EAD LAB INTERNAL 1 160122729015

SET: 9 AIML(J) VI SEM

AIM: E-Commerce Product Catalog (ReactJS + Bootstrap)

-Build a product listing page for an e-commerce site using ReactJS and Bootstrap Grid.

-Include product filtering, sorting, and a responsive UI

DESCRPTION:

ReactJS

- ReactJS is a popular JavaScript library for building dynamic and interactive user interfaces.
- Role in the Project:
    - Used to create reusable components like ProductList, filters, and sorting options.
    - Manages the state of filters, sorting, and product data dynamically.
    - Enables a single-page application (SPA) experience with smooth updates to the UI.

Bootstrap

- Bootstrap is a front-end framework for designing responsive and mobile-first websites.
- Role in the Project:
    - Provides a responsive grid system to organize product cards into rows and columns.
    - Ensures the UI is mobile-friendly and adapts to different screen sizes.
    - Offers pre-built components like cards, forms, and buttons for a clean and consistent design.

Features of the Project

1. Product Listing:
    - Displays products in a 3x3 grid layout using Bootstrap's grid system.
    - Each product card includes an image, name, price, discount, and rating.
2. Filtering:
    - Category: Checkboxes to filter products by category (e.g., Electronics, Clothing).
    - Color: Checkboxes with color boxes to filter products by color.
    - Price Range: A slider to filter products within a specific price range.
    - Discount: Radio buttons to filter products by discount percentage.
3. Sorting:

   - Dropdown to sort products by:
      - Price (Low to High or High to Low)
      - Rating (High to Low)
      - Discount (High to Low)

4. Responsive UI:
   - The layout adjusts seamlessly for desktop, tablet, and mobile views.
   - Filters and product cards stack vertically on smaller screens for better usability.

PROCEDURE:

Requirements--

1. Node.js: Ensure Node.js is installed on your system. Download it from nodejs.org.
2. ReactJS: The project is built using ReactJS, so you need to set up a React environment.
3. Bootstrap: Used for styling and responsive design.

Installation Steps--

1. Install Node.js:
   - Download and install Node.js from nodejs.org.
   - Verify the installation by running:

     ```
     node -v

     npm -v
     ```

2. Create a React App:
   - Use create-react-app to set up a new React project:

     ```
     npx create-react-app ecommerce-product-catalog

     cd ecommerce-product-catalog
     ```

3. Install Dependencies:
   - Install Bootstrap for styling:

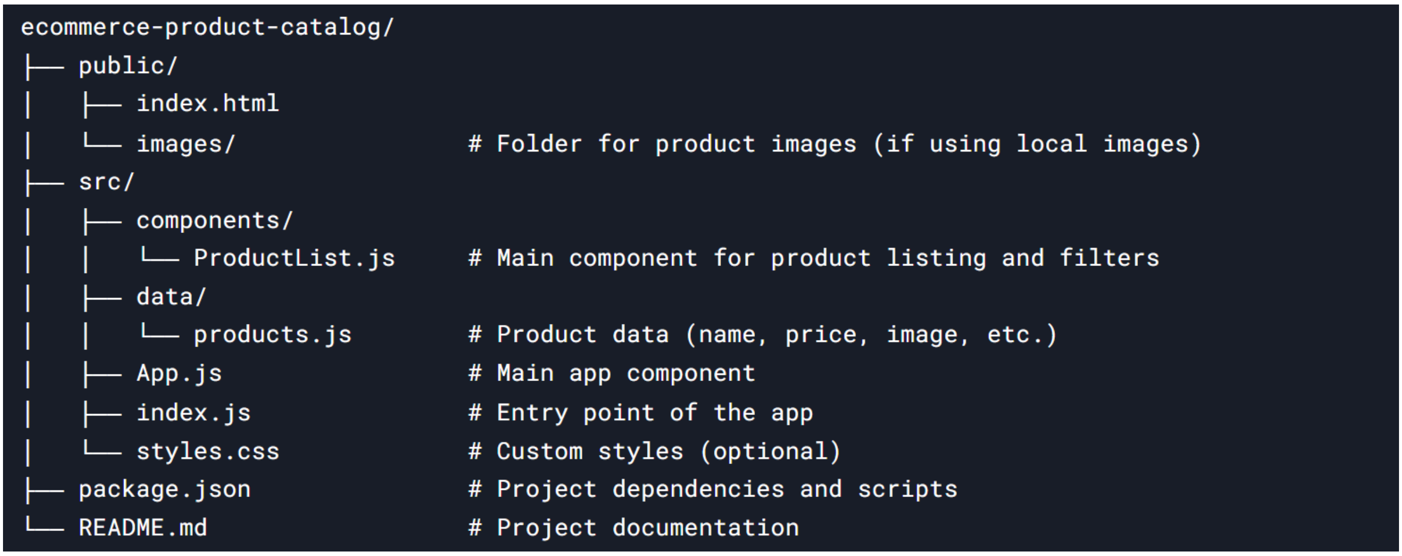
     ```
     npm install bootstrap
     ```

   - Install react-range for the price range slider:

