# EXP-4

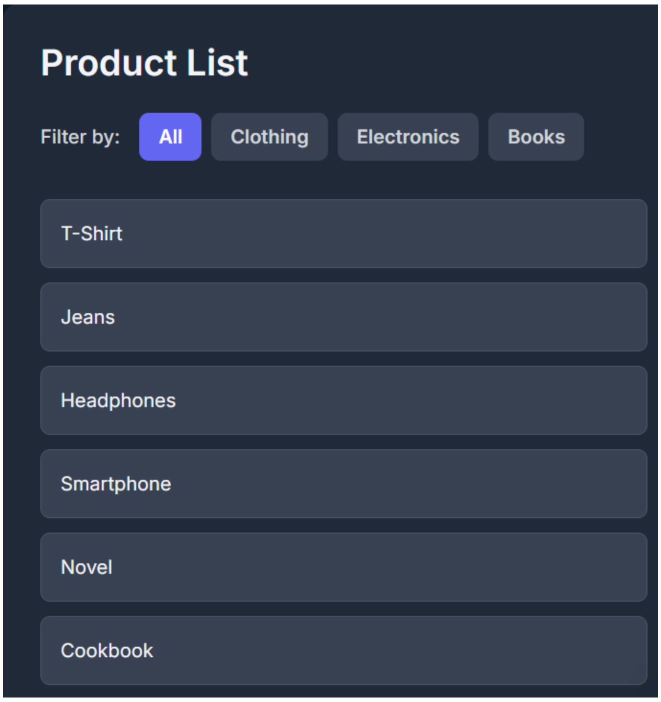
## Dynamic Product Filter with Dropdown using JavaScript DOM Manipulation:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Dynamic Product Filter</title>

  <!-- Tailwind CSS for styling -->

  <script src="https://cdn.tailwindcss.com"></script>

  <!-- Google Fonts for a clean, modern look -->

  <link rel="preconnect" href="https://fonts.googleapis.com">

  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600&display=swap"
rel="stylesheet">

  <style>

    /* Use the Inter font family */

    body {

      font-family: 'Inter', sans-serif;

    }

    /* Simple transition for product items */

    .product-item {

      transition: opacity 0.3s ease-in-out, transform 0.3s ease-in-out;

    }

  </style>

</head>

<body class="bg-gray-900 text-gray-200 flex items-center justify-center min-h-screen p-4">

  <!-- Main container card -->

  <div class="w-full max-w-2xl p-8 bg-gray-800 rounded-2xl shadow-lg">

    <!-- Header Section -->
    <div class="mb-6">
      <h1 class="text-3xl font-bold text-gray-100">Product List</h1>
    </div>


    <!-- Filter Section -->
    <div class="mb-8 flex items-center space-x-2">
      <label class="text-gray-300 font-medium mr-2">Filter by:</label>
      <div id="category-buttons" class="flex space-x-2">
        <button data-category="all" class="filter-btn bg-indigo-500 text-white font-semibold py-2 px-4 rounded-lg hover:bg-indigo-600 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-offset-gray-800 focus:ring-indigo-500 transition duration-300">All</button>
        <button data-category="clothing" class="filter-btn bg-gray-700 text-gray-300 font-semibold py-2 px-4 rounded-lg hover:bg-gray-600 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-offset-gray-800 focus:ring-indigo-500 transition duration-300">Clothing</button>
        <button data-category="electronics" class="filter-btn bg-gray-700 text-gray-300 font-semibold py-2 px-4 rounded-lg hover:bg-gray-600 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-offset-gray-800 focus:ring-indigo-500 transition duration-300">Electronics</button>
        <button data-category="books" class="filter-btn bg-gray-700 text-gray-300 font-semibold py-2 px-4 rounded-lg hover:bg-gray-600 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-offset-gray-800 focus:ring-indigo-500 transition duration-300">Books</button>
      </div>
    </div>


    <!-- Product List -->
    <div id="product-list" class="space-y-3">
      <!-- Products will be dynamically inserted here -->
    </div>
  </div>


  <script>
    // --- DATA ---
```

```
const products = [
  { name: 'T-Shirt', category: 'clothing' },
  { name: 'Jeans', category: 'clothing' },
  { name: 'Headphones', category: 'electronics' },
  { name: 'Smartphone', category: 'electronics' },
  { name: 'Novel', category: 'books' },
  { name: 'Cookbook', category: 'books' }
];

// --- DOM ELEMENTS ---
const categoryButtonsContainer = document.getElementById('category-buttons');
const productList = document.getElementById('product-list');

