cartTotal.toFixed(2)}</span>

</div>

<div className="flex justify-between mb-2 text-sm">

<span>Shipping</span>

<span className="text-green-600">FREE</span>

</div>

<div className="flex justify-between mb-4 text-lg font-bold">

<span>Total</span>

<span className="text-blue-600">${cartTotal.toFixed(2)}</span>

</div>

<button className="w-full bg-gradient-to-r from-blue-600 to-purple-600 text-white py-3 rounded-full font-semibold hover:shadow-lg transition-all mb-2">

Proceed to Checkout

</button>

<button

onClick={() => setShowCart(false)}

className="w-full border border-gray-300 py-3 rounded-full font-semibold hover:bg-gray-50 transition-all"

>

Continue Shopping

</button>

</div>

</>

)}

</div>

</div>

</>

)}

</div>

);

}