**NM script.js**

// 1. PRODUCT DATA (The Catalogue)﻿const products = [﻿    { id: 1, name: "Laptop Pro 2024", price: 499.99, category: "Electronics", inStock: true },﻿    { id: 2, name: "Cotton T-Shirt", price: 19.99, category: "Apparel", inStock: true },﻿    { id: 3, name: "JavaScript Mastery Book", price: 45.00, category: "Books", inStock: true },﻿    { id: 4, name: "Noise Cancelling Headphones", price: 120.50, category: "Electronics", inStock: false },﻿    { id: 5, name: "Denim Jeans Slim Fit", price: 75.00, category: "Apparel", inStock: true },﻿    { id: 6, name: "Web Dev Basics Guide", price: 12.00, category: "Books", inStock: false },﻿    { id: 7, name: "4K Monitor 27-inch", price: 350.00, category: "Electronics", inStock: true },﻿    { id: 8, name: "Leather Jacket", price: 299.99, category: "Apparel", inStock: true },﻿];﻿const productGrid = document.getElementById('product-grid');﻿const filterInputs = document.querySelectorAll('.sidebar input[type="checkbox"], #price-range');﻿// 2. RENDERING FUNCTION﻿function renderProducts(productList) {﻿    productGrid.innerHTML = ''; // Clear previous products﻿    if (productList.length === 0) {﻿        productGrid.innerHTML = '<p id="loading-message">No products match your current filters.</p>';﻿        return;﻿    }﻿    productList.forEach(product => {﻿        const card = document.createElement('div');﻿        card.className = `product-card ${product.inStock ? '' : 'out-of-stock'}`;﻿﻿        // Simple rendering of product data﻿        card.innerHTML = `﻿            <h3>${product.name}</h3>﻿            <p>Category: ${product.category}</p>﻿            <p class="price">$${product.price.toFixed(2)}</p>﻿            <p>Status: ${product.inStock ? 'In Stock' : 'Out of Stock'}</p>﻿        `;﻿        productGrid.appendChild(card);﻿    });﻿}﻿// 3. FILTERING LOGIC﻿function applyFilters() {﻿    let filteredProducts = [...products]; // Start with the full list﻿﻿    // --- 3a. CATEGORY FILTER ---﻿    const activeCategories = [Array.fr](Array.fr#Array.fr)om(document.querySelectorAll('input[data-filter="category"]:checked'))﻿                                  .map(input => input.value);﻿﻿    if (activeCategories.length > 0) {﻿        filteredProducts = filteredProducts.filter(product => ﻿            activeCategories.includes(product.category)﻿        );﻿    }﻿﻿    // --- 3b. PRICE RANGE FILTER ---﻿    const maxPrice = parseFloat(document.getElementById('price-range').value);﻿﻿    // Update the visual price display﻿    document.getElementById('price-value').textContent = maxPrice.toFixed(0); ﻿﻿    filteredProducts = filteredProducts.filter(product => ﻿        product.price <= maxPrice﻿    );﻿    // --- 3c. IN STOCK FILTER ---﻿    const inStockChecked = document.getElementById('in-stock-filter').checked;﻿    if (inStockChecked) {﻿        filteredProducts = filteredProducts.filter(product => product.inStock === true);﻿    }﻿﻿    renderProducts(filteredProducts);﻿}﻿// 4. EVENT LISTENERS﻿filterInputs.forEach(input => {﻿    // Every time a filter input changes, re-apply all filters﻿    input.addEventListener('change', applyFilters);﻿});﻿// Price range slider needs input event for real-time update﻿document.getElementById('price-range').addEventListener('input', applyFilters);﻿// Initial load: render all products﻿window.onload = () => {﻿    applyFilters(); ﻿};﻿// 5. CLEAR ALL FUNCTION﻿function clearAllFilters() {﻿    // Uncheck all checkboxes﻿    document.querySelectorAll('input[type="checkbox"]').forEach(checkbox => {﻿        checkbox.checked = false;﻿    });﻿    // Reset price range slider to max value﻿    const priceRange = document.getElementById('price-range');﻿    priceRange.value = priceRange.max;﻿    // Apply filters to reset the grid﻿    applyFilters();﻿}