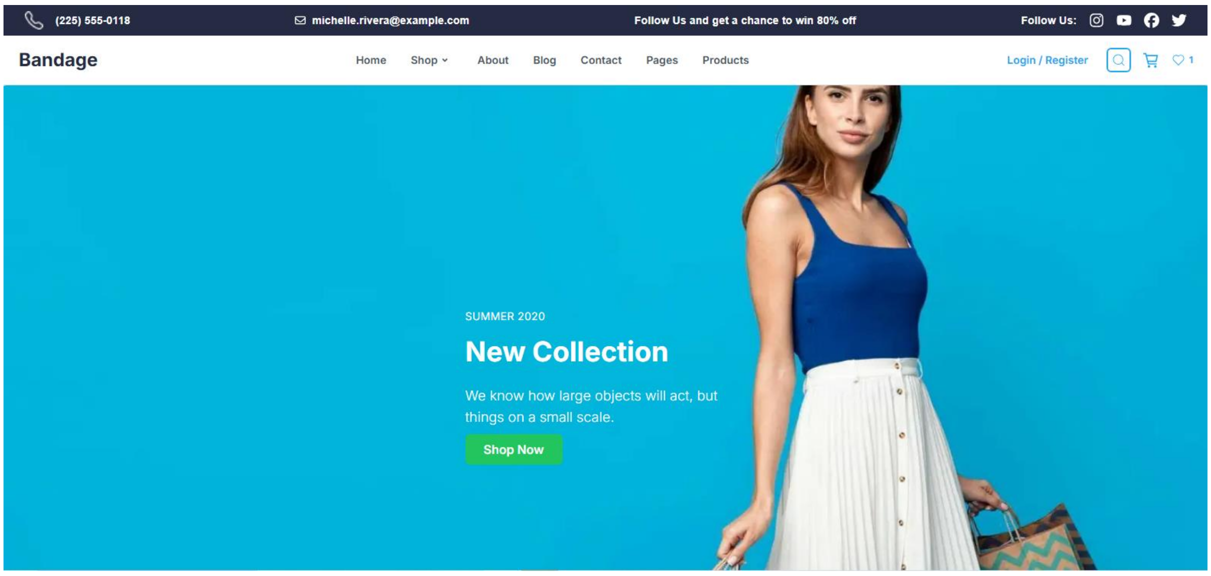
Name: Aafifa Afaq

Rollno:0007335

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

Introduction

This report documents the entire process I followed to develop the dynamic frontend components for my e-commerce platform. The goal 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 outline each step of the development process, challenges I faced, solutions I implemented, and additional ideas for improvement.

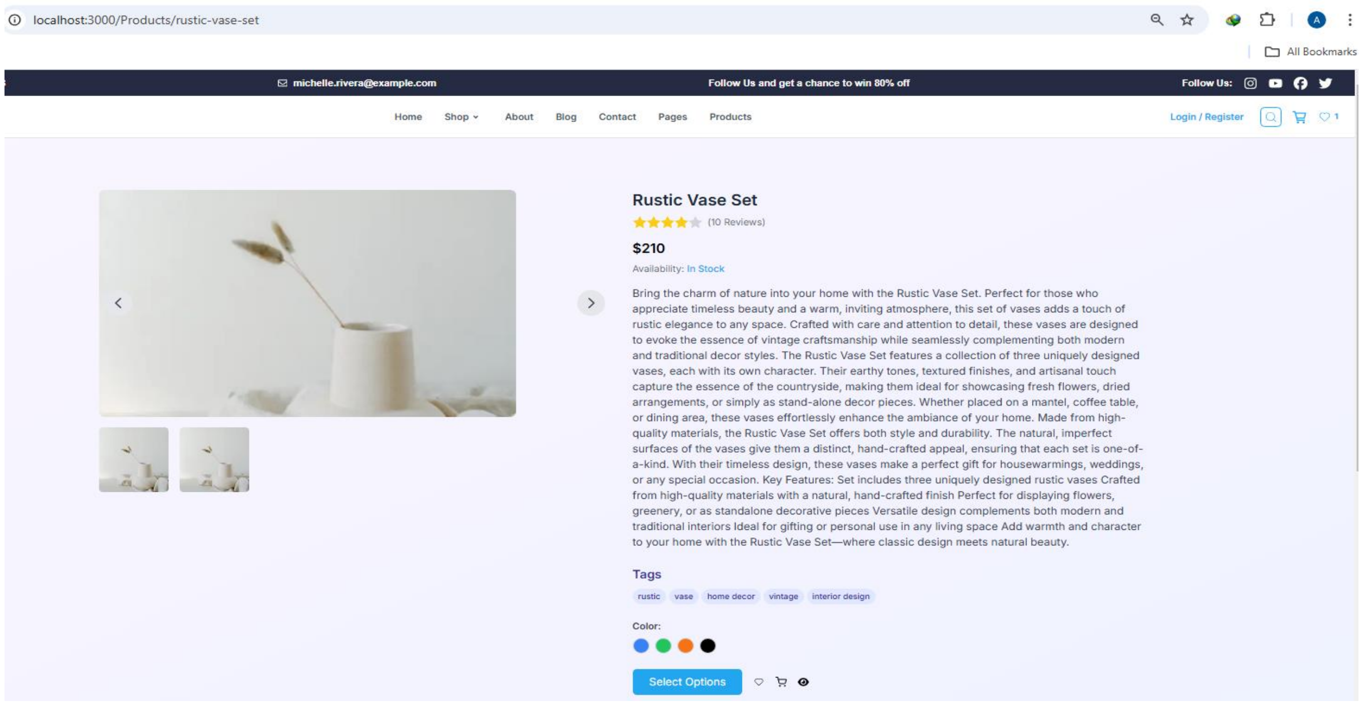

(225) 555-0118
michelle.rivera@example.com
Follow Us and get a chance to win 80% off
Follow Us:
Bandage
Home
Shop
About
Blog
Contact
Pages
Products
Login / Register
1
SUMMER 2020
New Collection
We know how large objects will act, but things on a small scale.
Shop Now

Process Overview

1. Product Card

- **What I Did:**
  - Designed a ProductCard component to display details like name, price, and image.
  - Styled the cards using Tailwind CSS for responsiveness.
  - Added interactive button for "Add to Cart".
- **Challenges I Faced:**
  - Handling inconsistent product image sizes.
- **Solution:**
  - Adjust it by Tailwind CSS properties to maintain uniformity.
- **Additional Features Can Be Added:**

- Displayed a quick “View Details” button for a modal preview.
- Highlighted items on sale with a discount badge.


localhost:3000/Products/rustic-vase-set
All Bookmarks
michelle.rivera@example.com
Follow Us and get a chance to win 80% off
Follow Us:
Home
Shop
About
Blog
Contact
Pages
Products
Login / Register
1
Select Options

Rustic Vase Set

★★★★☆ (10 Reviews)

