HACKATHON 3

DAY 4

Market place name: bandage

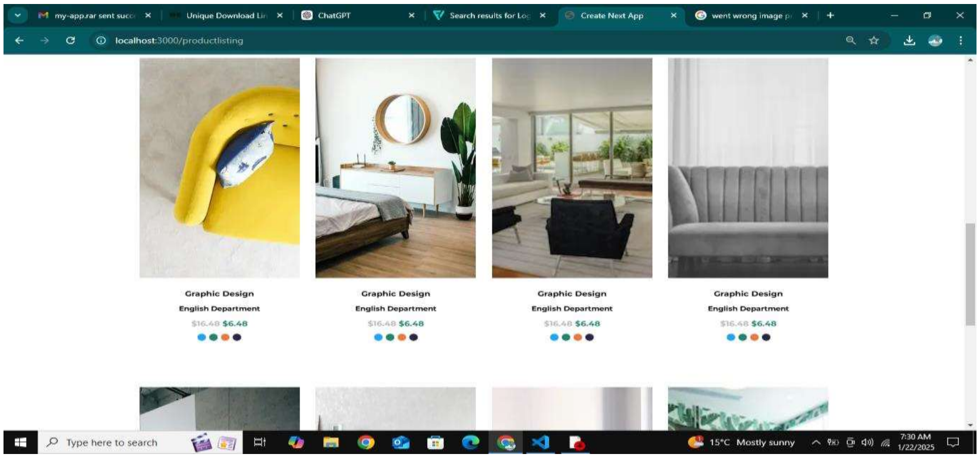
Objective

Today's task was all about creating the frontend components for our marketplace app. The goal was to make them dynamic, reusable, and responsive by fetching data from the Sanity CMS or APIs.

What I Did Today

1. **Product Listing Component:**
   a. I made a grid layout where products are shown with details like:
      - Name
      - Price
      - Image
      - Stock Status

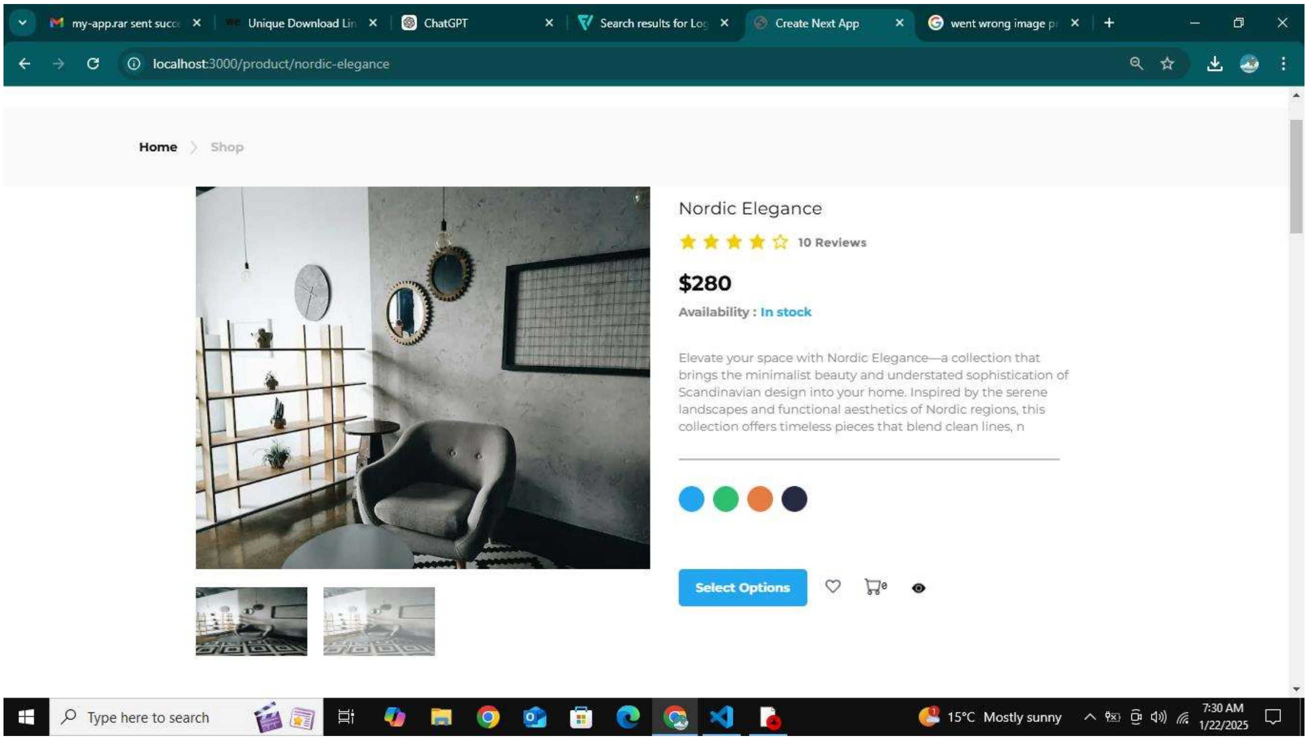
   b. I used a component to show each product. The data comes from the Sanity CMS API.