     ```
     npm install react-range
     ```

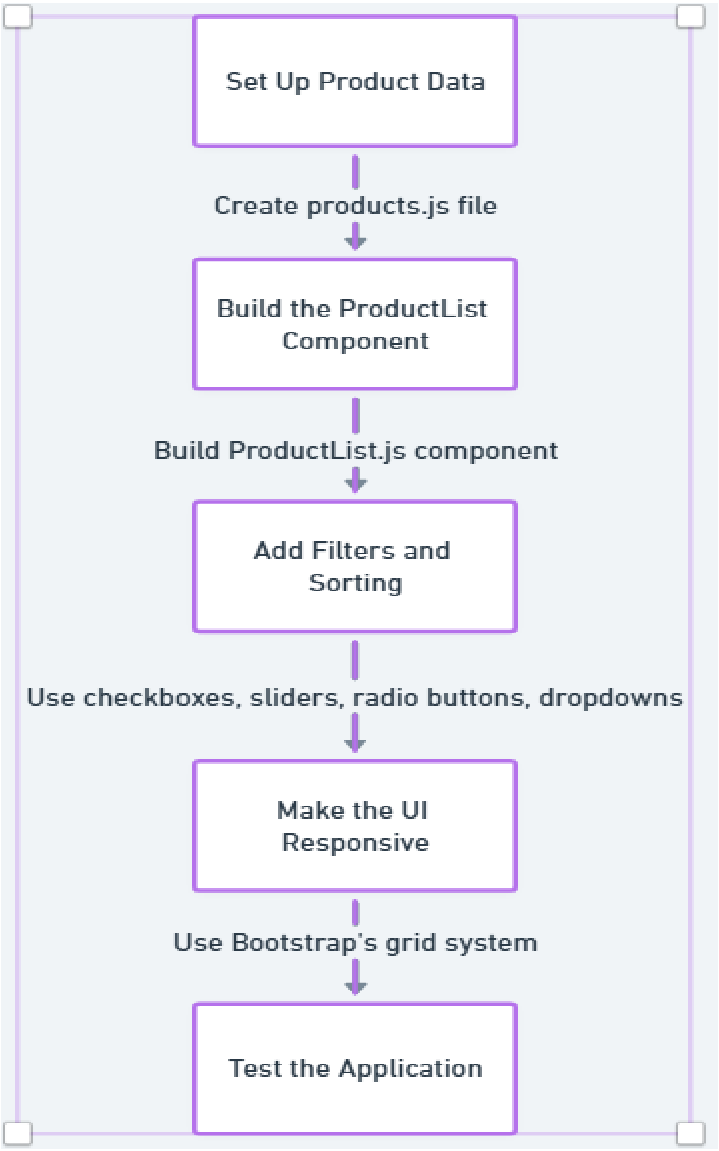
4. Start the Development Server:
   - Run the app in development mode: npm start

File Structure



```
ecommerce-product-catalog/
├── public/
│   ├── index.html
│   └── images/              # Folder for product images (if using local images)
├── src/
│   ├── components/
│   │   └── ProductList.js   # Main component for product listing and filters
│   ├── data/
│   │   └── products.js      # Product data (name, price, image, etc.)
│   ├── App.js               # Main app component
│   ├── index.js             # Entry point of the app
│   └── styles.css           # Custom styles (optional)
├── package.json             # Project dependencies and scripts
└── README.md                # Project documentation
```

PROJECT FLOW:


Set Up Product Data
Create products.js file
Build the ProductList Component
Build ProductList.js component
Add Filters and Sorting
Use checkboxes, sliders, radio buttons, dropdowns
Make the UI Responsive
Use Bootstrap's grid system
Test the Application

<u>CODE:</u>

public/index.html - This is the main HTML file where the React app is mounted.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>E-Commerce Product Catalog</title>
</head>
<body>
  <div id="root"></div>
</body>
</html>
```

src/App.js - Update App.js to render the ProductList component:

```
import React from "react";
import ProductList from "./components/ProductList";

function App() {
  return (
    <div className="App">
      <ProductList />
    </div>
  );
}
export default App;
```

src/index.js - This is the entry point for the React application.

```
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App";
import "bootstrap/dist/css/bootstrap.min.css";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

src/styles.css - add custom styles if needed.

```css
.card {
  height: 400px;
  border: 1px solid #ddd;
  border-radius: 8px;
  overflow: hidden;
  transition: transform 0.2s;
  display: flex;
  flex-direction: column;
}

.card:hover {
  transform: scale(1.05);
}

.card-img-top {
  width: 100%;
  height: 200px;
  object-fit: cover;

.card-body {
  padding: 1rem;
  flex: 1;
  overflow-y: auto;
}

.card-title {
  font-size: 1.25rem;
  font-weight: bold;
  margin-bottom: 0.5rem;
}

.card-text {
  margin-bottom: 0.5rem;
}
@media (max-width: 768px) {
  .col-md-4 {
    flex: 0 0 100%;
    max-width: 100%;
  }
}
```