$210

Availability: In Stock

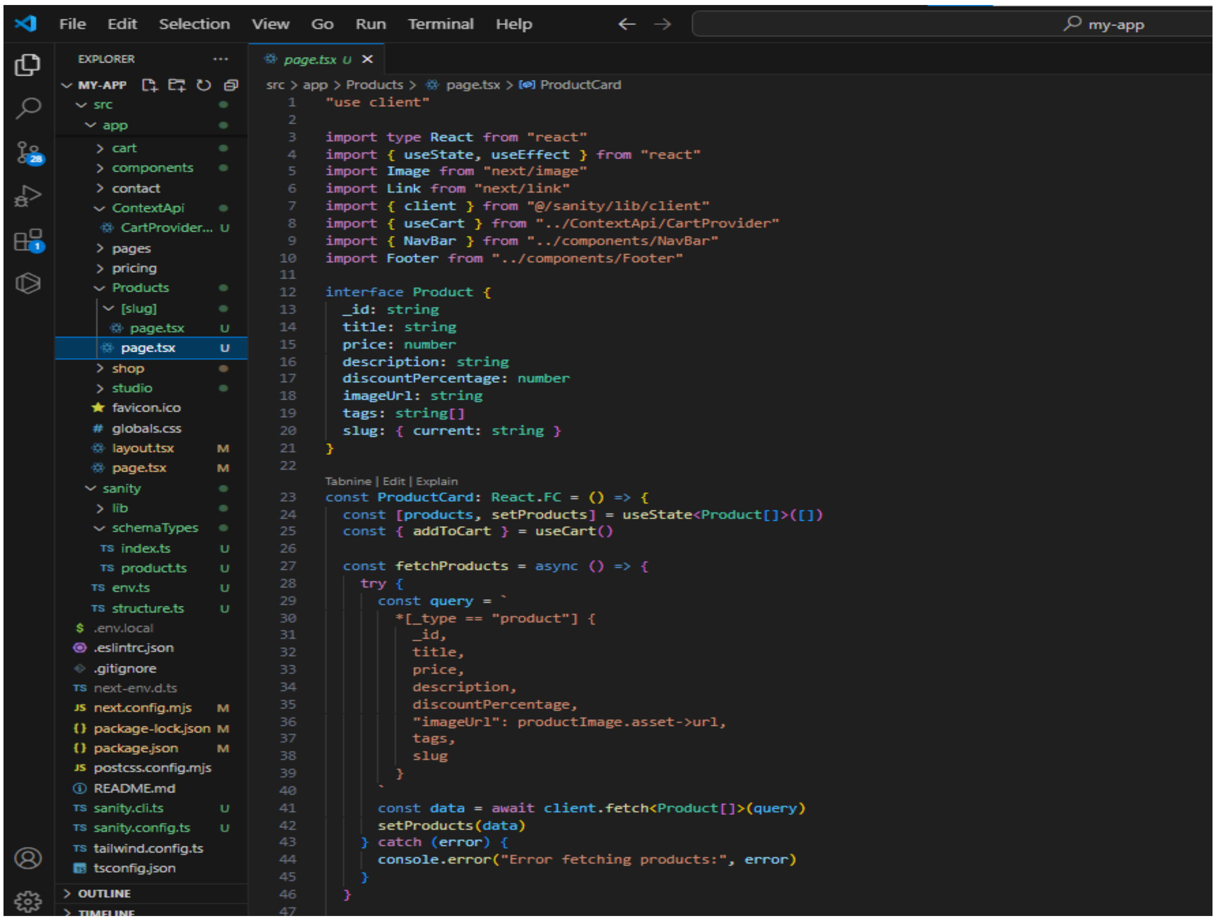
Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who appreciate timeless beauty and a warm, inviting atmosphere, this set of vases adds a touch of rustic elegance to any space. Crafted with care and attention to detail, these vases are designed to evoke the essence of vintage craftsmanship while seamlessly complementing both modern and traditional decor styles. The Rustic Vase Set features a collection of three uniquely designed vases, each with its own character. Their earthy tones, textured finishes, and artisanal touch capture the essence of the countryside, making them ideal for showcasing fresh flowers, dried arrangements, or simply as stand-alone decor pieces. Whether placed on a mantel, coffee table, or dining area, these vases effortlessly enhance the ambiance of your home. Made from high-quality materials, the Rustic Vase Set offers both style and durability. The natural, imperfect surfaces of the vases give them a distinct, hand-crafted appeal, ensuring that each set is one-of-a-kind. With their timeless design, these vases make a perfect gift for housewarmings, weddings, or any special occasion. Key Features: Set includes three uniquely designed rustic vases Crafted from high-quality materials with a natural, hand-crafted finish Perfect for displaying flowers, greenery, or as standalone decorative pieces Versatile design complements both modern and traditional interiors Ideal for gifting or personal use in any living space Add warmth and character to your home with the Rustic Vase Set—where classic design meets natural beauty.

Tags

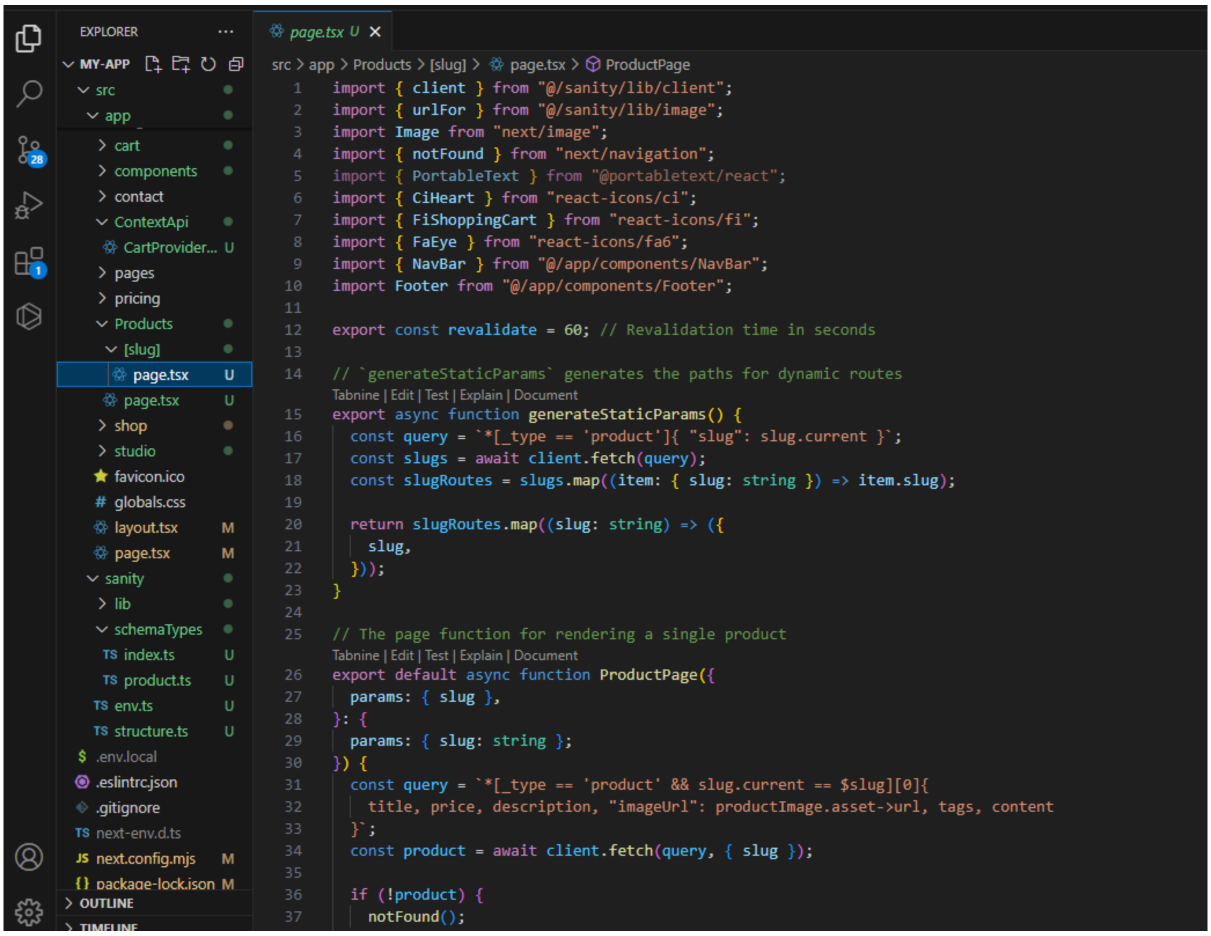
rustic vase home decor vintage interior design

