**EcommerceApp.jsx**

import { useState } from 'react';

import './style.css';

export default function EcommerceApp() {

// Sample product data

const initialProducts = [

{ id: 1, name: 'Laptop', price: 999, category: 'Electronics' },

{ id: 2, name: 'Headphones', price: 99, category: 'Electronics' },

{ id: 3, name: 'T-shirt', price: 25, category: 'Clothing' },

{ id: 4, name: 'Jeans', price: 45, category: 'Clothing' },

{ id: 5, name: 'Coffee Mug', price: 15, category: 'Home' },

{ id: 6, name: 'Desk Lamp', price: 35, category: 'Home' }

];

// State management

const [products] = useState(initialProducts);

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

const [activeCategory, setActiveCategory] = useState('All');

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

// Get unique categories

const categories = ['All', ...new Set(products.map(product => product.category))];

// Filter products by category

const filteredProducts = activeCategory === 'All'

? products

: products.filter(product => product.category === activeCategory);

// Add to cart function

const addToCart = (product) => {

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

if (existingItem) {

setCart(cart.map(item =>

item.id === product.id

? { ...item, quantity: item.quantity + 1 }

: item

));

} else {

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

}

};

// Remove from cart function

const removeFromCart = (productId) => {

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

};

// Calculate total price

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

return (

<div className="container">

<header>

<h1 className="shop-title">Simple Shop</h1>

<button

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

className="cart-button"

>

Cart ({cart.reduce((total, item) => total + item.quantity, 0)})

</button>

</header>

<div className="main-layout">

{/\* Main content \*/}

<div className={`content-area ${showCart ? 'cart-visible' : ''}`}>

{/\* Category tabs \*/}

<div className="category-container">

{categories.map(category => (

<button

key={category}

className={`category-tab ${activeCategory === category ? 'active' : 'inactive'}`}

onClick={() => setActiveCategory(category)}

>

{category}

</button>

))}

</div>

{/\* Products grid \*/}

<div className="products-grid">

{filteredProducts.map(product => (

<div key={product.id} className="product-card">

<div className="product-image">

Product Image

</div>

<h3 className="product-name">{product.name}</h3>

<p className="product-category">{product.category}</p>

<p className="product-price">${product.price}</p>

<button

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

className="add-to-cart-button"

>

Add to Cart

</button>

</div>

))}

</div>

</div>

{/\* Cart sidebar \*/}

<div className={`cart-sidebar ${showCart ? '' : 'cart-hidden'}`}>

<div className="cart-header">

<h2 className="cart-title">Your Cart</h2>

<button

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

className="close-cart-button"

>

Close

</button>

</div>

{cart.length === 0 ? (

<p className="empty-cart-message">Your cart is empty</p>

) : (

<>

<div className="cart-items">

{cart.map(item => (

<div key={item.id} className="cart-item">

<div className="cart-item-info">

<p>{item.name}</p>

<p>${item.price} × {item.quantity}</p>

</div>

<div className="cart-item-price-actions">

<p className="cart-item-price">${item.price \* item.quantity}</p>

<button

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

className="remove-item-button"

>

×

</button>

</div>

</div>

))}

</div>

<div className="cart-total">

<span>Total:</span>

<span>${totalPrice}</span>

</div>

<button className="checkout-button">

Checkout

</button>

</>

)}

</div>

</div>

</div>

);

}

OUTPUT



