

# Day 4 - Dynamic Frontend Components for [Bandage-App E-Commerce Marketplace]

## Introduction

This report documents the entire process I followed to develop the dynamic frontend components for my e-commerce platform. The purpose of this platform is to create a seamless shopping experience for users, featuring dynamic product listings, category. filtering, search functionality, and other essential e-commerce features. Below, I describe each step I took in detail, highlighting the challenges I faced and the solutions I implemented.

## Process Overview

### 1. Tags and Categories

What I Did

1. Implemented dynamic filtering based on categories and tags.
2. Created a reusable Filter Component for consistent filtering functionality
3. Used dropdowns and checkboxes for user input.

**Created –By = Emad Ahmed**

**Roll-NUM=227065**

**Slot=      Wednesday-7-10**

## Challenges | Faced:

Ensuring efficient API calls.

**Solution:** Applied debouncing and caching mechanisms to optimize performance.



**Bold Nest**

\$260.00

Add to Cart

View Details

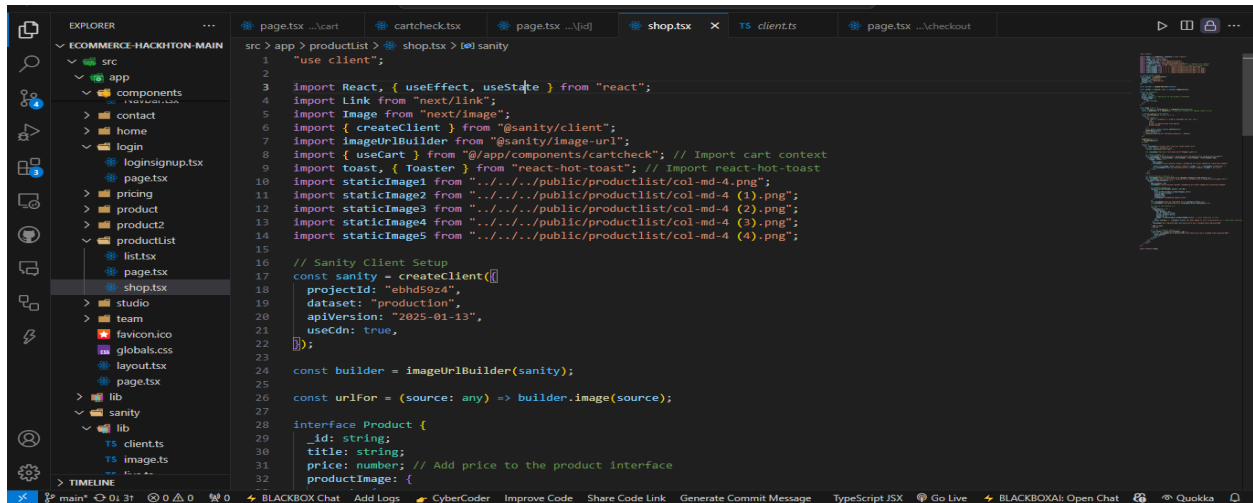


**Rustic Vase Set**

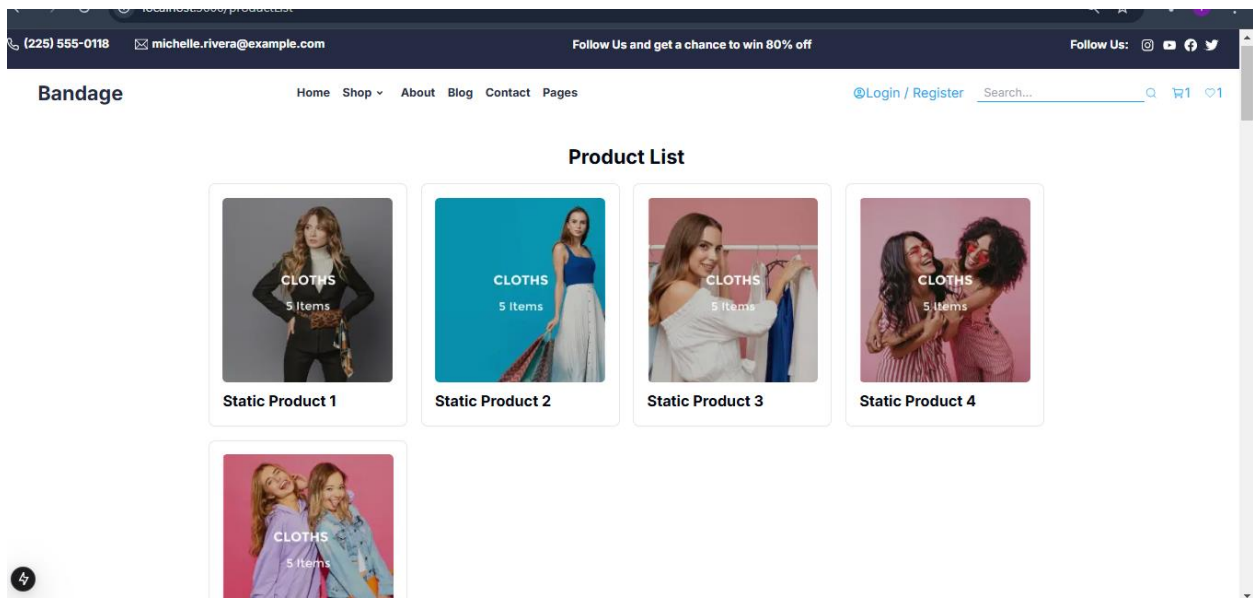
\$210.00

Add to Cart

View Details



```
src > app > productList > shop.tsx > (e) sanity
1  "use client";
2
3  import React, { useEffect, useState } from "react";