Color:

🔵 🟢 🟠 ⚫



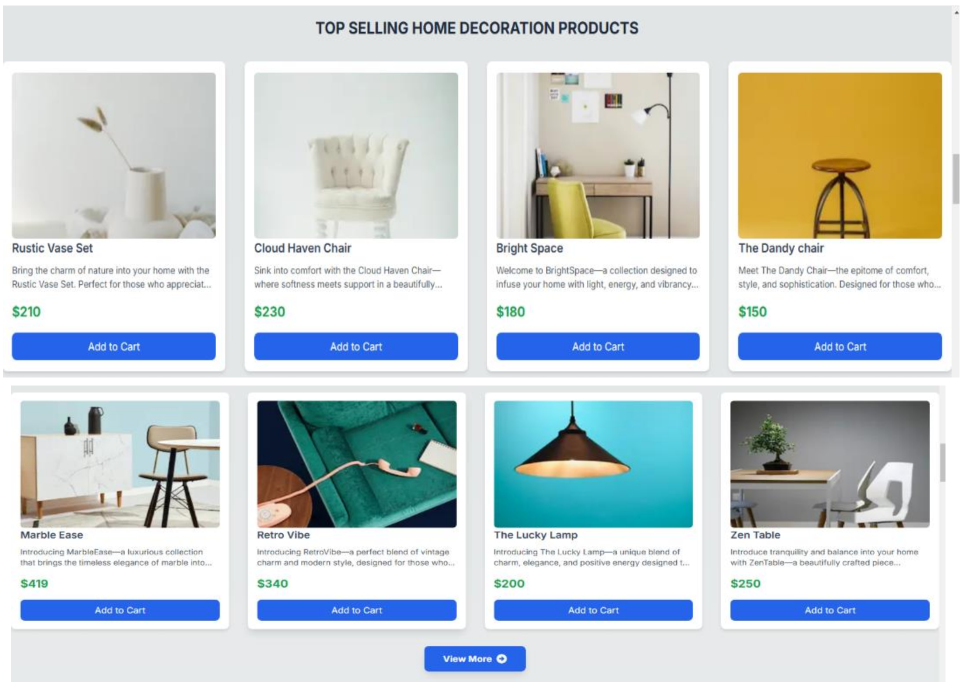
```tsx
"use client"

import type React from "react"
import { useState, useEffect } from "react"
import Image from "next/image"
import Link from "next/link"
import { client } from "@/sanity/lib/client"
import { useCart } from "../ContextApi/CartProvider"
import { NavBar } from "../components/NavBar"
import Footer from "../components/Footer"

interface Product {
  _id: string
  title: string
  price: number
  description: string
  discountPercentage: number
  imageUrl: string
  tags: string[]
  slug: { current: string }
}

const ProductCard: React.FC = () => {
  const [products, setProducts] = useState<Product[]>([])
  const { addToCart } = useCart()

  const fetchProducts = async () => {
    try {
      const query = `
        *[_type == "product"] {
          _id,
          title,
          price,
          description,
          discountPercentage,
          "imageUrl": productImage.asset->url,
          tags,
          slug
        }
      `
      const data = await client.fetch<Product[]>(query)
      setProducts(data)
    } catch (error) {
      console.error("Error fetching products:", error)
    }
  }

```



```tsx
import { client } from "@/sanity/lib/client";
import { urlFor } from "@/sanity/lib/image";
import Image from "next/image";
import { notFound } from "next/navigation";
import { PortableText } from "@portabletext/react";
import { CiHeart } from "react-icons/ci";
import { FiShoppingCart } from "react-icons/fi";
import { FaEye } from "react-icons/fa6";
import { NavBar } from "@/app/components/NavBar";
import Footer from "@/app/components/Footer";

export const revalidate = 60; // Revalidation time in seconds

// `generateStaticParams` generates the paths for dynamic routes
export async function generateStaticParams() {
  const query = `*[_type == 'product']{ "slug": slug.current }`;
  const slugs = await client.fetch(query);
  const slugRoutes = slugs.map((item: { slug: string }) => item.slug);

  return slugRoutes.map((slug: string) => ({
    slug,
  }));
}

// The page function for rendering a single product
export default async function ProductPage({
  params: { slug },
}: {
  params: { slug: string };
}) {
  const query = `*[_type == 'product' && slug.current == $slug][0]{
    title, price, description, "imageUrl": productImage.asset->url, tags, content
  }`;
  const product = await client.fetch(query, { slug });

  if (!product) {
    notFound();
```

2. Top Product Listing

- **What I Did:**
  - Created a TopProductList component to display a grid of products dynamically fetched from the API which fetch 8 products.
- **Challenges I Faced:**
  - Managing large datasets.
- **Solution:**
  - Added a view more button and link it to page having more products
- **Additional Features Can Be Added:**
  - Implement control over infinite scrolling.