src/data/products.js - Create a products.js file in the src folder to store product data:

```
const products = [
  {
    id: 1,
    name: "Smartphone",
    price: 299.99,
    category: "Electronics",
    color: "Black",
    discount: 10,
    rating: 4.5,
    image: "https://images.unsplash.com/photo-1598327105666-5b89351aff97?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  },
  {
    id: 2,
    name: "T-Shirt",
    price: 19.99,
    category: "Clothing",
    color: "White",
    discount: 20,
    rating: 3.8,
    image: "https://images.unsplash.com/photo-1581655353564-df123a1eb820?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  },
  {
    id: 3,
    name: "K-Swiss Shoes",
    price: 49.99,
    category: "Shoes",
    color: "White",
    discount: 15,
    rating: 4.2,
    image: "https://images.unsplash.com/photo-1595341888016-a392ef81b7de?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  },
  {
    id: 4,
    name: "Laptop",
    price: 999.99,
    category: "Electronics",
    color: "Gray",
    discount: 5,
    rating: 4.7,
    image: "https://images.unsplash.com/photo-1496181133206-80ce9b88a853?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  },
  {
    id: 5,
    name: "Jeans",
    price: 39.99,
    category: "Clothing",
    color: "Black",
    discount: 25,
    rating: 4.0,
    image: "https://images.unsplash.com/photo-1541099649105-f69ad21f3246?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  },
  {
    id: 6,
    name: "Headphones",
    price: 99.99,
    category: "Electronics",
    color: "Black",
    discount: 10,
    rating: 4.6,
    image: "https://images.unsplash.com/photo-1505740420928-5e560c06d30e?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  },
  {
    id: 7,
    name: "Nike Sneakers",
    price: 59.99,
    category: "Shoes",
    color: "Red",
    discount: 30,
    rating: 4.3,
    image: "https://images.unsplash.com/photo-1542291026-7eec264c27ff?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  },
  {
    id: 8,
    name: "T-Shirt",
    price: 79.99,
    category: "Clothing",
    color: "Blue",
    discount: 10,
    rating: 4.1,
    image: "https://m.media-amazon.com/images/I/71JPK6aUzeL._SX679_.jpg",
  },
  {
    id: 9,
    name: "Watch",
    price: 199.99,
    category: "Electronics",
    color: "Silver",
    discount: 15,
    rating: 4.4,
    image: "https://images.unsplash.com/photo-1523275335684-37898b6baf30?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  },
];

export default products;
```

