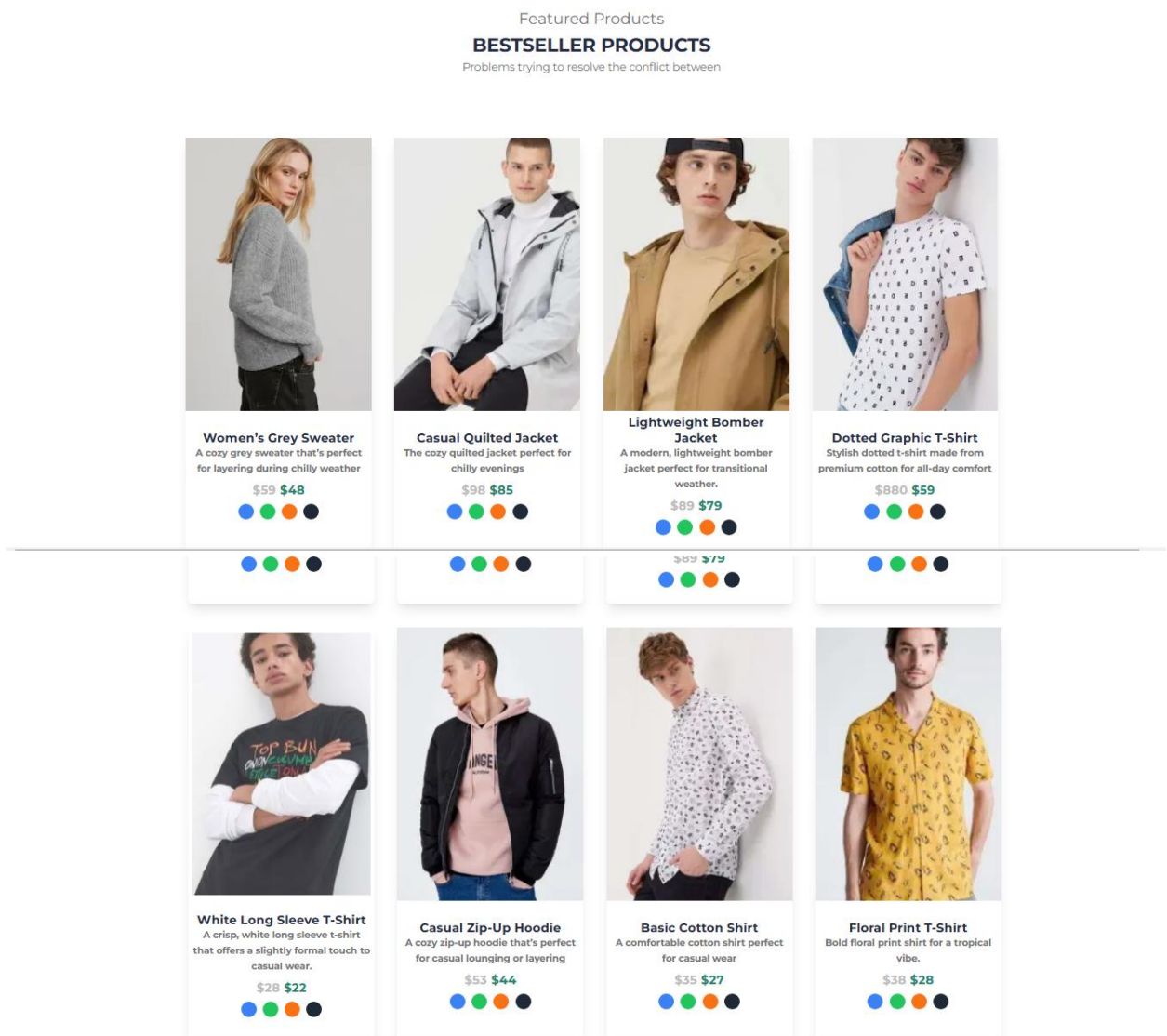


# Document Title: "Day 4 - Dynamic Frontend Components [Bandage E-commerce]"




## Functional Deliverables:

Screenshots showcasing:

1. The product listing page with dynamic data.



## 2. Individual product detail pages with accurate routing and data rendering.



### Women's Grey Sweater




★★★★★ (5/5)

**\$48**

Availability: 68

This grey sweater is crafted from soft, knitted fabric, providing warmth and comfort without compromising on style. The simple yet elegant design makes it easy to pair with skirts, trousers, or jeans. Whether worn at home or out for casual gatherings, this sweater will keep you cozy. Its versatile design and neutral color make it a timeless addition to any wardrobe.

[♥ Add to wishlist](#)[🛒 Add to Cart](#)



### Men's Black Leather Jacket

★★★★☆ (4/5)

**\$129**

Availability: 45

This black leather jacket is designed for both elegance and durability. Crafted from high-quality leather, it features a zip-up front, two side pockets, and a stylish slim fit. The soft inner lining ensures comfort, making it ideal for cool weather. Pair it with jeans or chinos for a polished, yet relaxed look that can take you from day to night.