TOP SELLING HOME DECORATION PRODUCTS

Rustic Vase Set
Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who appreciat...
$210
Add to Cart

Cloud Haven Chair
Sink into comfort with the Cloud Haven Chair—where softness meets support in a beautifully...
$230
Add to Cart

Bright Space
Welcome to BrightSpace—a collection designed to infuse your home with light, energy, and vibrancy...
$180
Add to Cart

The Dandy chair
Meet The Dandy Chair—the epitome of comfort, style, and sophistication. Designed for those who...
$150
Add to Cart

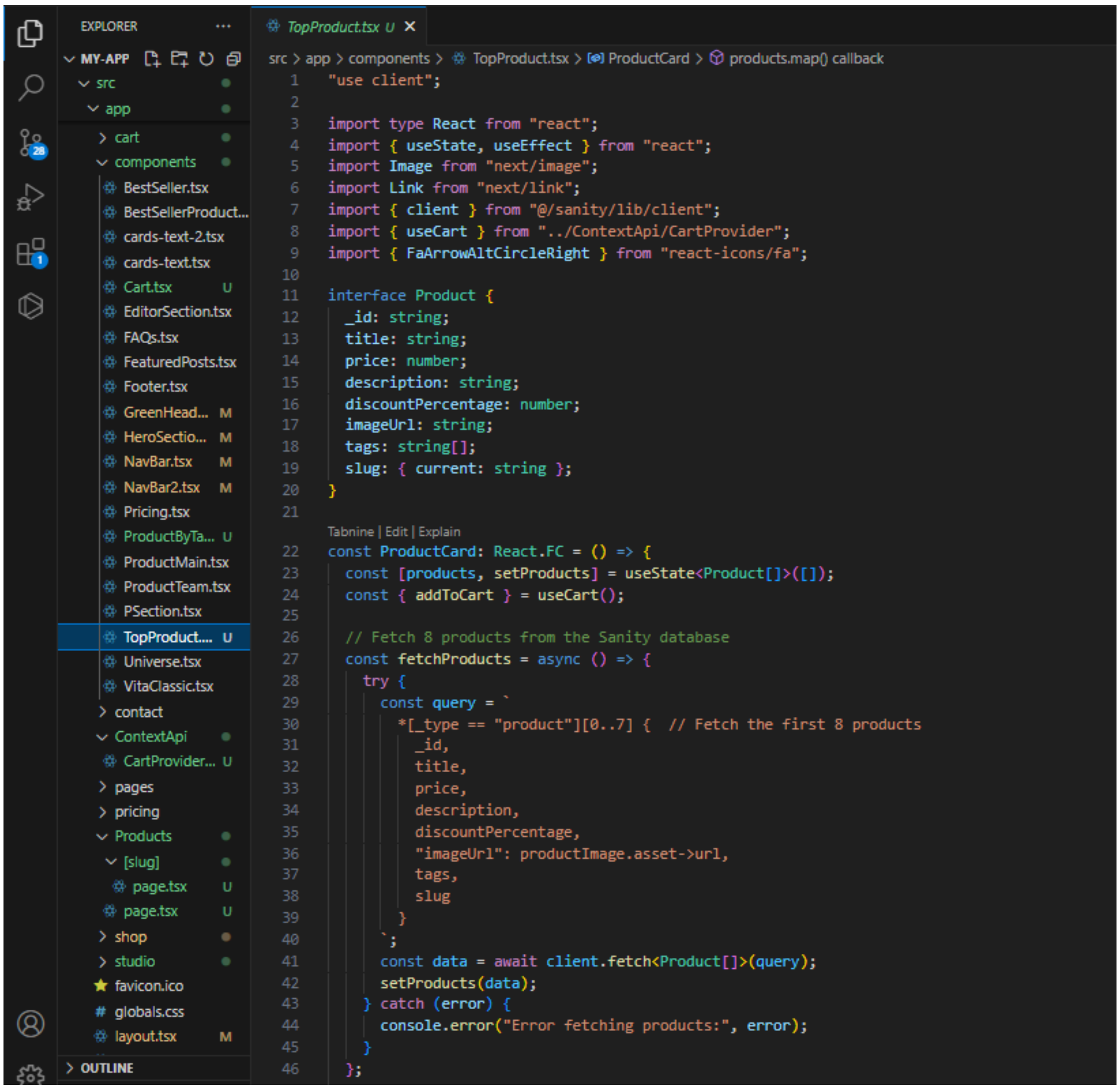
Marble Ease
Introducing MarbleEase—a luxurious collection that brings the timeless elegance of marble into...
$419
Add to Cart

Retro Vibe
Introducing RetroVibe—a perfect blend of vintage charm and modern style, designed for those who...
$340
Add to Cart

The Lucky Lamp
Introducing The Lucky Lamp—a unique blend of charm, elegance, and positive energy designed t...
$200
Add to Cart

Zen Table
Introduce tranquility and balance into your home with ZenTable—a beautifully crafted piece...
$250
Add to Cart

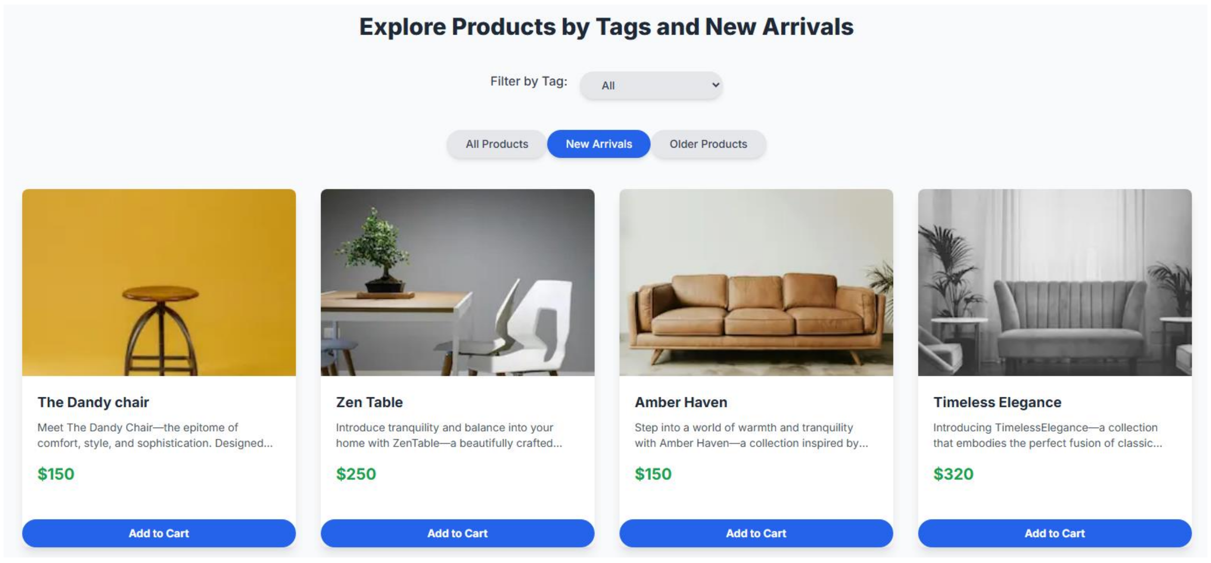
View More



```
"use client";

import type React from "react";
import { useState, useEffect } from "react";
import Image from "next/image";
import Link from "next/link";
import { client } from "@/sanity/lib/client";
import { useCart } from "../ContextApi/CartProvider";
import { FaArrowAltCircleRight } from "react-icons/fa";

interface Product {
  _id: string;
  title: string;
  price: number;
  description: string;
  discountPercentage: number;
  imageUrl: string;
  tags: string[];
  slug: { current: string };
}

const ProductCard: React.FC = () => {
  const [products, setProducts] = useState<Product[]>([]);
  const { addToCart } = useCart();

  // Fetch 8 products from the Sanity database
  const fetchProducts = async () => {
    try {
      const query = `
        *[_type == "product"][0..7] {  // Fetch the first 8 products
          _id,
          title,
          price,
          description,
          discountPercentage,
          "imageUrl": productImage.asset->url,
          tags,
          slug
        }
      `;
      const data = await client.fetch<Product[]>(query);
      setProducts(data);
    } catch (error) {
      console.error("Error fetching products:", error);
    }
  };
```