4  import Link from "next/link";
5  import Image from "next/image";
6  import { createClient } from "@sanity/client";
7  import imageUrlBuilder from "@sanity/image-url";
8  import { useCart } from "@app/components/cartcheck"; // Import cart context
9  import toast, { Toaster } from "react-hot-toast"; // Import react-hot-toast
10 import staticImage1 from "../../public/productlist/col-md-4 (1).png";
11 import staticImage2 from "../../public/productlist/col-md-4 (2).png";
12 import staticImage3 from "../../public/productlist/col-md-4 (3).png";
13 import staticImage4 from "../../public/productlist/col-md-4 (4).png";
14 import staticImage5 from "../../public/productlist/col-md-4 (5).png";
15
16 // Sanity Client Setup
17 const sanity = createClient({
18   projectId: "ebhd5924",
19   dataset: "production",
20   apiVersion: "2025-01-13",
21   useCdn: true,
22 });
23
24 const builder = imageUrlBuilder(sanity);
25
26 const urlFor = (source: any) => builder.image(source);
27
28 interface Product {
29   _id: string;
30   title: string;
31   price: number; // Add price to the product interface
32   productImage: {
```



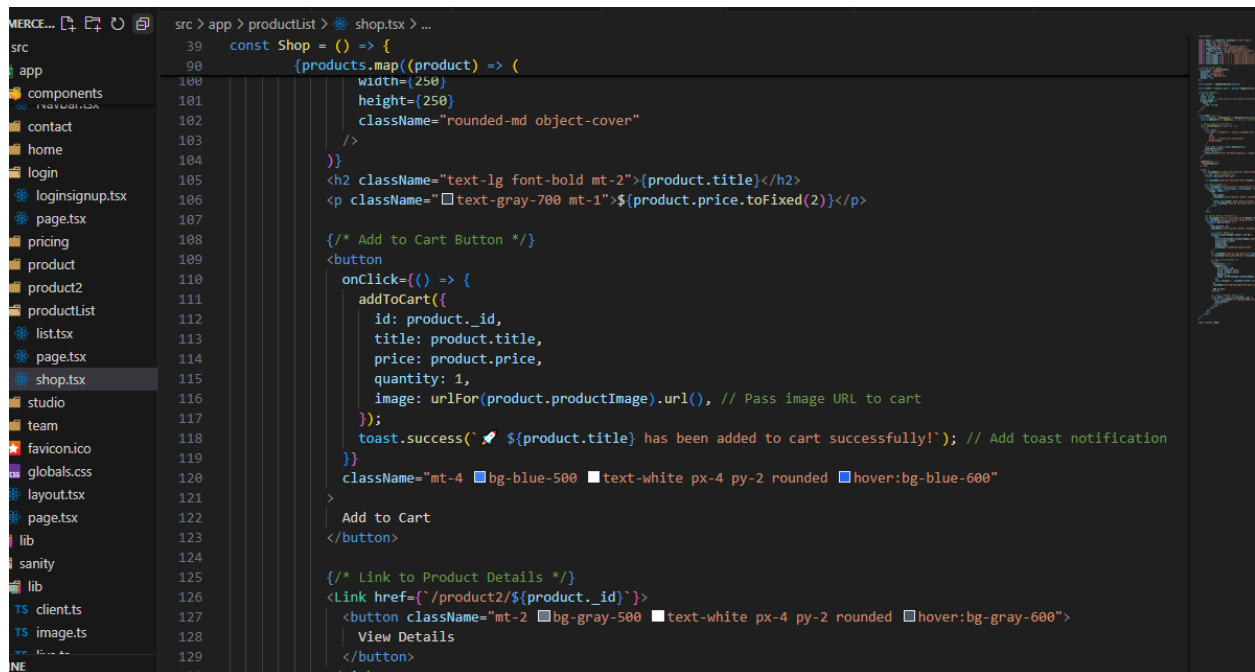
## 2. Pagination

What I Did:

Developed a `PaginationComponent` to divide products into smaller, navigable pages.

