Hackathon Day 4

Details Documentation for Dynamic Components and Functionalities

This documentation provides an in-depth analysis of the key functionalities for a dynamic marketplace,emphasizing modularity ,reusability, and integration with sanity CMS.Each feature is described comprehensively,followed by a conclusion summarizing the approach.

Step-1:Functionalities Overview

The Project implements the following core functionalities:

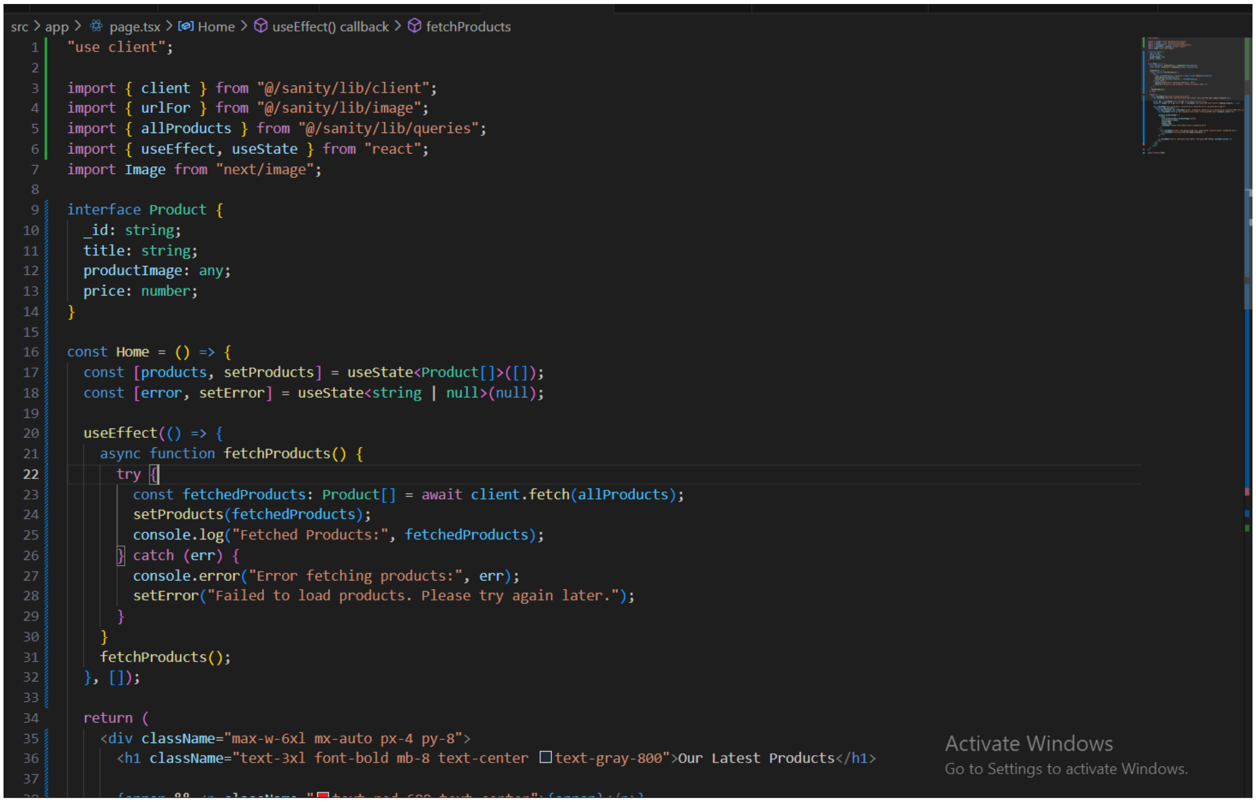
1 .Product Listing Page
2. Dynamic Route
3. Cart Functionality
4. Checkout
5. Price Calculation
6. Product Comparison
7. Inventory Management

Each functionality contributes to building a responsive and scalable market place.

Step-2: Functionality in Detail:

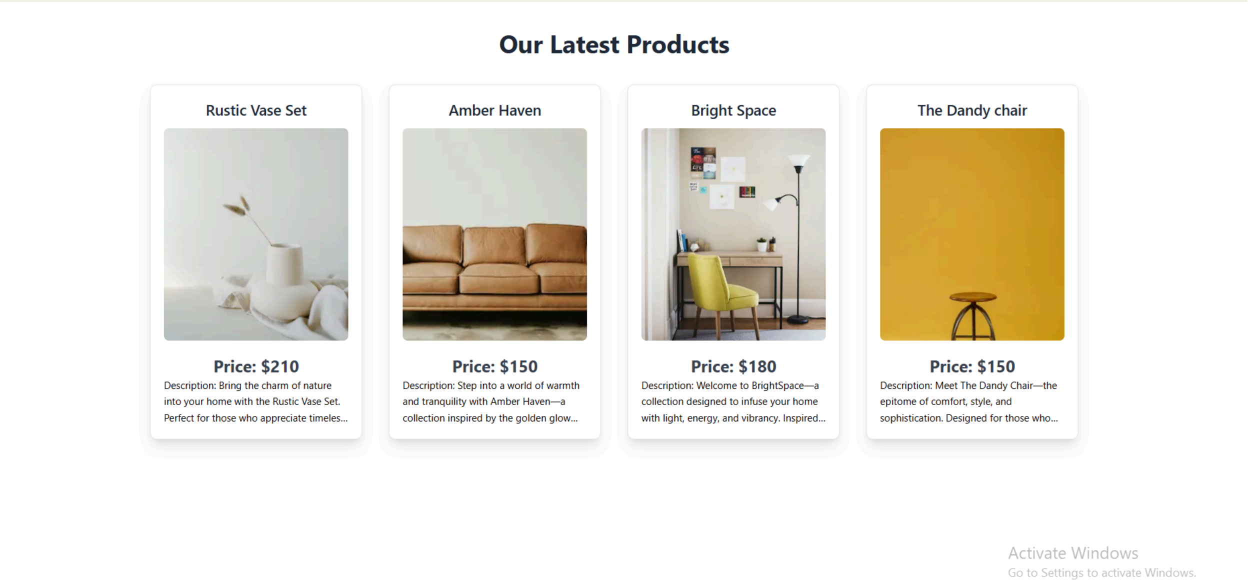
1. **Product Listing Page**

The Product Listing page is the primary interface where users can view all the available products in a structured and visually appealing format.products are displayed dynamically,fetched from sanityCMS,and rendered in a grid or list layout.



src > app > page.tsx > Home > useEffect() callback > fetchProducts

```tsx
"use client";

import { client } from "@/sanity/lib/client";
import { urlFor } from "@/sanity/lib/image";
import { allProducts } from "@/sanity/lib/queries";
import { useEffect, useState } from "react";
import Image from "next/image";

interface Product {
  _id: string;
  title: string;
  productImage: any;
  price: number;
}

const Home = () => {
  const [products, setProducts] = useState<Product[]>([]);
  const [error, setError] = useState<string | null>(null);

  useEffect(() => {
    async function fetchProducts() {
      try {
        const fetchedProducts: Product[] = await client.fetch(allProducts);
        setProducts(fetchedProducts);
        console.log("Fetched Products:", fetchedProducts);
      } catch (err) {
        console.error("Error fetching products:", err);
        setError("Failed to load products. Please try again later.");
      }
    }
    fetchProducts();
  }, []);

  return (
    <div className="max-w-6xl mx-auto px-4 py-8">
      <h1 className="text-3xl font-bold mb-8 text-center □text-gray-800">Our Latest Products</h1>

      {error && <p className="🟥text-red-600 text-center">{error}</p>}
```



Our Latest Products

Rustic Vase Set

Price: $210

Description: Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who appreciate timeles...

Amber Haven

Price: $150

Description: Step into a world of warmth and tranquility with Amber Haven—a collection inspired by the golden glow...

Bright Space

Price: $180

Description: Welcome to BrightSpace—a collection designed to infuse your home with light, energy, and vibrancy. Inspired...

The Dandy chair

Price: $150

Description: Meet The Dandy Chair—the epitome of comfort, style, and sophistication. Designed for those who...

Detailed Description:

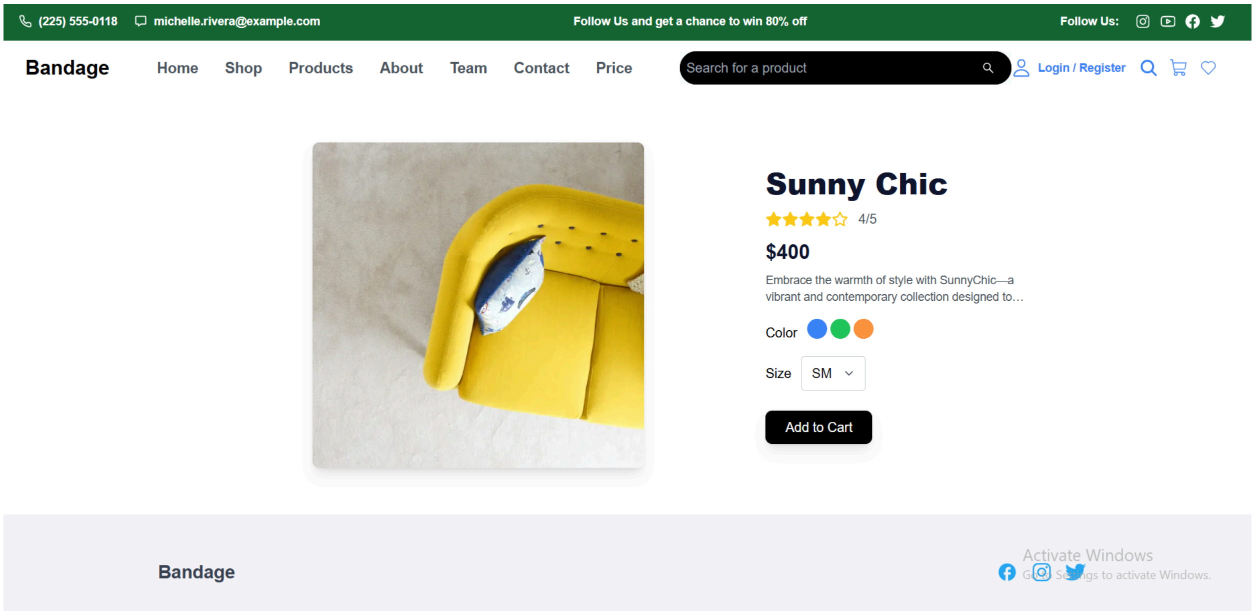
1- The page offers sorting and filtering options to enhance usability,allowing users to organize products based on price,categories,or popularity.

