**Name :Nikhil Sharma**

**UID: 23BCC70030**

**Class: 23BCC-1(A)**

FULL STACK

EXPERIMENT-2.2

**AIM:**

To design and implement a dynamic product filtering system using JavaScript DOM Manipulation, where products can be filtered based on category, price, rating, and search text with dropdowns and input fields.

**THEORY:**

1. **DOM Manipulation** – JavaScript allows selecting, creating, and updating HTML elements dynamically using methods like querySelector, appendChild, and innerHTML.
2. **Events** – Filters trigger events (change, input, click), which update the product list in real time.
3. **Filtering Logic** – An array of product objects is iterated, applying conditions based on selected filters.
4. **Sorting** – Products can be sorted (by price, rating, or name) using array sorting methods.
5. **Output Rendering** – Matching products are displayed as **cards**. If no product matches, an empty state message is shown.

This simulates a **real-world e-commerce filter system** like Amazon, Flipkart, etc.

**PROCEDURE:**

1. Create Dataset – Define an array of products with properties (id, name, category, price, rating, tags).
2. UI Design – Add dropdowns for category, price, rating, and input for search.
3. Event Binding – Attach listeners to dropdowns and input fields.
4. Filtering Function – Apply multiple conditions on the product array.
5. DOM Update – Render filtered products as cards in a grid.
6. Reset Function – Restore all products when filters are cleared.

**CODE:**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Dynamic Product Filter</title>

<style>

body{font-family:Arial,sans-serif;background:#f6f8fa;margin:0;padding:20px;}

h2{text-align:center}

.toolbar{margin-bottom:20px;display:flex;gap:10px;flex-wrap:wrap}

select,input{padding:8px;border-radius:6px;border:1px solid #ccc}

button{padding:8px 14px;border:none;border-radius:6px;background:#007bff;color:white;cursor:pointer}

.cards{display:grid;grid-template-columns:repeat(auto-fill,minmax(200px,1fr));gap:14px}

.card{border:1px solid #ccc;border-radius:10px;padding:12px;background:white}

.title{font-weight:bold}

.price{color:green}

.empty{text-align:center;color:#888;margin-top:20px}

</style>

</head>

<body>

<h2>Dynamic Product Filter</h2>

<!-- Filter Controls -->

<div class="toolbar">

<select id="category">

<option value="all">All Categories</option>

<option>Electronics</option>

<option>Clothing</option>

<option>Home</option>

<option>Books</option>

</select>

<select id="price">

<option value="all">All Prices</option>

<option value="0-1000">₹0–₹1000</option>

<option value="1000-5000">₹1000–₹5000</option>

<option value="5000-20000">₹5000–₹20000</option>

</select>

<select id="rating">

<option value="0">Any Rating</option>

<option value="3">3★ & above</option>

<option value="4">4★ & above</option>

</select>

<input type="text" id="search" placeholder="Search product...">

<button id="reset">Reset</button>

</div>

<!-- Products Display -->

<div class="cards" id="cards"></div>

<div class="empty" id="empty" style="display:none">No products found!</div>

<script>

const products = [

{id:1,name:"Wireless Earbuds",category:"Electronics",price:2499,rating:4.3},

{id:2,name:"Smart TV 43\"",category:"Electronics",price:22999,rating:4.6},

{id:3,name:"Cotton T-Shirt",category:"Clothing",price:799,rating:4.1},

{id:4,name:"Running Shoes",category:"Clothing",price:3999,rating:4.7},

{id:5,name:"Steel Kettle",category:"Home",price:1299,rating:4.2},

{id:6,name:"Foam Pillow",category:"Home",price:1599,rating:4.4},

{id:7,name:"The Pragmatic Programmer",category:"Books",price:999,rating:4.8},

{id:8,name:"Data Structures in JS",category:"Books",price:649,rating:4.5}

];

const UI = {

category: document.querySelector('#category'),

price: document.querySelector('#price'),

rating: document.querySelector('#rating'),

search: document.querySelector('#search'),

reset: document.querySelector('#reset'),

cards: document.querySelector('#cards'),

empty: document.querySelector('#empty')