[♥ Add to wishlist](#)[🛒 Add to Cart](#)

## 3. Working category filters, search bar.

Bandage

[Home](#) [Shop](#) [About](#) [Contact](#) [Team](#) [Pricing](#)

[Login / Register](#)



SUMMER 2020

NEW COLLECTION

We know how large objects will act,  
but things on a small scale.

women

Women's Grey Sweater

Women's Black Leather Jacket

Stylish Women's shirt

Elegant Women's jacket

Bandage

[Home](#) [Shop](#) [About](#) [Contact](#) [Team](#) [Pricing](#)

[Login / Register](#)



jacket

Casual Quilted Jacket

Lightweight Bomber Jacket

Women's Black Leather Jacket

Category

Showing all 0 results [hide](#)

Search products...

Category

## Code Deliverables:

♣ Code snippets for key components (e.g., ProductCard, ProductList, and SearchBar).

**ProductCard:** 

```
1 import Image from "next/image";
2
3 import Link from "next/link";
4 import ColorCircles from "../colors";
5 import { Tproduct } from "../../utils/componentType";
6
7
8
9
10 export default function Card(product: Tproduct) {
11   return (
12     <>
13       <div className="w-[238px] h-[530px] mx-auto">
14         <div className="w-[239px] h-full rounded-md shadow-lg">
15           <Image src={product.imageUrl} alt={product.title} priority width={200} height={250}
16             className="w-[250px] h-[350px] object-cover hover:scale-95 duration-500 hover:opacity-80" />
17           <Link href={`productList/${product.id}`}>
18             <div className={
19               `w-full h-[158px] py-[25px] px-[5px] flex flex-col items-center justify-center`>
20               <h5 className=
21                 "w-full font-bold text-[16px] leading-[20px] text-center text-[#252B42]">
22                 {product.title}
23               </h5>
24               <p className=
25                 "w-full font-bold text-[12px] leading-[20px] text-center text-[#737373]">
26                 {product.description}
27               </p>
28               <div className="w-full px-[3px] py-[5px] flex justify-center gap-[5px]">
29                 <h5 className="font-bold text-[16px] leading-[24px] text-[#BDBDBD]">
30                   ${product.price}
31                 </h5>
32                 <h5 className="font-bold text-[16px] leading-[24px] text-[#23856D]">
33                   ${product.discountedprice}
34                 </h5>
35               </div>
36               <div className="w-[82.2px] h-[16px] flex justify-center">
37                 <ColorCircles/>
38               </div>
39             </div></Link>
40           </div>
41         </div>
42       </div>
43     </>
44   );
45 }
```

**ProductList:** 

```

1  "use client"
2  import { useEffect, useState } from "react";
3  import Card from "../components/card";
4  import { client } from "@sanity/lib/client";
5  import { Tproduct } from "../../utils/componentType";
6
7
8
9
10 const Bestseller = () => {
11   const [data, setData] = useState<Tproduct[]>([]);
12
13   useEffect(() => {
14     const fetchData = async () => {
15       const result = await client.fetch(
16         `*[ _type == "product" ] {
17           id,
18           title,
19           description,
20           price,
21           discountedprice,
22           imageUrl: productImage.asset->url
23         }`
24       );
25       setData(result);
26     };
27
28     fetchData();
29   }, []);
30
31
32   return (
33     <div className="w-screen mt-10 flex justify-center">
34       <div className=
35         "w-[328px] md:w-[1124px] py-[80px] flex flex-col items-center gap-[80px]">
36         /* Text Section */
37         <div className={` w-[300px] md:w-[500px] text-center`>
38           <h4 className="font-normal text-[20px] text-[#737373]">
39             Featured Products
40           </h4>
41           <h3 className="font-bold text-[24px] text-[#252B42]">
42             BESTSELLER PRODUCTS
43           </h3>
44           <p className="font-normal text-[14px] text-[#737373]">
45             Problems trying to resolve the conflict between
46           </p>
47         </div>
48         /* Card Section */
49         <div className=
50           "grid grid-cols-1 sm:grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-[30px]">
51           {data.slice(0,8).map((product: Tproduct, index: number) => (
52             <div key={index}>
53               <Card
54                 title={product.title}
55                 description={product.description}
56                 discountedprice={product.discountedprice}
57                 price={product.price}
58                 imageUrl={product.imageUrl}
59                 id={product.id} quantity={undefined} />
60             </div>
61           ))}
62         </div>
63       </div>
64     </div>
65   );
66 };
67
68 export default Bestseller;
69
70