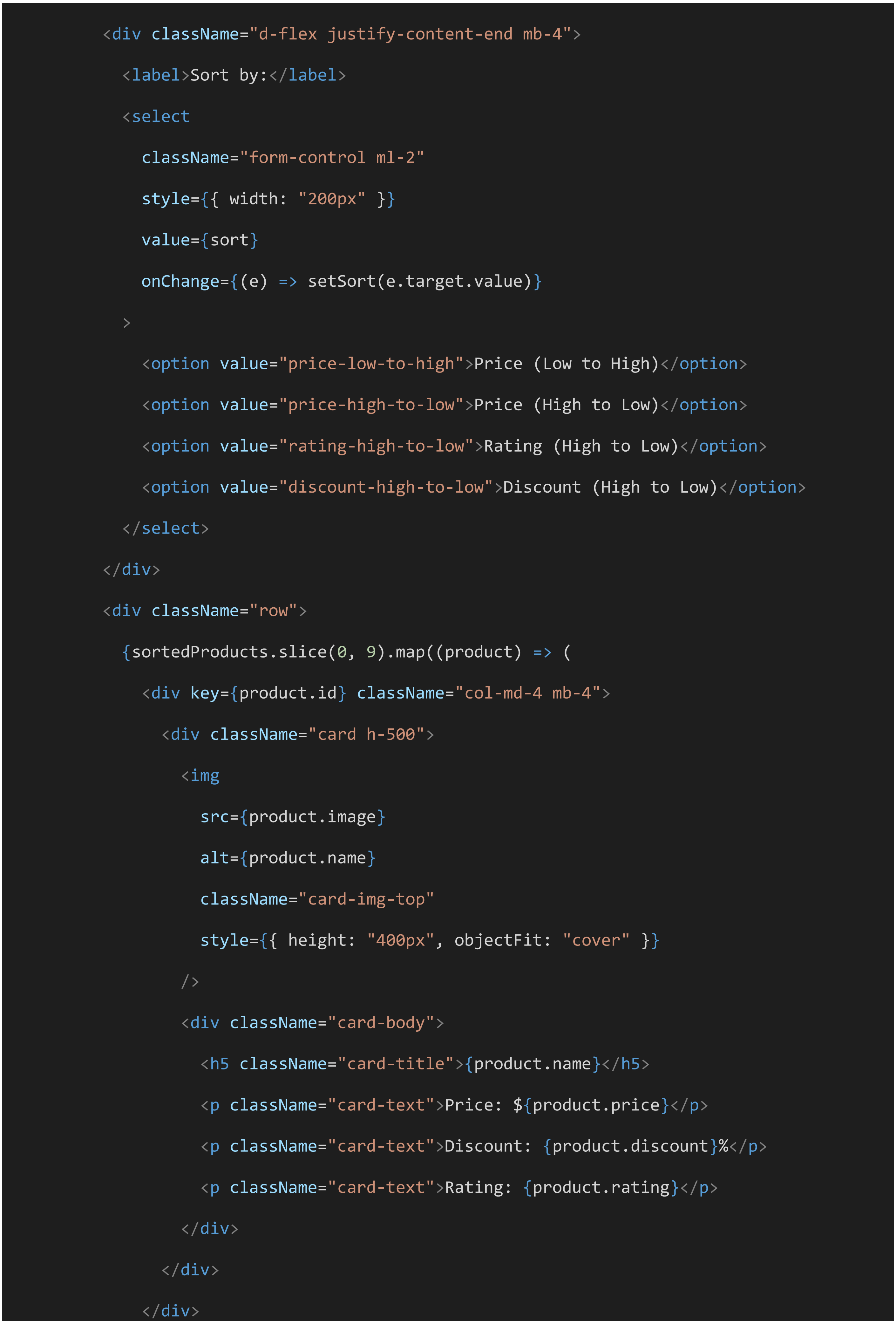
src/components/ProductList.js - Create a ProductList.js component to display the products:

```
import React, { useState } from "react";
import { Range } from "react-range";
import products from "../data/products";
import "bootstrap/dist/css/bootstrap.min.css";

const ProductList = () => {
  const [filter, setFilter] = useState({
    categories: [],
    colors: [],
    price: [0, 1000],
    discount: "All",
  });
  const [sort, setSort] = useState("price-low-to-high");
  const categories = ["Electronics", "Clothing", "Shoes"];
  const colors = ["Black", "Blue", "Silver", "Gray", "White", "Red"];
  const discounts = ["All", "10% or more", "15% or more", "20% or more", "25% or more",
"30% or more"];


  const filteredProducts = products.filter((product) => {
    return (
      (filter.categories.length === 0 || filter.categories.includes(product.category)) &&
      (filter.colors.length === 0 || filter.colors.includes(product.color)) &&
      product.price >= filter.price[0] &&
      product.price <= filter.price[1] &&
      (filter.discount === "All" ||
        (filter.discount === "10% or more" && product.discount >= 10) ||
        (filter.discount === "15% or more" && product.discount >= 15) ||
        (filter.discount === "20% or more" && product.discount >= 20) ||
        (filter.discount === "25% or more" && product.discount >= 25) ||
        (filter.discount === "30% or more" && product.discount >= 30))
    );
  });


  const sortedProducts = filteredProducts.sort((a, b) => {
    if (sort === "price-low-to-high") return a.price - b.price;
    if (sort === "price-high-to-low") return b.price - a.price;
    if (sort === "rating-high-to-low") return b.rating - a.rating;
    if (sort === "discount-high-to-low") return b.discount - a.discount;
    return 0;
  });


  const handleCategoryChange = (category) => {
    if (filter.categories.includes(category)) {
      setFilter({ ...filter, categories: filter.categories.filter((c) => c !== category)
});
```

```
  } else {
    setFilter({ ...filter, categories: [...filter.categories, category] });
  }
};

const handleColorChange = (color) => {
  if (filter.colors.includes(color)) {
    setFilter({ ...filter, colors: filter.colors.filter((c) => c !== color) });
  } else {
    setFilter({ ...filter, colors: [...filter.colors, color] });
  }
};

return (
  <div className="container-fluid mt-4 p-4" style={{ padding: "20px" }}>
    <h1 className="text-center mb-4">Ecommerce Product Catalog</h1>

    <div className="row">
      <div className="col-md-3">
        <div className="card p-3">
          <h5>Filters</h5>
          <div className="mb-4">
            <label>Category:</label>
            {categories.map((category) => (
              <div key={category} className="form-check">
                <input
                  type="checkbox"
                  className="form-check-input"
                  id={category}
                  checked={filter.categories.includes(category)}
                  onChange={() => handleCategoryChange(category)}
                />
```

```
                  <label className="form-check-label" htmlFor={category}>
                    {category}
                  </label>
                </div>
              ))}
            </div>

            <div className="mb-4">
              <label>Color:</label>
              {colors.map((color) => (
                <div key={color} className="form-check">
                  <input
                    type="checkbox"
                    className="form-check-input"
                    id={color}
                    checked={filter.colors.includes(color)}
                    onChange={() => handleColorChange(color)}
                  />
                  <label className="form-check-label" htmlFor={color} style={{ display: "flex", alignItems: "center" }}>
                    <div
                      style={{
                        width: "16px",
                        height: "16px",
                        backgroundColor: color.toLowerCase(),
                        marginRight: "8px",
                        border: "1px solid #ccc",
                        borderRadius: "3px",
                      }}
                    ></div>
                    {color}
                  </label>
```

```
    </div>
  ))}
</div>

<div className="mb-4">
  <label>Price Range: ${filter.price[0]} - ${filter.price[1]}</label>
  <Range
    step={10}
    min={0}
    max={1000}
    values={filter.price}
    onChange={(values) => setFilter({ ...filter, price: values })}
    renderTrack={({ props, children }) => (
      <div
        {...props}
        style={{
          ...props.style,
          height: "6px",
          width: "100%",
          backgroundColor: "#ddd",
          borderRadius: "3px",
        }}
      >
        {children}
      </div>
    )}
    renderThumb={({ props }) => (
      <div
        {...props}
        style={{
          ...props.style,
          height: "20px",
                      width: "20px",
                      backgroundColor: "#007bff",
                      borderRadius: "50%",
                    }}
                  />
                )}
              />
            </div>

            <div className="mb-3">
              <label>Discount:</label>
              {discounts.map((discount) => (
                <div key={discount} className="form-check">
                  <input
                    type="radio"
                    className="form-check-input"
                    id={discount}
                    name="discount"
                    value={discount}
                    checked={filter.discount === discount}
                    onChange={(e) => setFilter({ ...filter, discount: e.target.value })}
                  />
                  <label className="form-check-label" htmlFor={discount}>
                    {discount}
                  </label>
                </div>
              ))}
            </div>
          </div>
        </div>

        <div className="col-md-9">
```