};

// Render Function

function render(list){

UI.cards.innerHTML = "";

if(list.length === 0){

UI.empty.style.display = "block";

return;

}

UI.empty.style.display = "none";

list.forEach(p=>{

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

card.className = 'card';

card.innerHTML = `<div class='title'>${p.name}</div>

<div>${p.category}</div>

<div class='price'>₹${p.price}</div>

<div>⭐ ${p.rating}</div>`;

UI.cards.appendChild(card);

});

}

// Filter Function

function filterProducts(){

let list = products.filter(p=>{

let cat = UI.category.value==="all" || p.category===UI.category.value;

let pr = true;

if(UI.price.value!=="all"){

const [min,max] = UI.price.value.split('-').map(Number);

pr = p.price>=min && p.price<=max;

}

let rt = p.rating >= Number(UI.rating.value);

let sr = p.name.toLowerCase().includes(UI.search.value.toLowerCase());

return cat && pr && rt && sr;

});

render(list);

}

// Event Listeners

[UI.category,UI.price,UI.rating,UI.search].forEach(el=>{

el.addEventListener('input',filterProducts);

});

UI.reset.addEventListener('click',()=>{

UI.category.value='all';UI.price.value='all';UI.rating.value='0';UI.search.value='';

render(products);

});

// Initial render

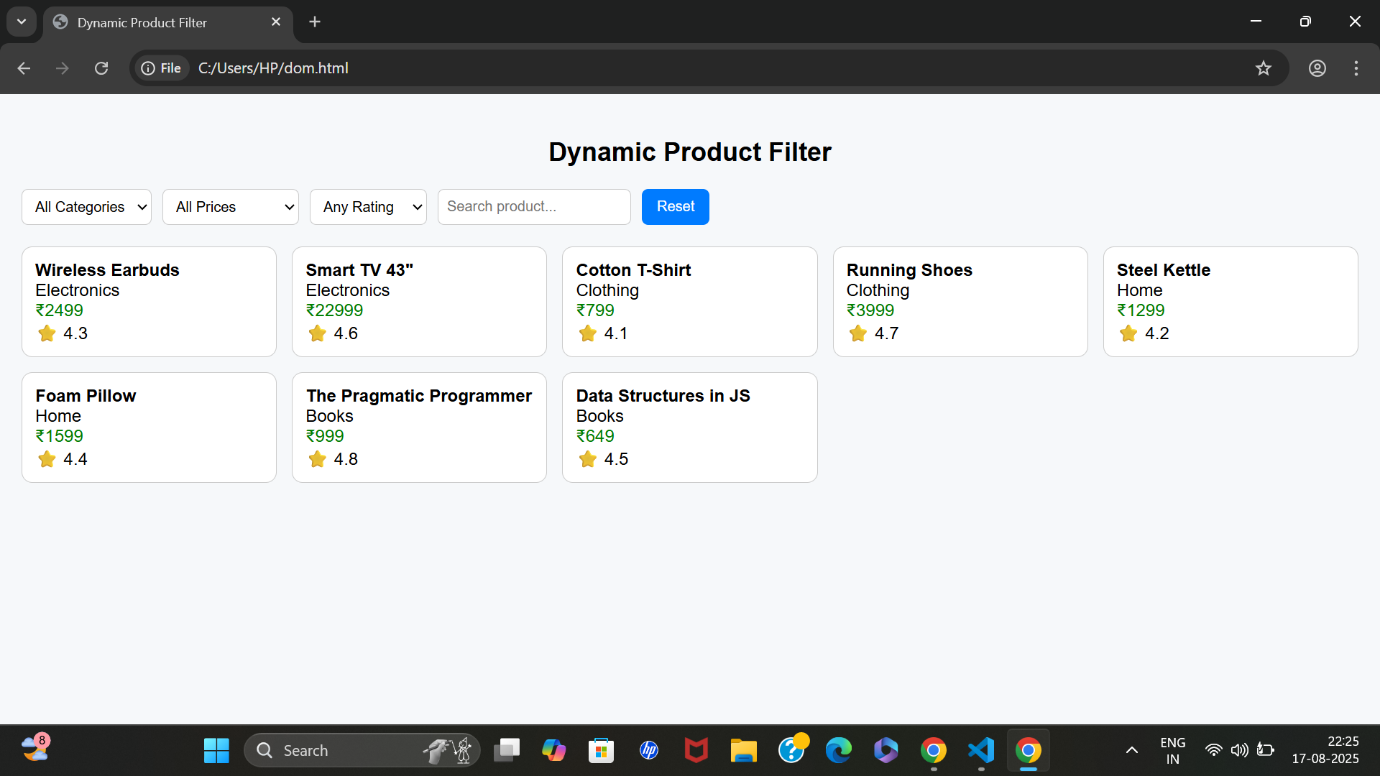
render(products);