```

SearchBar: 



```

1  "use client";
2
3  import { useRouter } from "next/navigation";
4  import { client } from "@sanity/lib/client";
5  import React, { useEffect, useState } from "react";
6  import { CiSearch } from "react-icons/ci";
7  import { Tproduct } from "../../utils/componentType";
8
9  const Search = () => {
10   const [searchTerm, setSearchTerm] = useState("");
11   const [showInput, setShowInput] = useState(false);
12   const [products, setProducts] = useState<Tproduct[]>([]);
13   const [filteredProducts, setFilteredProducts] = useState<
14     { id: number;
15       title: string;
16       imageUrl: string
17     }[]>([]);
18   const router = useRouter();
19
20   useEffect(() => {
21     const fetchData = async () => {
22       const fetchedProducts = await client.fetch(
23         `*[_type == "product"]{
24           id,
25           title,
26           "imageUrl": productImage.asset->url
27         }`
28       );
29       setProducts(fetchedProducts);
30     };
31     fetchData();
32   }, []);
33
34   const handleSearch = (e:any) => {
35     const term = e.target.value;
36     setSearchTerm(term);
37
38     if (term.trim() === "") {
39       setFilteredProducts([]);
40       return;
41     }
42
43     const results = products.filter((product) =>
44       product.title.toLowerCase().includes(term.toLowerCase())
45     );
46     setFilteredProducts(
47       results.map((product) => ({
48         id: product.id,
49         title: product.title,
50         imageUrl: product.imageUrl,
51       }))
52     );
53   };
54
55   const handleProductClick = (id: number) => {
56     router.push(`/productList/${id}`);
57   };
58
59   return (
60     <div className="relative">
61       <button className="text-xs flex" onClick={() => setShowInput(!showInput)}>
62         <CiSearch className="text-3xl" />
63       </button>
64
65       {showInput && (
66         <div className=
67           "absolute top-12 left-0 w-full flex justify-center items-center z-50">
68           <input
69             type="text"
70             placeholder="Search products..."
71             value={searchTerm}
72             onChange={handleSearch}
73             className="w-80 h-10 px-4 rounded-md border"/>
74         </div>
75       )}
76
77       {filteredProducts.length > 0 && (
78         <div className=
79           "absolute top-24 left-1/2 transform -translate-x-1/2 justify-center w-[270px] md:w-[400px]
80           bg-white shadow-lg max-h-60 overflow-y-auto rounded-md border z-50"
81         >
82           <ul>
83             {filteredProducts.map((product) => (
84               <li
85                 key={product.id}
86                 className="p-2 hover:bg-gray-200 flex items-center gap-2 cursor-pointer"
87                 onClick={() => handleProductClick(product.id)}>
88                 <img
89                   src={product.imageUrl}
90                   alt={product.title}
91                   className="w-10 h-10 object-cover rounded"/>
92                 <span>{product.title}</span>
93               </li>
94             ))}
95           </ul>
96         </div>
97       )}
98     </div>
99   );
100   export default Search;

```

## Key Features of the E-Commerce Project:

### Core Components:

#### 1. Product Card Component:

- A reusable card for showcasing product details.
- Displays product image, name, and price.
- Includes a toggle button to add/remove item from the wishlist.
- Navigates to a detailed product page on click.
- Uses local storage for managing wishlist functionality.
- 

#### 2. Product List Component:

- Arranges products in a responsive grid format.
- Allows category-based filtering via URL parameters.
- 

#### 3. Category Filter Component:

- Offers a navigation Images component with multiple product categories like shirts, t-shirts, jackets, sweaters, and hoodies.
  - Filters products based on categories using URL query parameters.
  - Adapts seamlessly to different screen sizes with responsive design.
- 

### API Integration and Routing:

#### 1. Sanity CMS Integration:

- Connects to Sanity CMS for managing product data.
- Fetches product details, including images, prices, and categories, with GROQ queries.
- Optimizes images using Sanity's built-in CDN.

#### 2. Dynamic Routing System:

- Generates unique URLs for each product using their Ids.
- Supports SEO-friendly URLs for products and categories.
- Handles category filtering.

#### 3. Data Flow:

- Server components fetch initial product data.
- Client components manage user interactions, like filtering and wishlist updates.
- Local storage keeps track of the wishlist.
- URL parameters maintain the current filter state.

---

## Technical Highlights:

### 1. Technologies Used:

- Built with **Next.js 14**, integrated with **Sanity CMS** for content management.
- Styled using **Tailwind CSS** for modern and responsive design.

### 2. Best Practices:

- **Code Organization:** Uses a component-based structure and Typescript for type safety.
- **Performance Optimization:**
  - Optimized images with Next.js Image component.
  - Lazy loading for faster page performance.
- **Security:**
  - Protects sensitive data with environment variables.
  - Implements input sanitization and API route protection.

---

## Key Features:

1. **Related Products Section:** Suggests similar products.
2. **Dynamic Search:** Real-time filtering with a responsive search bar.
3. **Responsive Design:** Adapts perfectly to different devices and screen sizes.

---

## Challenges and Solutions:

### 1. Dynamic Routing with Sanity:

- **Challenge:** Generating routes dynamically.
- **Solution:** Used Next.js `generateStaticParams` for efficient route handling.

### 2. Real-Time Search:

- **Challenge:** Providing fast, real-time search results.
- **Solution:** Implemented client-side filtering with debouncing to avoid performance issues



### 3. Responsive Design:

- **Challenge:** Ensuring the platform is mobile-friendly.
- **Solution:** Used Tailwind CSS's grid and flex box utilities with breakpoints.

This project combines efficient data management, modern design principles, and a seamless user experience to deliver a fully functional e-commerce platform.