Used Next.js's `getStaticProps` and `getServerSideProps` for efficient data fetching.

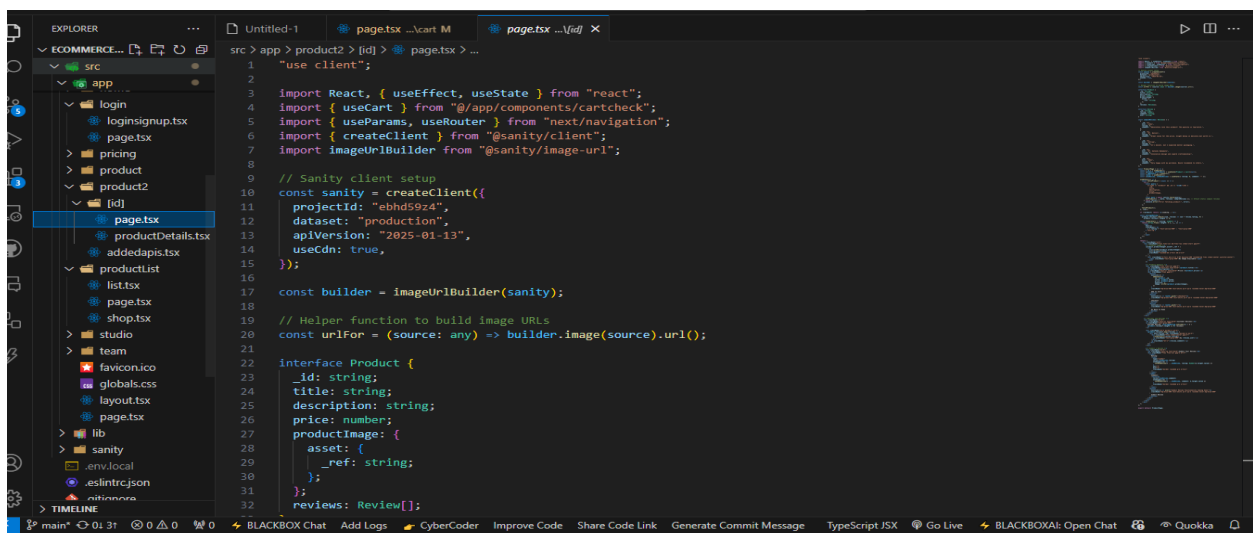
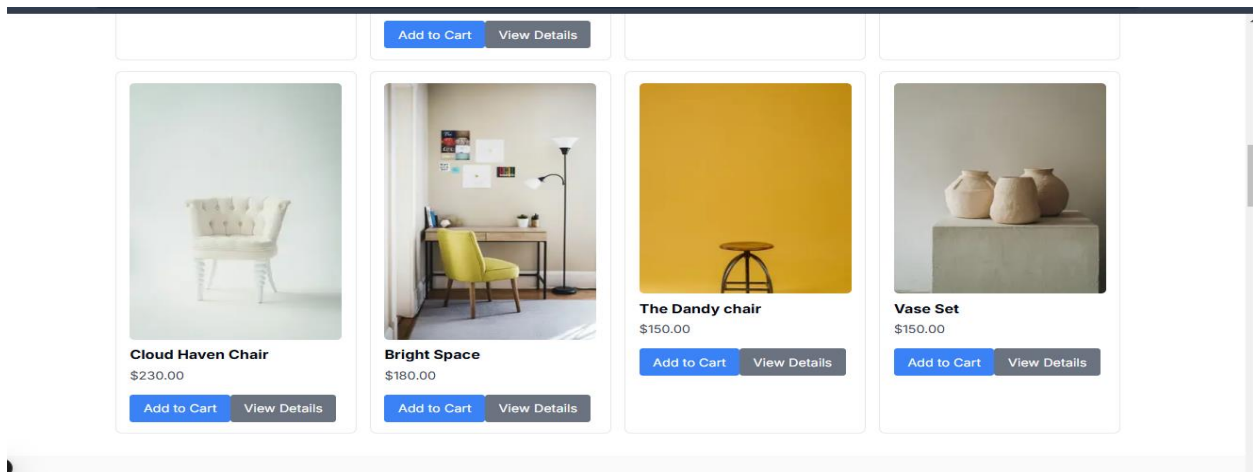
Styled navigation buttons for easy use.



```
src > app > productList > shop.tsx > ...
39  const Shop = () => {
90    {products.map((product) => (
100      width={250}
101      height={250}
102      className="rounded-md object-cover"
103    )}
104  )}
105  <h2 className="text-lg font-bold mt-2">{product.title}</h2>
106  <p className="text-gray-700 mt-1">${product.price.toFixed(2)}</p>
107
108  {/* Add to Cart Button */}
109  <button
110    onClick={() => {
111      addToCart({
112        id: product.id,
113        title: product.title,
114        price: product.price,
115        quantity: 1,
116        image: urlFor(product.productImage).url(), // Pass image URL to cart
117      });
118      toast.success(`${product.title} has been added to cart successfully!`); // Add toast notification
119    }}
120    className="mt-4 bg-blue-500 text-white px-4 py-2 rounded hover:bg-blue-600"
121  >
122    Add to Cart
123  </button>
124
125  {/* Link to Product Details */}
126  <Link href={`/product2/${product.id}`}>
127    <button className="mt-2 bg-gray-500 text-white px-4 py-2 rounded hover:bg-gray-600">
128      View Details
129    </button>
130  </Link>
```