3. Tags and New Arrival

- **What I Did:**
  - Implemented dynamic filtering based on New Arrival and tags.
  - Used dropdowns for user input.
- **Challenges I Faced:**
  - Efficiently handling API calls.
- **Solution:**
  - Applied debouncing and caching mechanisms to optimize performance.
- **Additional Features Can Be Added:**
  - Implemented a "Clear All Filters" button to reset filters for better usability.



Explore Products by Tags and New Arrivals

Filter by Tag: All

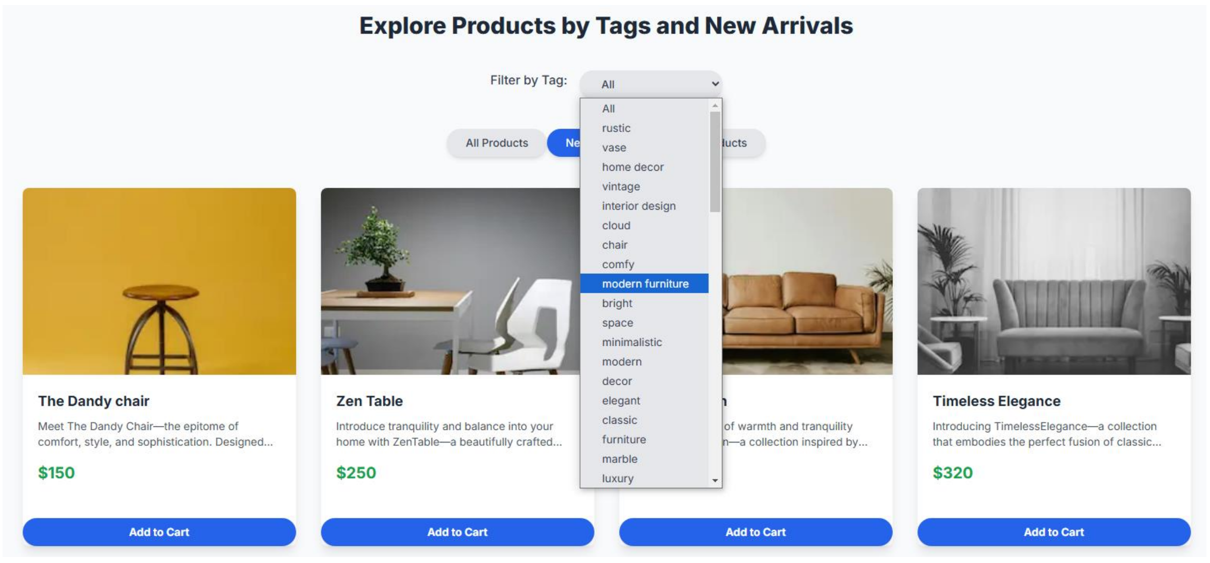
All Products | New Arrivals | Older Products

The Dandy chair
Meet The Dandy Chair—the epitome of comfort, style, and sophistication. Designed...
$150
Add to Cart

Zen Table
Introduce tranquility and balance into your home with ZenTable—a beautifully crafted...
$250
Add to Cart

Amber Haven
Step into a world of warmth and tranquility with Amber Haven—a collection inspired by...
$150
Add to Cart

Timeless Elegance
Introducing TimelessElegance—a collection that embodies the perfect fusion of classic...
$320
Add to Cart