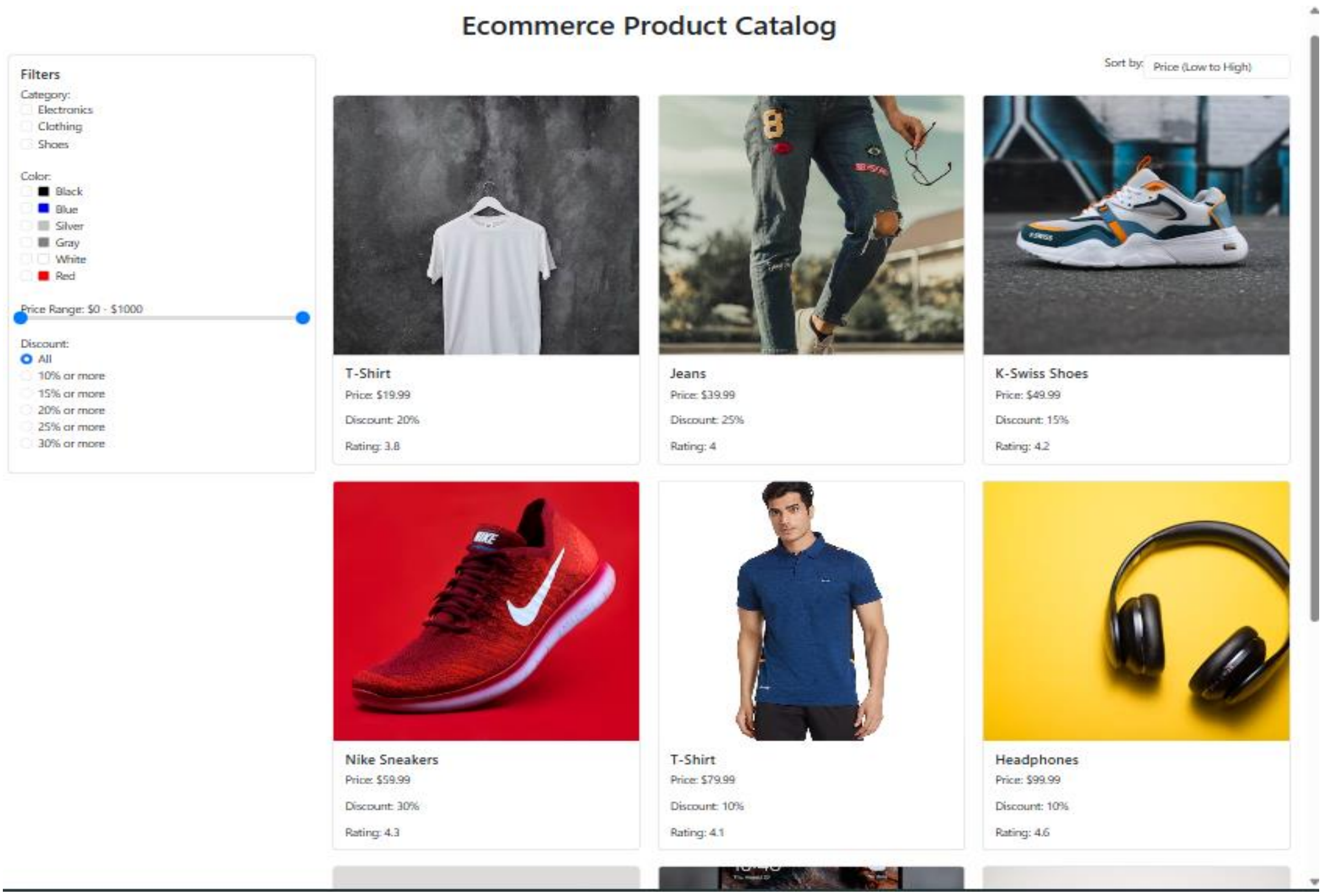
```
          <div className="d-flex justify-content-end mb-4">
            <label>Sort by:</label>
            <select
              className="form-control ml-2"
              style={{ width: "200px" }}
              value={sort}
              onChange={(e) => setSort(e.target.value)}
            >
              <option value="price-low-to-high">Price (Low to High)</option>
              <option value="price-high-to-low">Price (High to Low)</option>
              <option value="rating-high-to-low">Rating (High to Low)</option>
              <option value="discount-high-to-low">Discount (High to Low)</option>
            </select>
          </div>
          <div className="row">
            {sortedProducts.slice(0, 9).map((product) => (
              <div key={product.id} className="col-md-4 mb-4">
                <div className="card h-500">
                  <img
                    src={product.image}
                    alt={product.name}
                    className="card-img-top"
                    style={{ height: "400px", objectFit: "cover" }}
                  />
                  <div className="card-body">
                    <h5 className="card-title">{product.name}</h5>
                    <p className="card-text">Price: ${product.price}</p>
                    <p className="card-text">Discount: {product.discount}%</p>
                    <p className="card-text">Rating: {product.rating}</p>
                  </div>
                </div>
              </div>
            ))}
          </div>
        </div>
      </div>
    </div>
  );
};
export default ProductList;
```