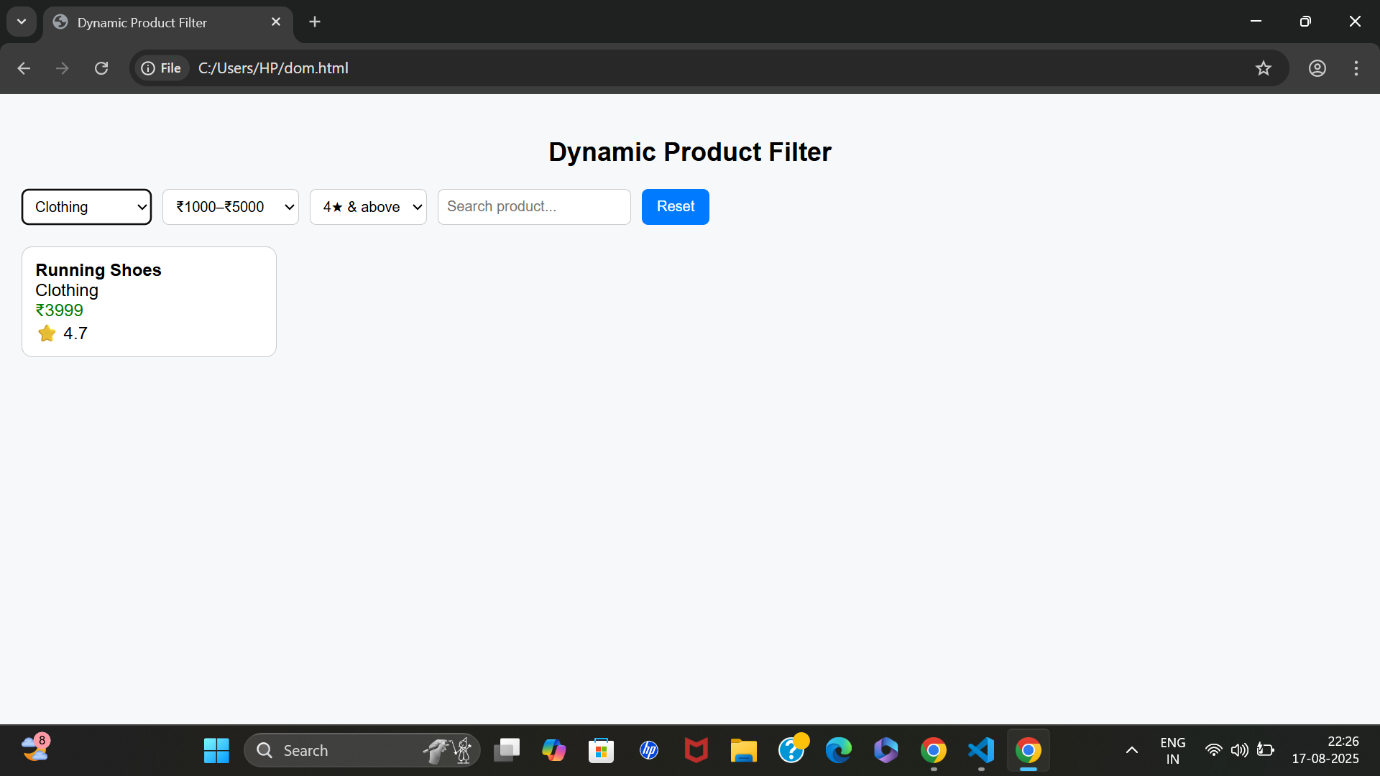
</script>

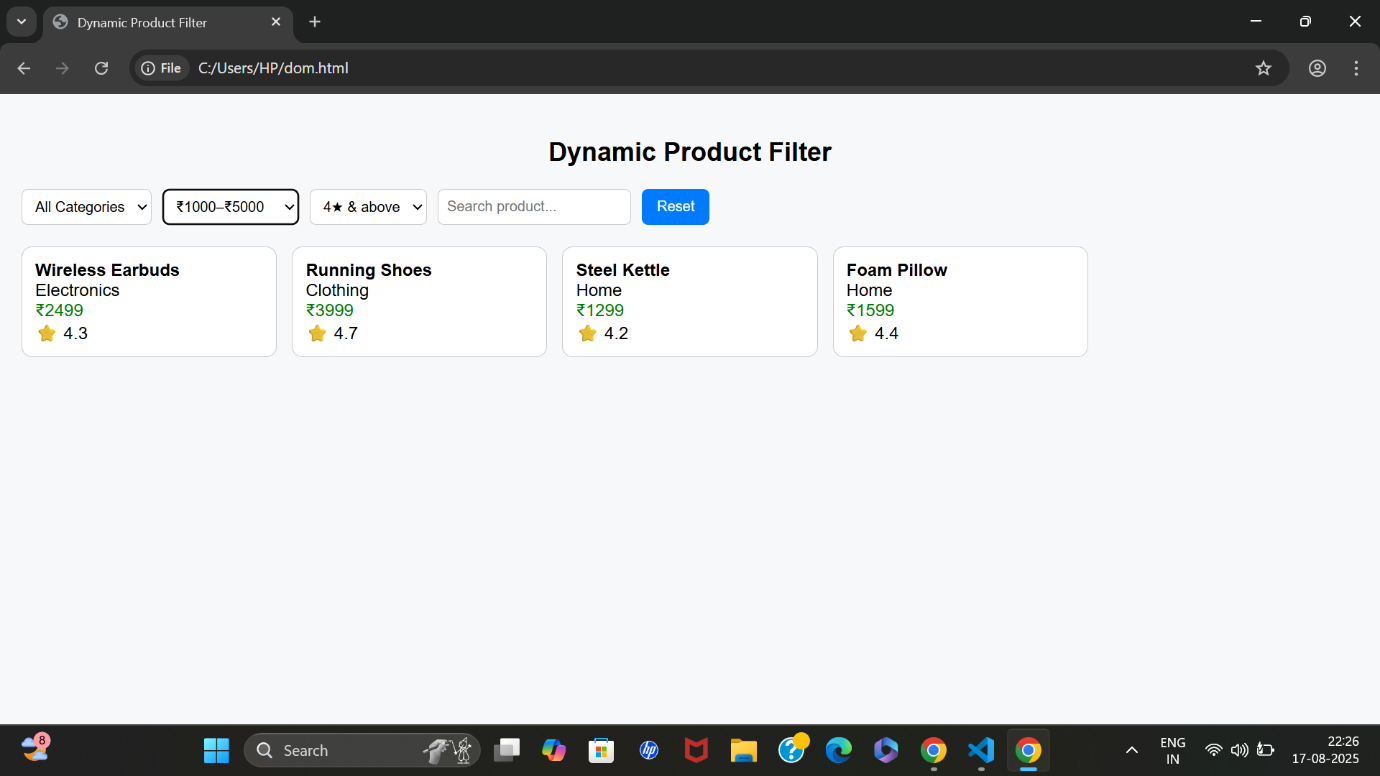
</body>

</html>

**OUTPUT:**

****

****

****

**LEARNING OUTCOME:**

* At first, all products are displayed as cards.
* Selecting category = Electronics shows only electronics.
* Filtering by price = 0–1000 narrows results further.
* Searching "js" shows the *Data Structures in JS* book.
* If no product matches → "No products found!" message appears.