Explore Products by Tags and New Arrivals

Filter by Tag: All

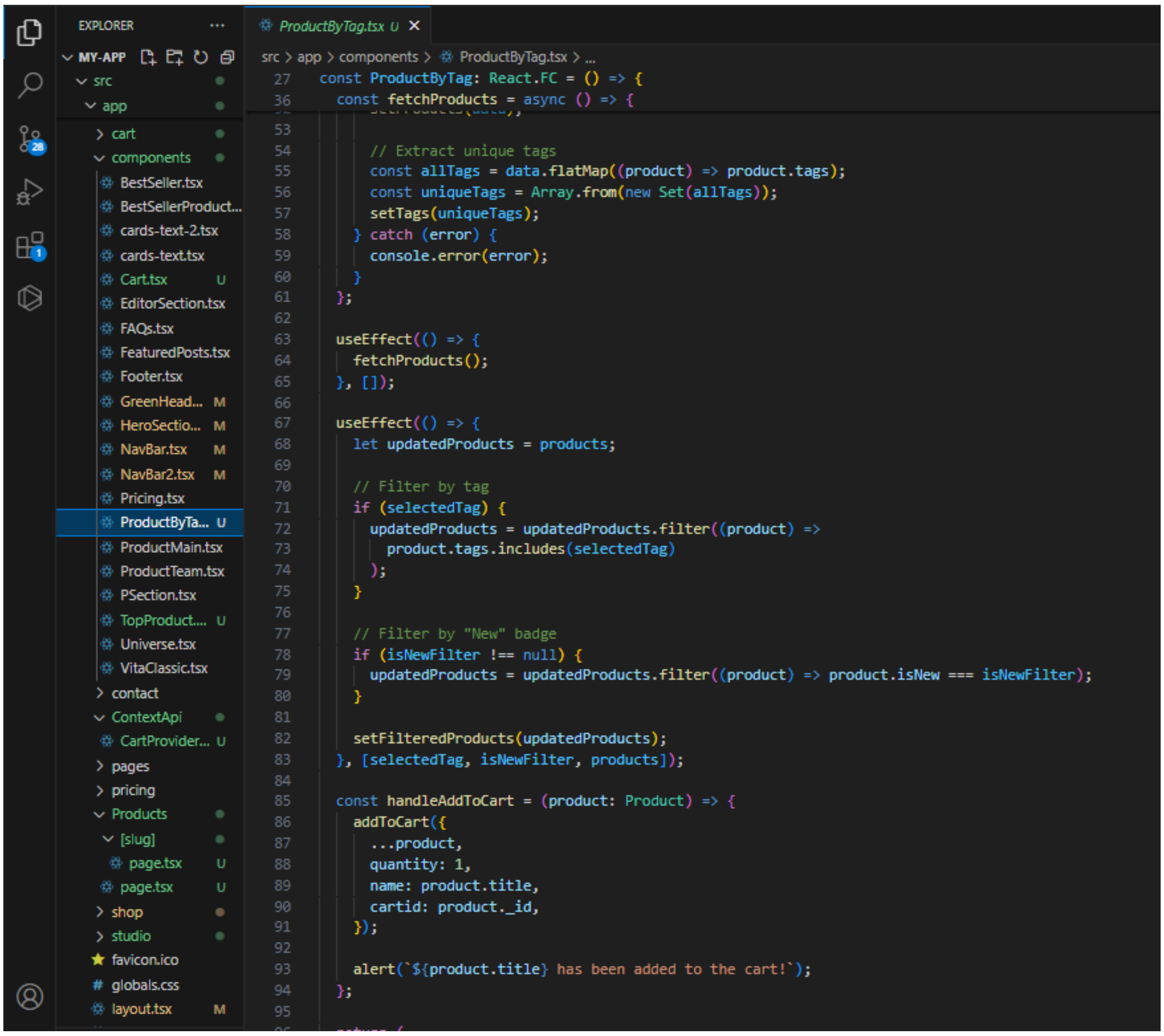
All Products | Ne | lucts

The Dandy chair
Meet The Dandy Chair—the epitome of comfort, style, and sophistication. Designed...
$150
Add to Cart

Zen Table
Introduce tranquility and balance into your home with ZenTable—a beautifully crafted...
$250
Add to Cart

n
of warmth and tranquility
n—a collection inspired by...
Add to Cart

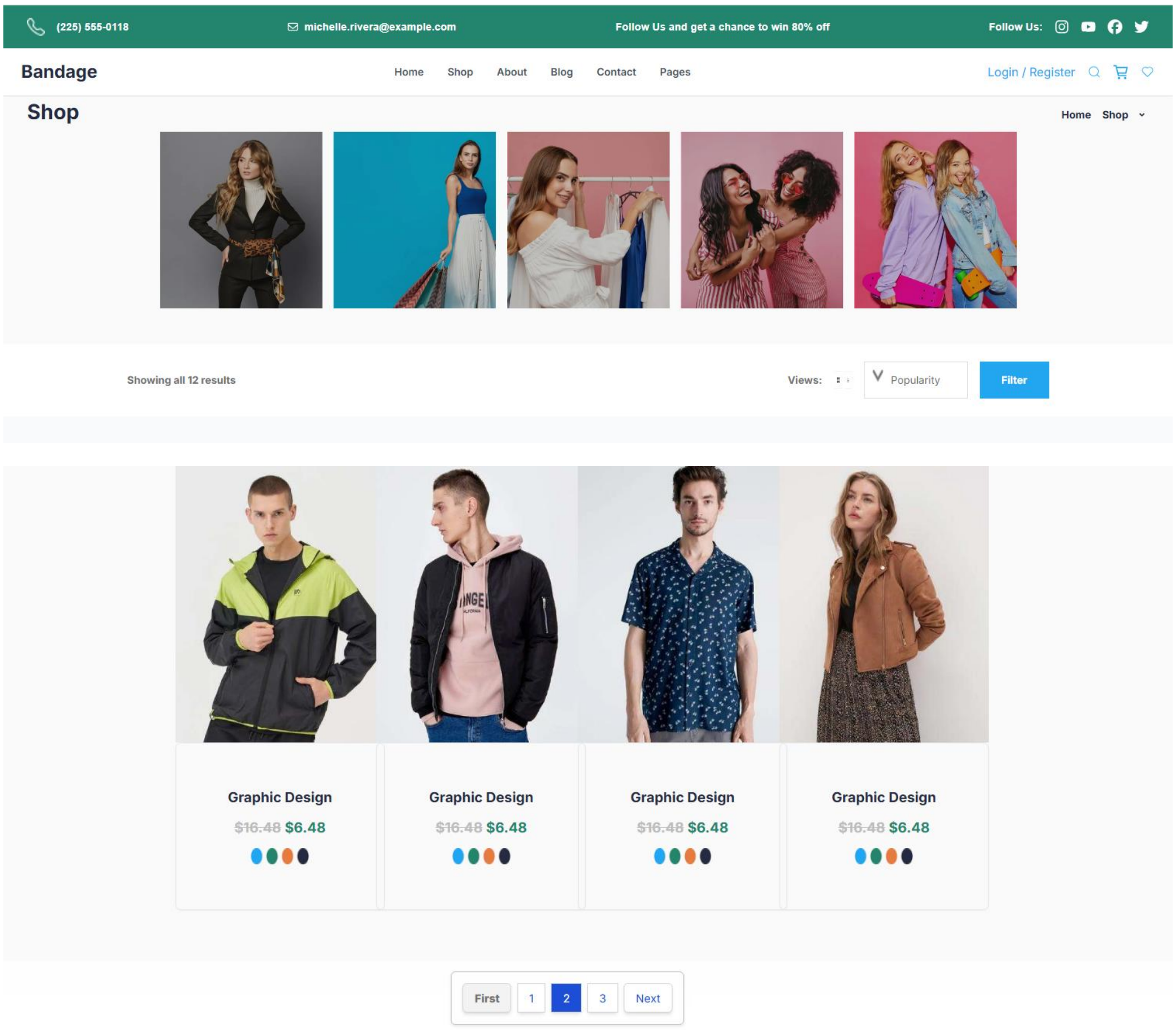
Timeless Elegance
Introducing TimelessElegance—a collection that embodies the perfect fusion of classic...
$320
Add to Cart



```
const ProductByTag: React.FC = () => {
  const fetchProducts = async () => {

      // Extract unique tags
      const allTags = data.flatMap((product) => product.tags);
      const uniqueTags = Array.from(new Set(allTags));
      setTags(uniqueTags);
    } catch (error) {
      console.error(error);
    }
  };

  useEffect(() => {
    fetchProducts();
  }, []);

  useEffect(() => {
    let updatedProducts = products;

    // Filter by tag
    if (selectedTag) {
      updatedProducts = updatedProducts.filter((product) =>
        product.tags.includes(selectedTag)
      );
    }

    // Filter by "New" badge
    if (isNewFilter !== null) {
      updatedProducts = updatedProducts.filter((product) => product.isNew === isNewFilter);
    }

    setFilteredProducts(updatedProducts);
  }, [selectedTag, isNewFilter, products]);

  const handleAddToCart = (product: Product) => {
    addToCart({
      ...product,
      quantity: 1,
      name: product.title,
      cartid: product._id,
    });

    alert(`${product.title} has been added to the cart!`);
  };
```