package.json - This file contains project dependencies and scripts.

```
{
  "name": "ecommerce-product-catalog",
  "version": "1.0.0",
  "private": true,
  "dependencies": {
    "bootstrap": "^5.3.0",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-range": "^1.8.9",
    "react-scripts": "5.0.1"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

OUTPUT:

1. Product Listing – Desktop View



Ecommerce Product Catalog

Filters

Category:

□ Electronics
□ Clothing
□ Shoes

Color:

□ Black
□ Blue
□ Silver
□ Gray
□ White
□ Red

Price Range: $0 - $1000

Discount:

● All
○ 10% or more
○ 15% or more
○ 20% or more
○ 25% or more
○ 30% or more

Sort by: Price (Low to High)

T-Shirt
Price: $19.99
Discount: 20%
Rating: 3.8

Jeans
Price: $39.99
Discount: 25%
Rating: 4

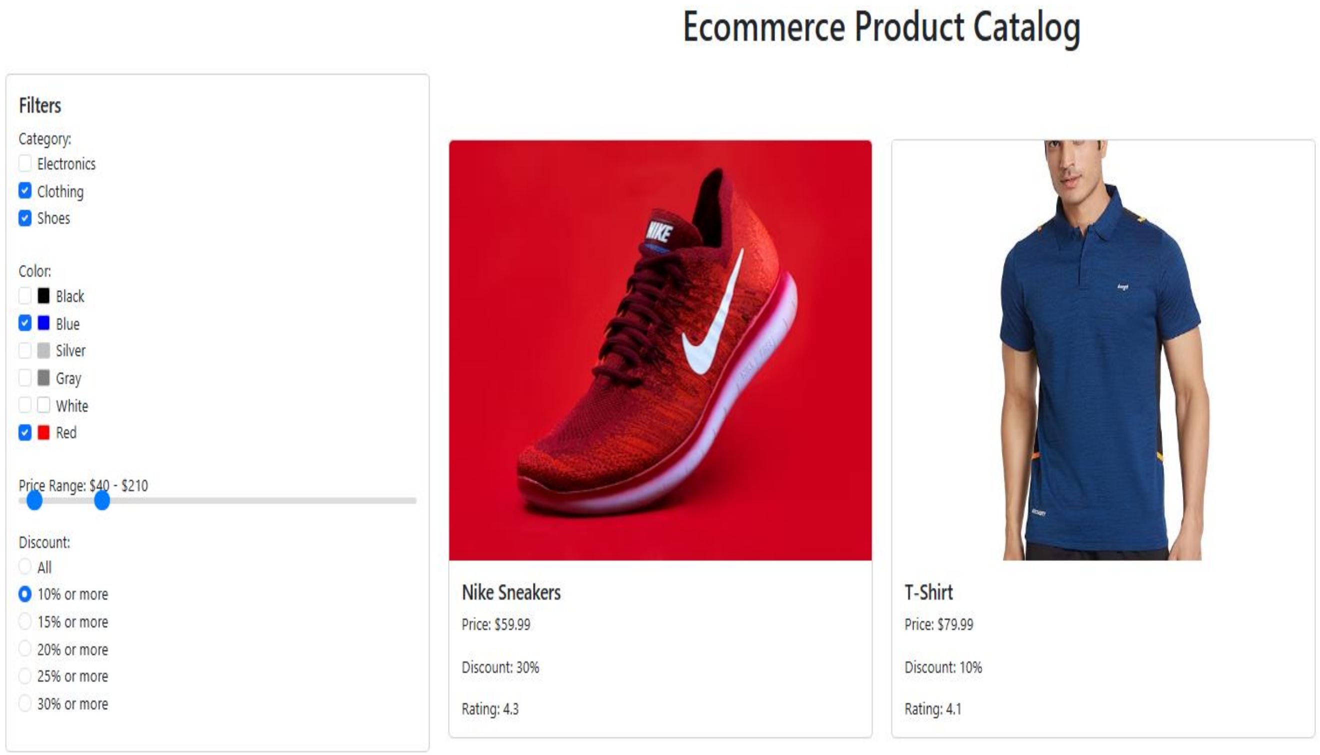
K-Swiss Shoes
Price: $49.99
Discount: 15%
Rating: 4.2

Nike Sneakers
Price: $59.99
Discount: 30%
Rating: 4.3

T-Shirt
Price: $79.99
Discount: 10%
Rating: 4.1

Headphones
Price: $99.99
Discount: 10%
Rating: 4.6

2. Filtering – Applied some filters in the image below.



Ecommerce Product Catalog

Filters

Category:

□ Electronics
☑ Clothing
☑ Shoes

Color:

□ Black
☑ Blue
□ Silver
□ Gray
□ White
☑ Red

Price Range: $40 - $210

Discount:

○ All
● 10% or more
○ 15% or more
○ 20% or more
○ 25% or more
○ 30% or more

Nike Sneakers

Price: $59.99

Discount: 30%

Rating: 4.3

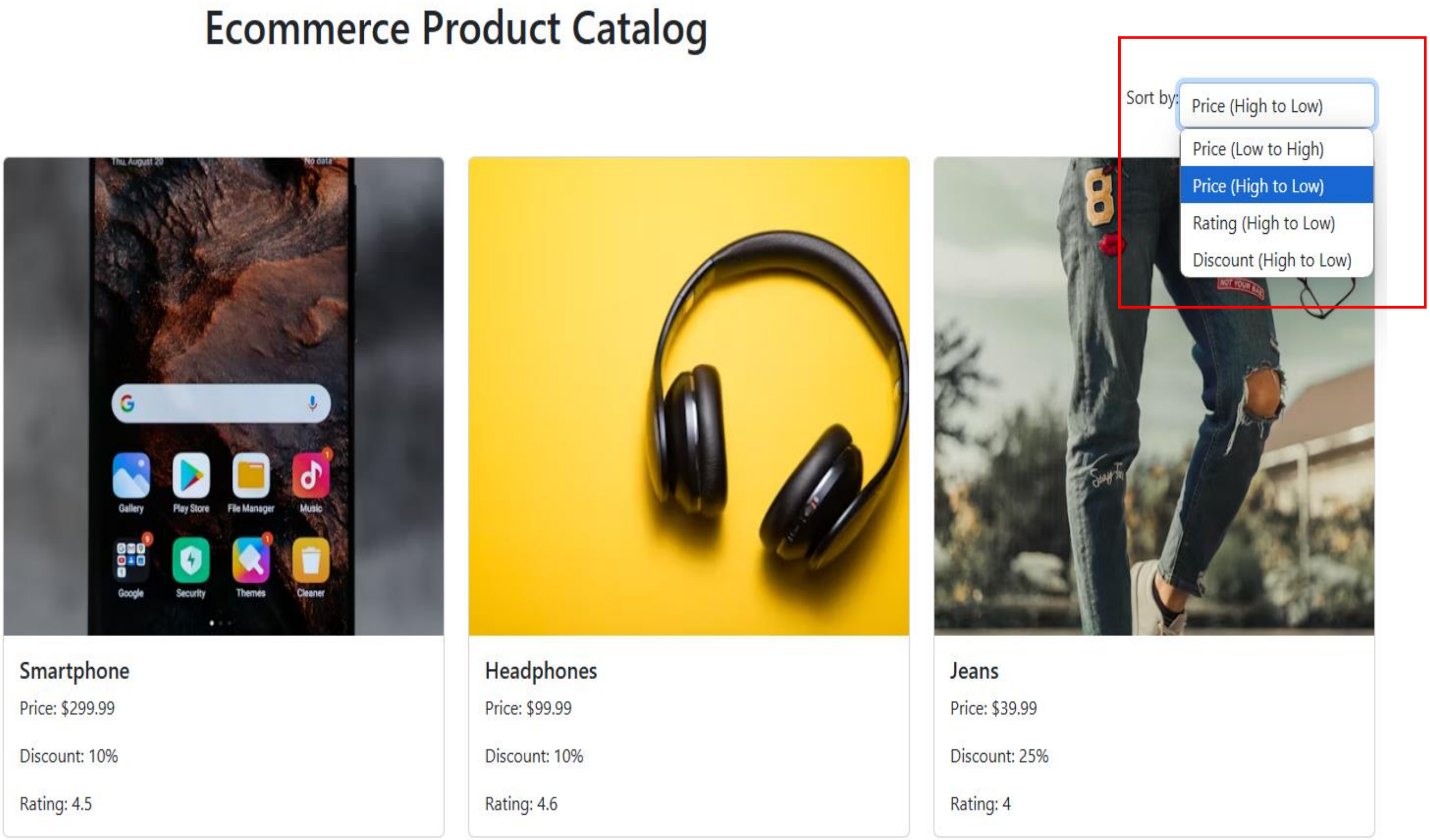
T-Shirt

Price: $79.99

