```
/* ==================================
   Product Catalog with Filter (JS Only)
   Author: Revathi
   Project: NM Mini Project
   ================================== */

// Sample product data
const products = [
  {
    id: 1,
    name: "Samsung Galaxy S23",
    category: "Mobile",
    price: 74999,
    brand: "Samsung",
    rating: 4.7,
    stock: 10
  },
  {
    id: 2,
    name: "Apple iPhone 14",
    category: "Mobile",
    price: 79999,
    brand: "Apple",
    rating: 4.8,
    stock: 8
  },
  {
    id: 3,
    name: "OnePlus 12",
    category: "Mobile",
  price: 59999,
  brand: "OnePlus",
  rating: 4.5,
  stock: 15
},
{
  id: 4,
  name: "Sony WH-1000XM5",
  category: "Headphones",
  price: 29999,
  brand: "Sony",
  rating: 4.9,
  stock: 20
},
{
  id: 5,
  name: "JBL Tune 720BT",
  category: "Headphones",
  price: 4999,
  brand: "JBL",
  rating: 4.3,
  stock: 30
},
{
  id: 6,
  name: "HP Pavilion 15",
  category: "Laptop",
  price: 69999,
  brand: "HP",
  rating: 4.6,
  stock: 12
},
{
  id: 7,
  name: "ASUS TUF Gaming F15",
  category: "Laptop",
  price: 84999,
  brand: "ASUS",
  rating: 4.8,
  stock: 7
},
{
  id: 8,
  name: "Apple MacBook Air M2",
  category: "Laptop",
  price: 104999,
  brand: "Apple",
  rating: 4.9,
  stock: 5
},
{
  id: 9,
  name: "Canon EOS 3000D",
  category: "Camera",
  price: 34999,
  brand: "Canon",
  rating: 4.4,
  stock: 11
},
{
    id: 10,
    name: "Nikon D5600",
    category: "Camera",
    price: 52999,
    brand: "Nikon",
    rating: 4.6,
    stock: 6
  }
];

// Global state
let filters = {
  category: "All",
  brand: "All",
  minPrice: 0,
  maxPrice: 200000,
  searchQuery: "",
  sortBy: "name"
};

// Helper function to format price
function formatPrice(price) {
  return "₹" + price.toLocaleString("en-IN");
}

// Display product list (mocked console output for now)
function displayProducts(productList) {
  console.clear();
  console.log("=== PRODUCT CATALOG ===");
  if (productList.length === 0) {
    console.log("No products found for selected filters.");
    return;
  }
  productList.forEach((p) => {
    console.log(
      ${p.name} | ${p.brand} | ${p.category} | ${formatPrice(p.price)} | ☆${p.rating}
    );
  });
  console.log(\nTotal Products: ${productList.length});
}

// Apply filters and sorting
function applyFilters() {
  let filtered = [...products];

  // Category filter
  if (filters.category !== "All") {
    filtered = filtered.filter((p) => p.category === filters.category);
  }

  // Brand filter
  if (filters.brand !== "All") {
    filtered = filtered.filter((p) => p.brand === filters.brand);
  }

  // Price filter
  filtered = filtered.filter(
    (p) => p.price >= filters.minPrice && p.price <= filters.maxPrice
  );
  // Search filter
  if (filters.searchQuery.trim() !== "") {
    filtered = filtered.filter((p) =>
      p.name.toLowerCase().includes(filters.searchQuery.toLowerCase())
    );
  }

  // Sorting logic
  switch (filters.sortBy) {
    case "priceLowHigh":
      filtered.sort((a, b) => a.price - b.price);
      break;
    case "priceHighLow":
      filtered.sort((a, b) => b.price - a.price);
      break;
    case "ratingHighLow":
      filtered.sort((a, b) => b.rating - a.rating);
      break;
    default:
      filtered.sort((a, b) => a.name.localeCompare(b.name));
  }

  displayProducts(filtered);
}

// Functions to update filters
function setCategory(cat) {
  filters.category = cat;
  console.log(Filter: Category → ${cat});
  applyFilters();
```

```
}

function setBrand(brand) {
  filters.brand = brand;
  console.log(Filter: Brand → ${brand});
  applyFilters();
}

function setPriceRange(min, max) {
  filters.minPrice = min;
  filters.maxPrice = max;
  console.log(Filter: Price Range → ₹${min} - ₹${max});
  applyFilters();
}

function setSearch(query) {
  filters.searchQuery = query;
  console.log(Search Query: ${query});
  applyFilters();
}

function setSort(option) {
  filters.sortBy = option;
  console.log(Sorting By: ${option});
  applyFilters();
}

// Reset filters
function resetFilters() {
  filters = {
    category: "All",
    brand: "All",
    minPrice: 0,
    maxPrice: 200000,
    searchQuery: "",
    sortBy: "name"
  };
  console.log("Filters Reset to Default");
  applyFilters();
}

// Example UI event simulation
function simulateUserActions() {
  console.log("Welcome to Product Catalog Simulation\n");

  // Initial display
  applyFilters();

  // Step 1: Category filter
  setCategory("Mobile");

  // Step 2: Brand filter
  setBrand("Samsung");

  // Step 3: Price range
  setPriceRange(50000, 90000);

  // Step 4: Sorting by price low to high
  setSort("priceLowHigh");
  // Step 5: Search product
  setSearch("Galaxy");

  // Step 6: Reset all filters
  resetFilters();

  // Step 7: Sorting by rating
  setSort("ratingHighLow");

  // Step 8: Category: Laptop
  setCategory("Laptop");
}

// Utility: Get unique categories
function getUniqueCategories() {
  const cats = [...new Set(products.map((p) => p.category))];
  return ["All", ...cats];
}

// Utility: Get unique brands
function getUniqueBrands() {
  const brands = [...new Set(products.map((p) => p.brand))];
  return ["All", ...brands];
}

// Display all available filter options
function showAvailableFilters() {
  console.log("=== AVAILABLE FILTERS ===");
  console.log("Categories:", getUniqueCategories().join(", "));
  console.log("Brands:", getUniqueBrands().join(", "));
  console.log("============================");
}

// Load mock “frontend” initialization
function initCatalog() {
  console.log("Initializing Product Catalog System...");
  showAvailableFilters();
  applyFilters();
}

// Developer Testing Console Menu Simulation
function showConsoleMenu() {
  console.log(`
  ==== Product Catalog Menu ====
  1. Show All Products
  2. Filter by Category
  3. Filter by Brand
  4. Set Price Range
  5. Search Product
  6. Sort Products
  7. Reset Filters
  8. Exit
  ==============================
  `);
}

// Mock interaction simulation
function mockConsoleInteraction() {
  initCatalog();
  showConsoleMenu();
  // Simulate few user actions
  setCategory("Headphones");
  setBrand("Sony");
  setPriceRange(2000, 35000);
  setSort("ratingHighLow");
  setSearch("Sony");
  resetFilters();

  console.log("\n--- Simulated User Actions Complete ---");
}

// Run the simulation
simulateUserActions();
// mockConsoleInteraction(); // Alternate simulation
```