4. Pagination

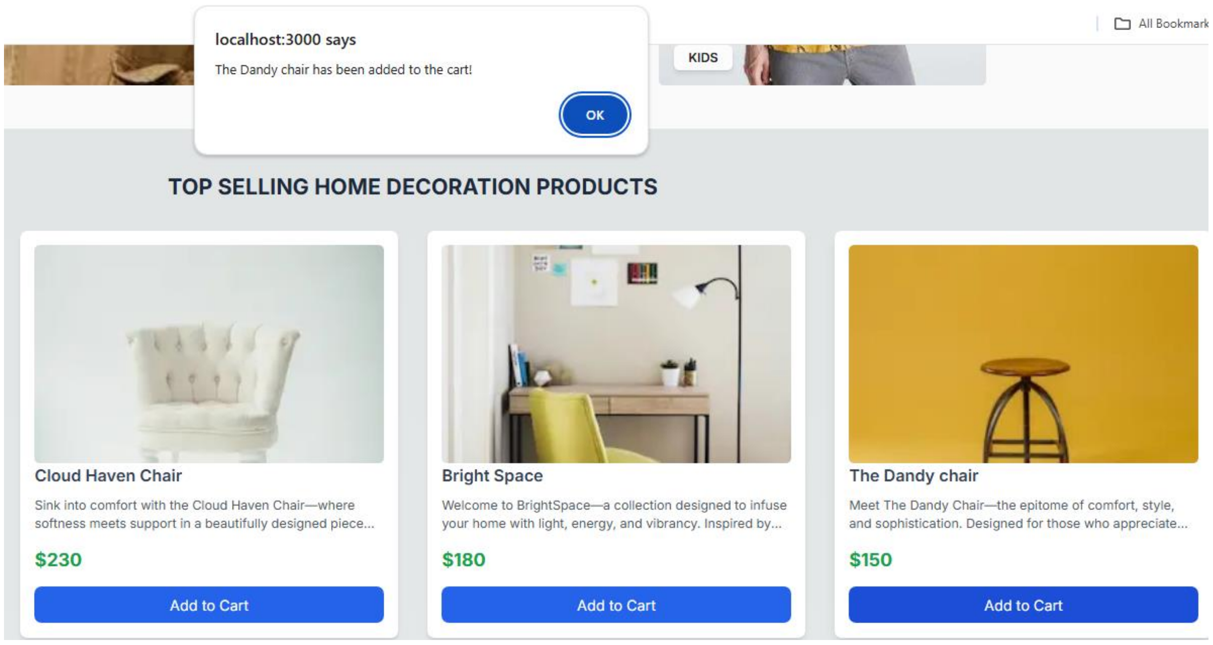
- **What I Did:**
  - Developed a PaginationComponent to divide products into smaller, navigable pages.
  - Styled navigation buttons for user-friendly interaction.
- **Challenges I Faced:**
  - Smooth page transitions.
- **Solution:**
  - I will solve it after hackathone.
- **Additional Features Can Be Added:**
  - Integrated page number highlighting to indicate the current page.
  - Used Next.js's getStaticProps and getServerSideProps for efficient data fetching.
  - Added "Jump to Page" functionality for quicker navigation.



5. Add to Cart

- **What I Did:**
  - Developed the cart functionality using the React Context API to manage the global state.
  - Enabled dynamic quantity updates for cart items.
- **Challenges I Faced:**
  - Syncing cart state across multiple components.
- **Solution:**
  - Used local storage to persist cart data.
- **Additional Features Added:**

- Integrated a cart summary that dynamically updates the total price and discounts



localhost:3000 says

The Dandy chair has been added to the cart!

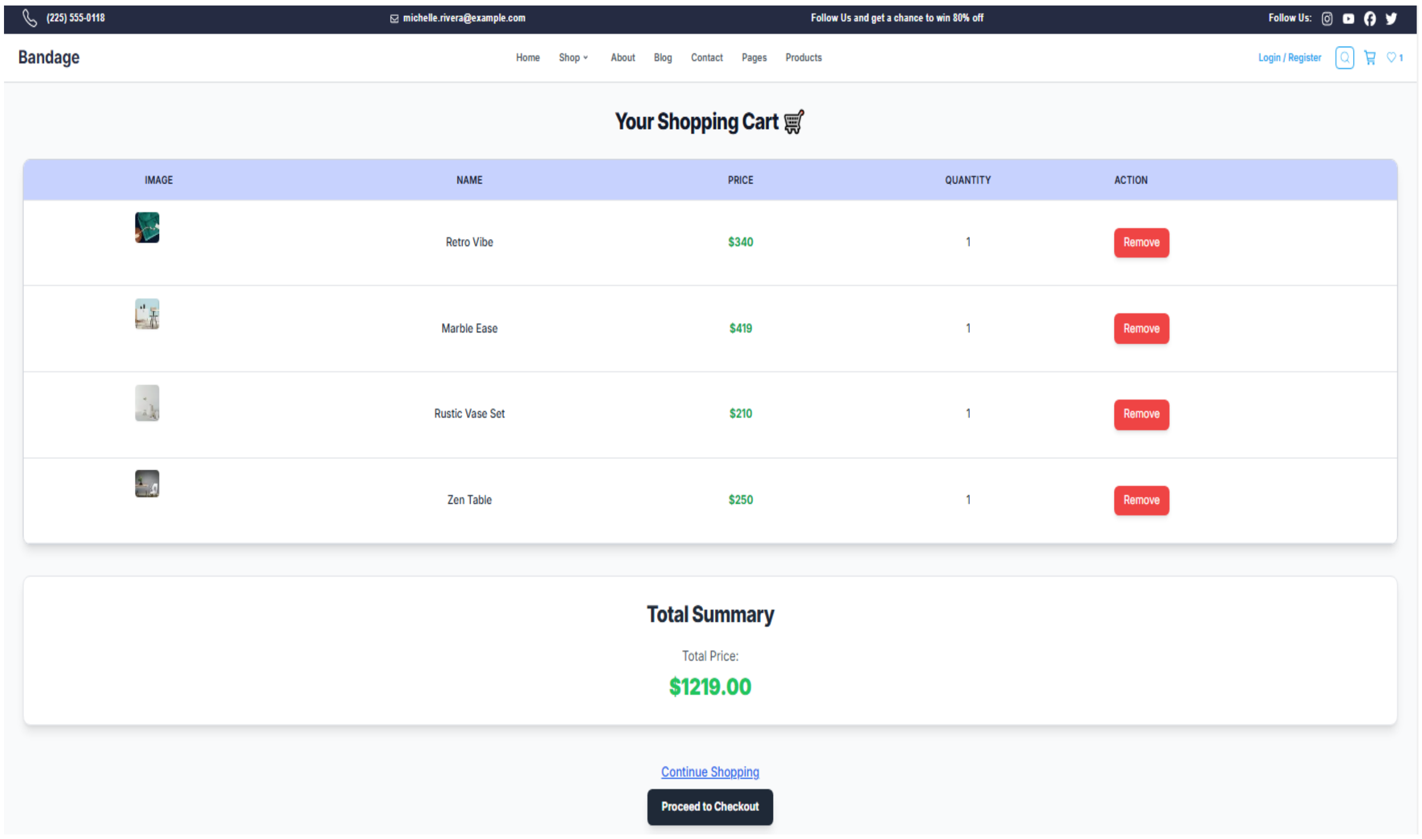
OK

TOP SELLING HOME DECORATION PRODUCTS

Cloud Haven Chair
Sink into comfort with the Cloud Haven Chair—where softness meets support in a beautifully designed piece...
$230
Add to Cart