### 3. Product Card

- What I Did:
  1. Designed a Product Card component to display product details like name, price, stock status, and image.
  2. Styled the cards using Tailwind CSS for responsiveness.
  3. Added interactive buttons for "Add to Cart" and "Wishlist."
- Challenges | Faced:
  - Handling inconsistent product image sizes.
  - Solution: Used CSS object-fit properties to maintain uniformity.



## 4. Product Listing

- What I Did:
- Created a ProductList component to display a grid of products fetched dynamically from the API.
- Implemented lazy loading for improved performance

### 1. Challenges I Faced:

- Managing large datasets.

**Solution:** Implemented infinite scrolling for smoother user experience.

## 5. Add to Cart

### What I Did:

- Developed the cart functionality using React Context API to manage the global state
- Enabled dynamic quantity updates for cart items.

### Challenges | Faced

- Syncing cart state across multiple components.
- **Solution:** Used local storage to persist cart data.

•

#### Shopping Cart



##### Timber Craft

Price: \$320.00

Quantity: 1

[Remove](#)



##### Amber Haven

Price: \$150.00

Quantity: 1

[Remove](#)

#### Order Summary

Items (2): \$470.00

Shipping: Free

**Total: \$470.00**

[Proceed to Checkout](#)

[Clear Cart](#)

## Login

Email

Password

Login

Don't have an account? [Signup](#)

OR

Login with Google

Login with Apple

[Go Back to Home](#)

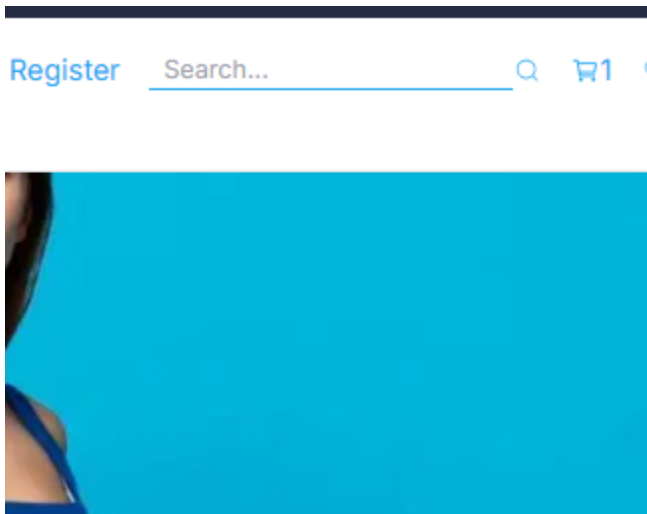
```

src > app > login > loginsignup.tsx > LoginSignup
1  "use client";
2
3  import React, { useState } from "react";
4  import { useRouter } from "next/navigation";
5
6  const LoginSignup = () => {
7    const router = useRouter();
8    const [isLogin, setIsLogin] = useState(true);
9    const [email, setEmail] = useState("");
10   const [password, setPassword] = useState("");
11
12   const handleToggle = () => setIsLogin(!isLogin);
13
14   const handleSubmit = (e: React.FormEvent) => {
15     e.preventDefault();
16     if (isLogin) {
17       console.log("Logging in with", email, password);
18     } else {
19       console.log("Signing up with", email, password);
20     }
21     router.push("/");
22   };
23
24   const handleGoogleLogin = () => {
25     console.log("Google Login Clicked");
26     // Add Google OAuth functionality here
27   };
  
```

## 6. Login and Sign-Up Page

- **What I Did:**

- **Built user authentication pages for login and sign-up with form validation, Integrated Firebase for secure authentication and user management.**
- **Challenges | Faced**
- **Handling errors during login and sign-up.**
- **Solution: Provided user-friendly error messages for better feedback.**



## 7. Search Bar Component

- **What I Did:**
- **Developed a Searchilar component for searching products by name or tags.**
- **Integrated API calls to fetch search results in real time.**
- **Challenges | Faced:**
- **Ensuring case-insensitive searches.**
- **Solution: Normalized search input and data using JavaScript string methods.**