// --- FUNCTIONS ---

/**
 * Renders a list of products to the DOM.
 * @param {Array<Object>} productsToDisplay - The array of product objects to display.
 */
function renderProducts(productsToDisplay) {
  productList.innerHTML = '';
  if (productsToDisplay.length === 0) {
    productList.innerHTML = `<p class="text-gray-400">No products found in this category.</p>`;
    return;
  }
  productsToDisplay.forEach(product => {
    const productElement = document.createElement('div');
    productElement.className = 'product-item p-4 border border-gray-600 rounded-lg bg-gray-700 text-gray-100';
    productElement.textContent = product.name;
```

```
        productElement.dataset.category = product.category;
        productList.appendChild(productElement);
    });
}

/**
 * Filters products based on the selected category button.
 * @param {string} selectedCategory - The category to filter by.
 */
function filterProducts(selectedCategory) {
    const filteredProducts = (selectedCategory === 'all')
        ? products
        : products.filter(product => product.category === selectedCategory);
    renderProducts(filteredProducts);
}

/**
 * Handles click events on the category buttons.
 * @param {Event} event - The click event object.
 */
function handleCategoryClick(event) {
    // Check if a filter button was clicked
    if (!event.target.classList.contains('filter-btn')) {
        return;
    }

    // Remove active styles from all buttons
    const buttons = categoryButtonsContainer.querySelectorAll('.filter-btn');
    buttons.forEach(button => {
        button.classList.remove('bg-indigo-500', 'text-white');
        button.classList.add('bg-gray-700', 'text-gray-300');
      });

      // Add active styles to the clicked button
      const clickedButton = event.target;
      clickedButton.classList.add('bg-indigo-500', 'text-white');
      clickedButton.classList.remove('bg-gray-700', 'text-gray-300');

      // Filter the products
      const selectedCategory = clickedButton.dataset.category;
      filterProducts(selectedCategory);
    }

    // --- EVENT LISTENERS ---
    categoryButtonsContainer.addEventListener('click', handleCategoryClick);

    // --- INITIAL RENDER ---
    // Initially display all products when the page loads.
    filterProducts('all');
  </script>
</body>
</html>
```

### OUTPUTS:


Product List

T-Shirt

Jeans

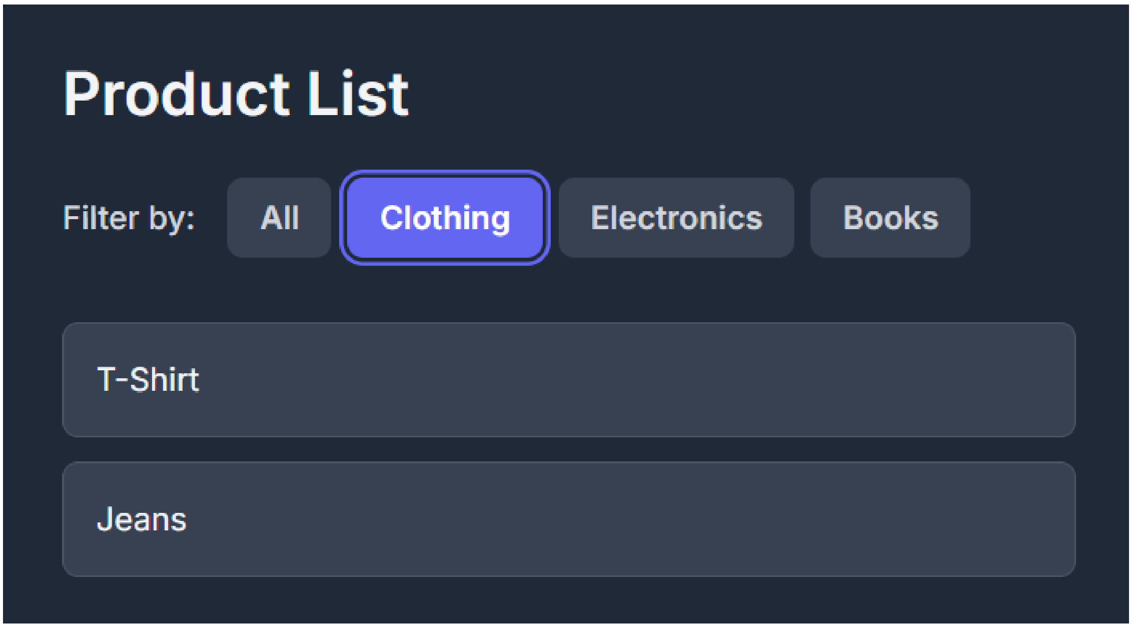
Headphones

Smartphone

Novel

Cookbook

*Figure 1 Product list*


Product List

T-Shirt

Jeans

*Figure 2 Using filters*