2- Pagination ensures the seamless handling of large datasets,improving performance and user experience.

3- Responsive design ensures compatibility across device,from desktops to mobile phones.
4- Integration with Sanity CMS ensures real time updates,so any product changes in the backend are instantly reflected.

2-Dynamic Route

Dynamic routing allows for the creation of individual product detial pages,enabling users to view detailed information about each product.



(225) 555-0118 michelle.rivera@example.com

Follow Us and get a chance to win 80% off

Follow Us:

Bandage

Sunny Chic

★★★★☆ 4/5

$400

Embrace the warmth of style with SunnyChic—a vibrant and contemporary collection designed to...

Color ● ● ●

Size SM

Bandage

Detailed Description

Every product has a unique identifier (ID or Slug) used to dynamically generate its URL(e.g /product/ [id]).
These pages are server-rendered to improve SEO and provide faster initial load times.
Dynamic routes display details like product descriptions,images,price,stock status and reviews.
This approach ensures scalability, allowing new products to automatically generate corresponding pages without manual intervention.

3-Cart Functionality

The Cart Functionality manages the user ‘s selected items,providing a seamless shopping experience by tracking their choices and summarizing costs.

4-Checkout

The checkout Functionality streamlines the purchase process,collecting and validating user information to finalize the order.

5-Price Calculation

Price calculation dynamically computes the total cost of items in the cart,factoring in taxes,discount, and other adjustments.

6-Product Comparison

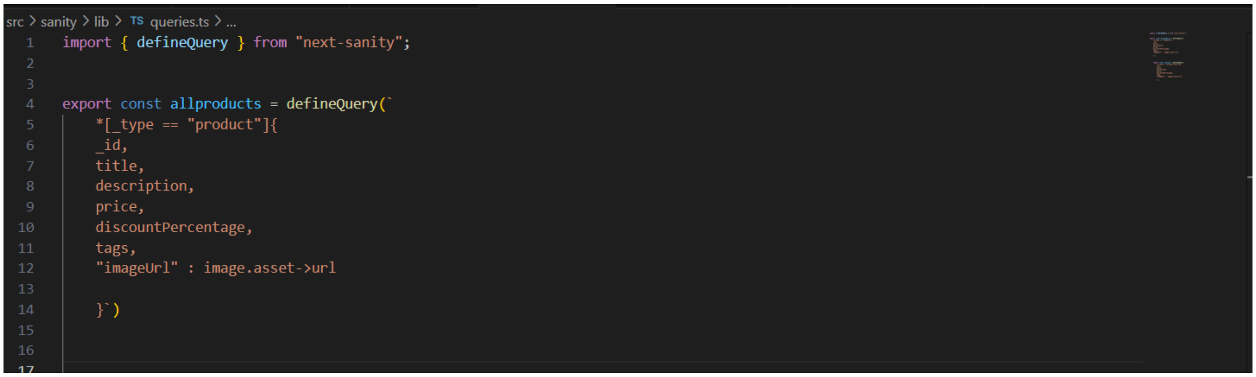
Product Comparison enables users to evaluate multiple products side by side, facilitating informed purchasing decisions.

7-Inventory Management

Inventory Management tracks the availability of product, ensuring users are informed about stock level.

Step-3: Integration with Sanity CMS

Sanity CMS serves as the backend for managing and retrieving product data dynamically.



```
import { defineQuery } from "next-sanity";


export const allproducts = defineQuery(`
    *[_type == "product"]{
    _id,
    title,
    description,
    price,
    discountPercentage,
    tags,
    "imageUrl" : image.asset->url

    }`)
```

Conclusion

 This documentation outlines a comprehensive approach to building dynamic and responsive marketplace components.By leveraging sanity CMS for backend management and modular frontend development techniques,the application achieves scalability ,efficiency and a superior experience.

Each functionality from product listing inventory management plays a vital role in delivering a professional marketplace that meets real world needs. Future enhancements,such as integrating advanced analytic or AI based recommendation,can further elevate the platform.

For any additional details,enhancements or implementation support,feel free to reach-out.