Discount: 10%

Rating: 4.1

3.Sorting



Ecommerce Product Catalog

Sort by: Price (High to Low)

Smartphone

Price: $299.99

Discount: 10%

Rating: 4.5

Headphones

Price: $99.99

Discount: 10%

Rating: 4.6

Jeans

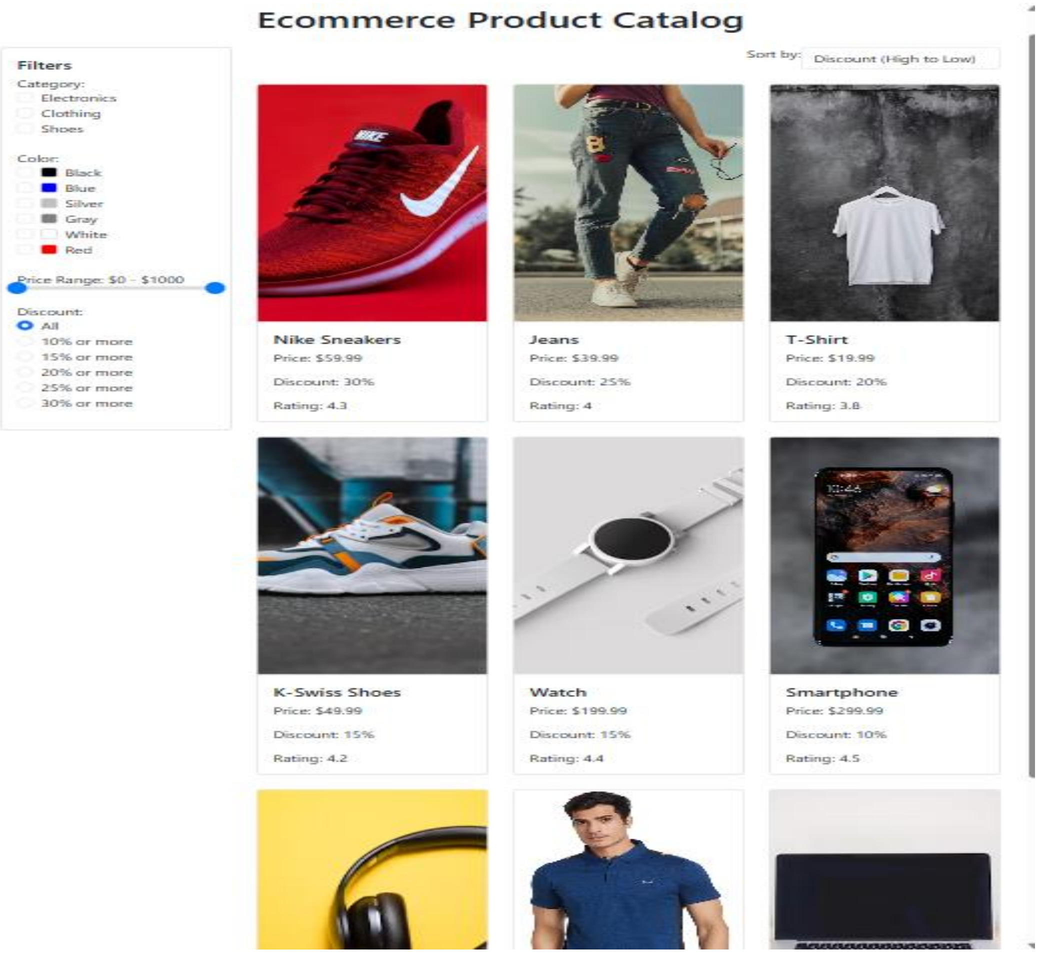
Price: $39.99

Discount: 25%

Rating: 4

4.Responsive View

TAB-



Ecommerce Product Catalog

Filters

Category:

□ Electronics

□ Clothing

□ Shoes

Color:

□ ⬛ Black

□ 🟦 Blue

□ ▨ Silver

□ ▨ Gray

□ ⬜ White

□ 🟥 Red

Price Range: $0 - $1000

Discount:

● All

○ 10% or more

○ 15% or more

○ 20% or more

○ 25% or more

○ 30% or more

Sort by: Discount (High to Low)

Nike Sneakers
Price: $59.99
Discount: 30%
Rating: 4.3

Jeans
Price: $39.99
Discount: 25%
Rating: 4

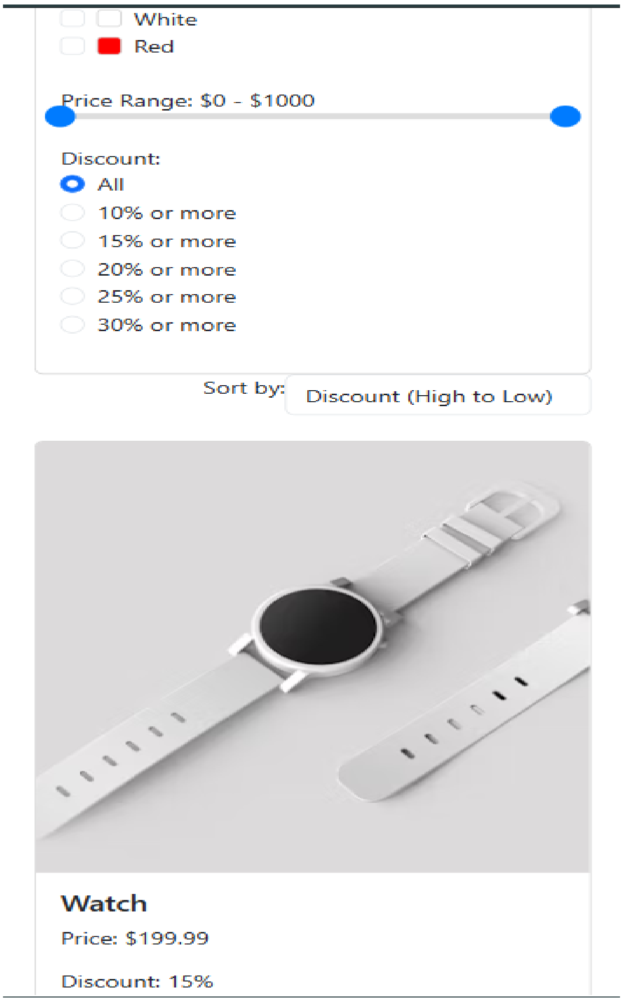
T-Shirt
Price: $19.99
Discount: 20%
Rating: 3.8

K-Swiss Shoes
Price: $49.99
Discount: 15%
Rating: 4.2

Watch
Price: $199.99
Discount: 15%
Rating: 4.4

Smartphone
Price: $299.99
Discount: 10%
Rating: 4.5

MOBILE-



□ ⬜ White

□ 🟥 Red

Price Range: $0 - $1000

Discount:

● All

○ 10% or more

○ 15% or more

○ 20% or more

○ 25% or more

○ 30% or more

Sort by: Discount (High to Low)

Watch

Price: $199.99

Discount: 15%

RESULT ANALYSIS:

1. Performance

- Result: The application performs well, with smooth updates to the UI when filters or sorting options are applied.
- Analysis:
    - React's state management ensures that only the necessary components are re-rendered, improving performance.
    - The use of functional components and hooks (e.g., useState) makes the code clean and efficient.

2. User Experience

- Result: The application provides a seamless and enjoyable user experience.
- Analysis:
    - The clean and intuitive UI makes it easy for users to browse products, apply filters, and sort results. The responsive design ensures that the application is accessible on all devices.

3. Limitations

- Result: The project has a few limitations that could be addressed in future updates.
- Analysis:
    - Static Data: The product data is hardcoded in products.js. In a real-world application, this data would be fetched from a backend API.
    - Limited Features: Additional features like pagination, search, and product details pages could enhance the application.
    - Styling: While Bootstrap provides a clean design, custom CSS could be added for a more unique look.

4. Future Enhancements

- Result: The project can be extended with additional features and improvements.
- Analysis:
    - Backend Integration: Fetch product data from a backend API instead of using static data.
    - Pagination: Add pagination to handle a large number of products.
    - Search Bar: Implement a search bar to allow users to search for specific products.
    - Product Details Page: Add a dedicated page for viewing detailed information about each product.
    - Authentication: Implement user authentication for features like wishlists and cart.