Bright Space
Welcome to BrightSpace—a collection designed to infuse your home with light, energy, and vibrancy. Inspired by...
$180
Add to Cart

The Dandy chair
Meet The Dandy Chair—the epitome of comfort, style, and sophistication. Designed for those who appreciate...
$150
Add to Cart



Your Shopping Cart 🛒

IMAGE	NAME	PRICE	QUANTITY	ACTION
	Retro Vibe	$340	1	Remove
	Marble Ease	$419	1	Remove
	Rustic Vase Set	$210	1	Remove
	Zen Table	$250	1	Remove

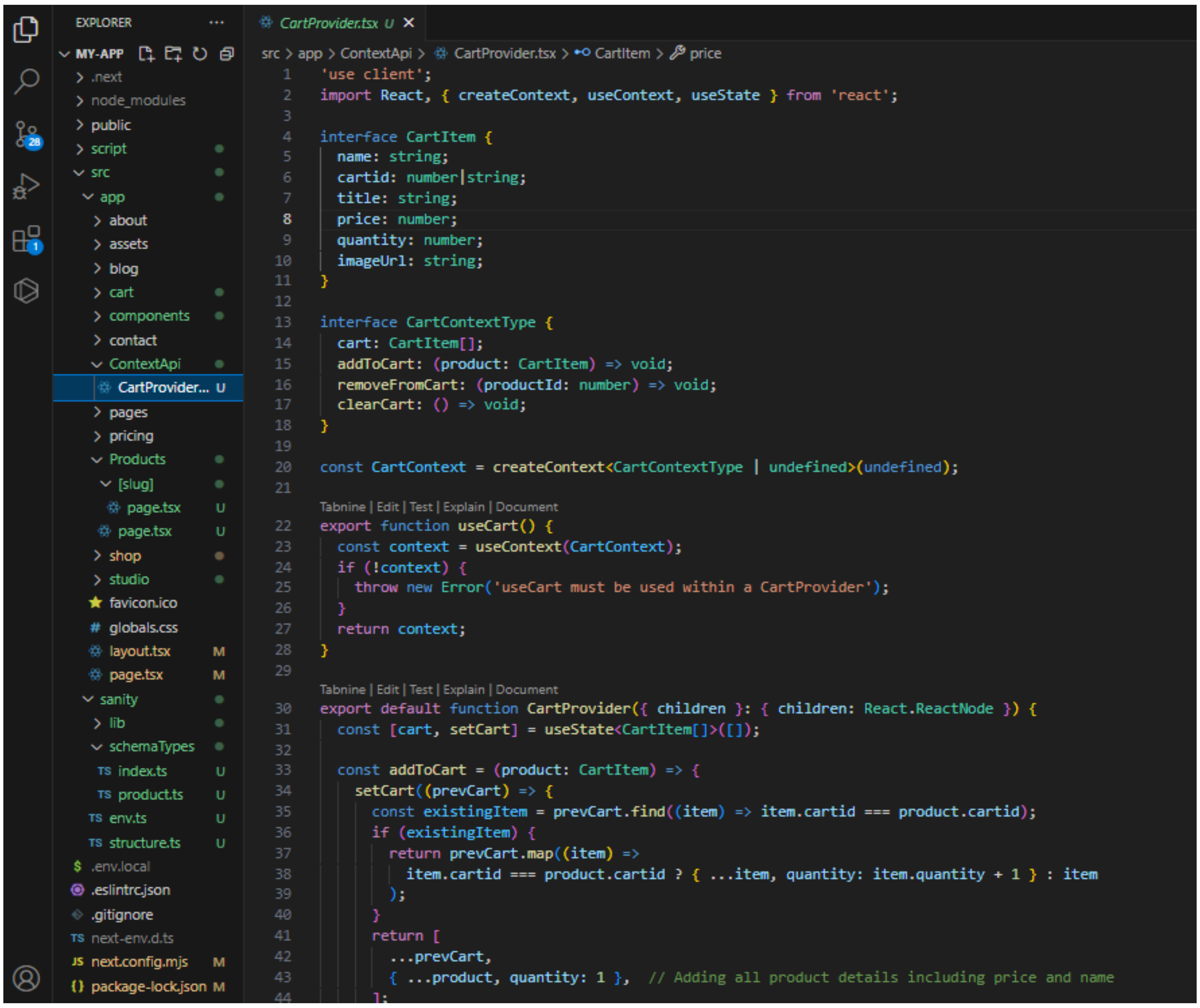
Total Summary

Total Price:

$1219.00

Continue Shopping

Proceed to Checkout



```tsx
'use client';
import React, { createContext, useContext, useState } from 'react';

interface CartItem {
  name: string;
  cartid: number|string;
  title: string;
  price: number;
  quantity: number;
  imageUrl: string;
}

interface CartContextType {
  cart: CartItem[];
  addToCart: (product: CartItem) => void;
  removeFromCart: (productId: number) => void;
  clearCart: () => void;
}

const CartContext = createContext<CartContextType | undefined>(undefined);

export function useCart() {
  const context = useContext(CartContext);
  if (!context) {
    throw new Error('useCart must be used within a CartProvider');
  }
  return context;
}

export default function CartProvider({ children }: { children: React.ReactNode }) {
  const [cart, setCart] = useState<CartItem[]>([]);

  const addToCart = (product: CartItem) => {
    setCart((prevCart) => {
      const existingItem = prevCart.find((item) => item.cartid === product.cartid);
      if (existingItem) {
        return prevCart.map((item) =>
          item.cartid === product.cartid ? { ...item, quantity: item.quantity + 1 } : item
        );
      }
      return [
        ...prevCart,
        { ...product, quantity: 1 },  // Adding all product details including price and name
      ];
```

6. Login and Sign-Up Page

- **What I Did:**
  - Built user authentication pages for login and sign-up with form validation.
  - Integrated Clerk for secure authentication and user management.
- **Challenges I Faced:**
  - Handling errors during login and sign-up.
- **Solution:**
  - Provided user-friendly error messages for better feedback.
- **Additional Features Can Be Added:**
  - Added "Forgot Password" functionality with email recovery support.
  - Enabled login via social accounts like Google.

Summary:

By the end of the development process, I successfully delivered the following:

1. A fully functional product listing page displaying dynamic data from the API.
2. Individual product detail pages implemented with dynamic routing.
3. Advanced filters for New Arrival and tags.
4. Pagination for better user experience with large datasets.
5. Responsive and professional styling for all components.
6. Modular and reusable components for future scalability.
7. Enhanced cart and user authentication functionality for a smoother shopping experience.