Product Detail Component:

- I made a dynamic page for each product using Next.js dynamic routing.

- This page shows:
  1. Product description.
  2. Price.
  3. Product name.
  4. Tags



Nordic Elegance

★★★★☆ 10 Reviews

$280

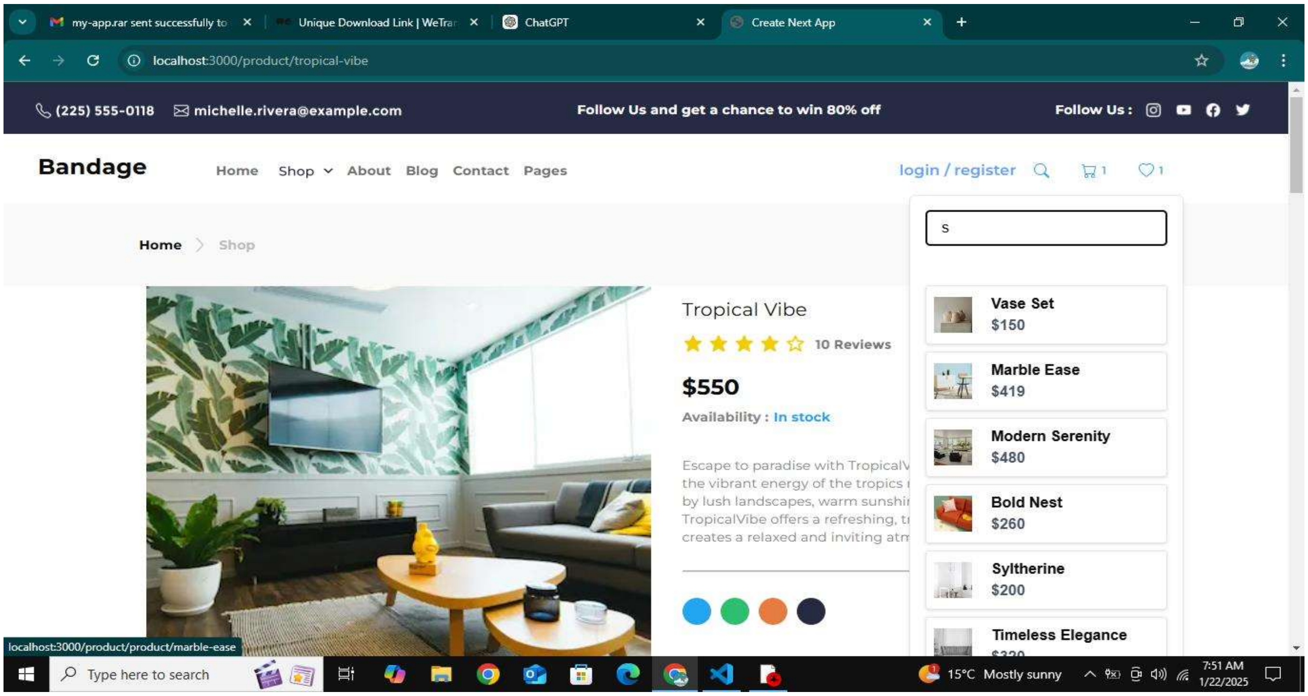
Availability : In stock

Elevate your space with Nordic Elegance—a collection that brings the minimalist beauty and understated sophistication of Scandinavian design into your home. Inspired by the serene landscapes and functional aesthetics of Nordic regions, this collection offers timeless pieces that blend clean lines, n

Search Bar:

- I added a search bar so users can find products by typing a name or tag.
- It updates the product list as you type.

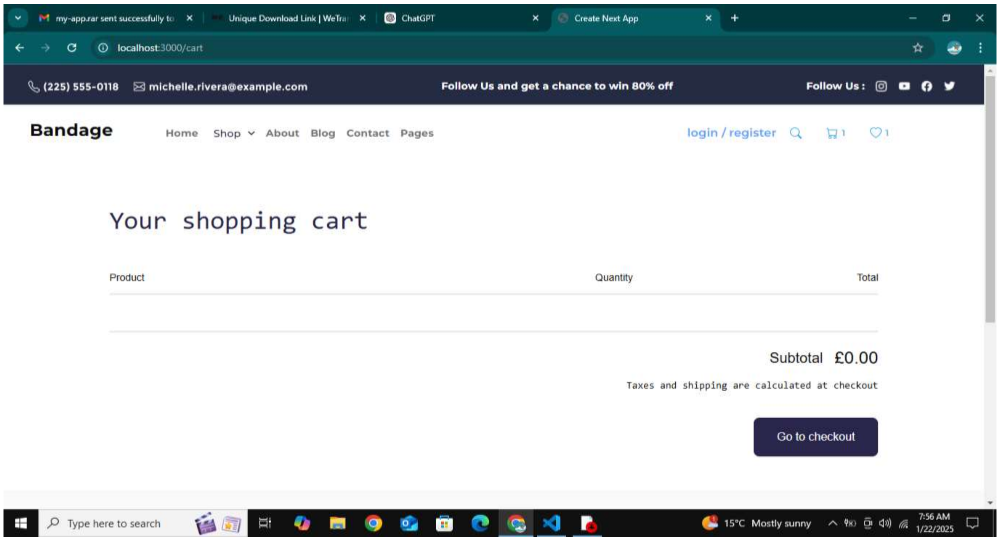
Search bar image:



Escape to paradise with TropicalV
the vibrant energy of the tropics
by lush landscapes, warm sunshi
TropicalVibe offers a refreshing, t
creates a relaxed and inviting atm

Cart Component:

- I made a cart where:
  1. Users can see the list of items they added to their cart.
  2. Quantities and total price are displayed.
- React state and client's local storage is used to keep track of the cart items.



Your shopping cart

Product	Quantity	Total

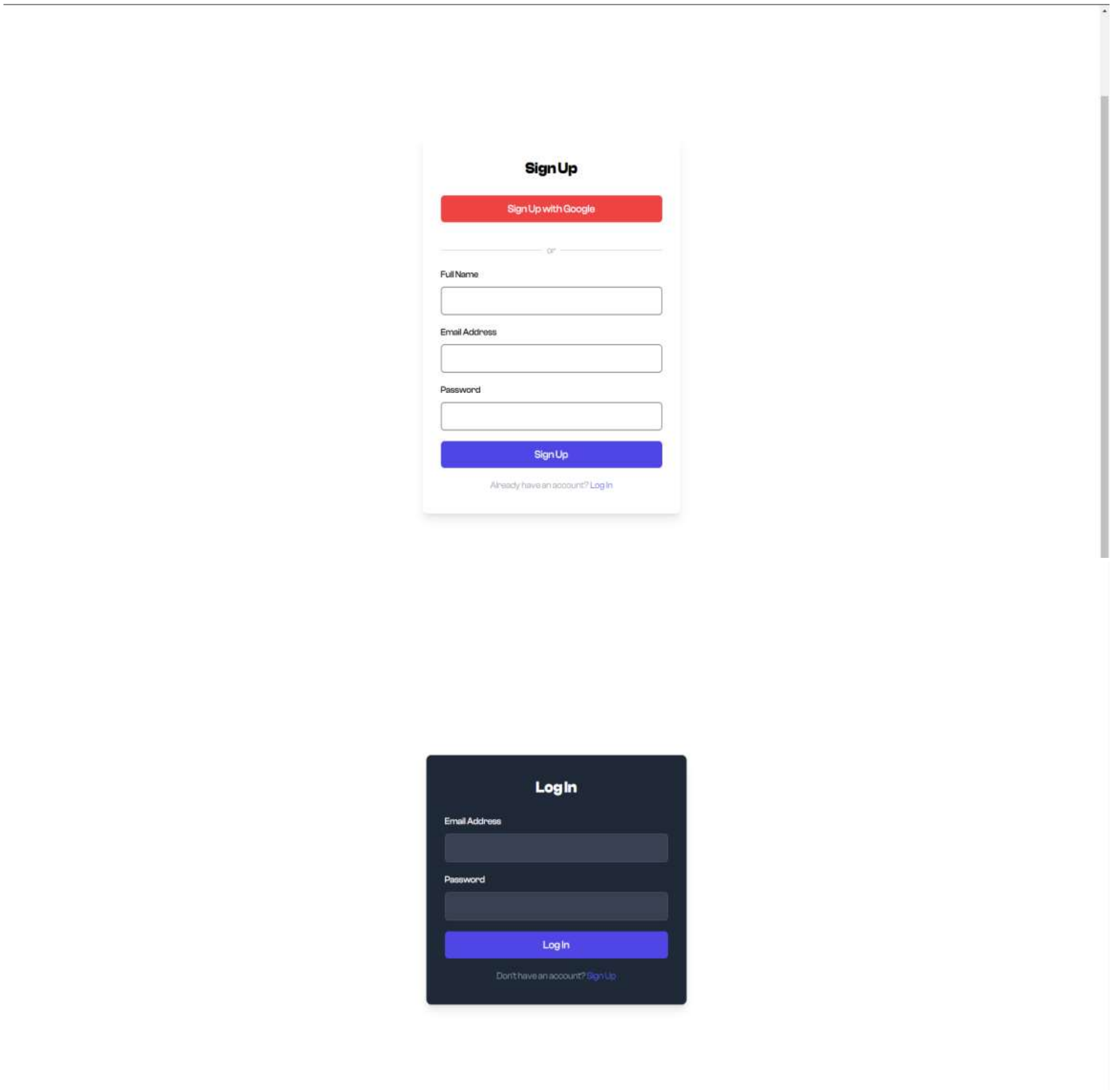
Subtotal £0.00

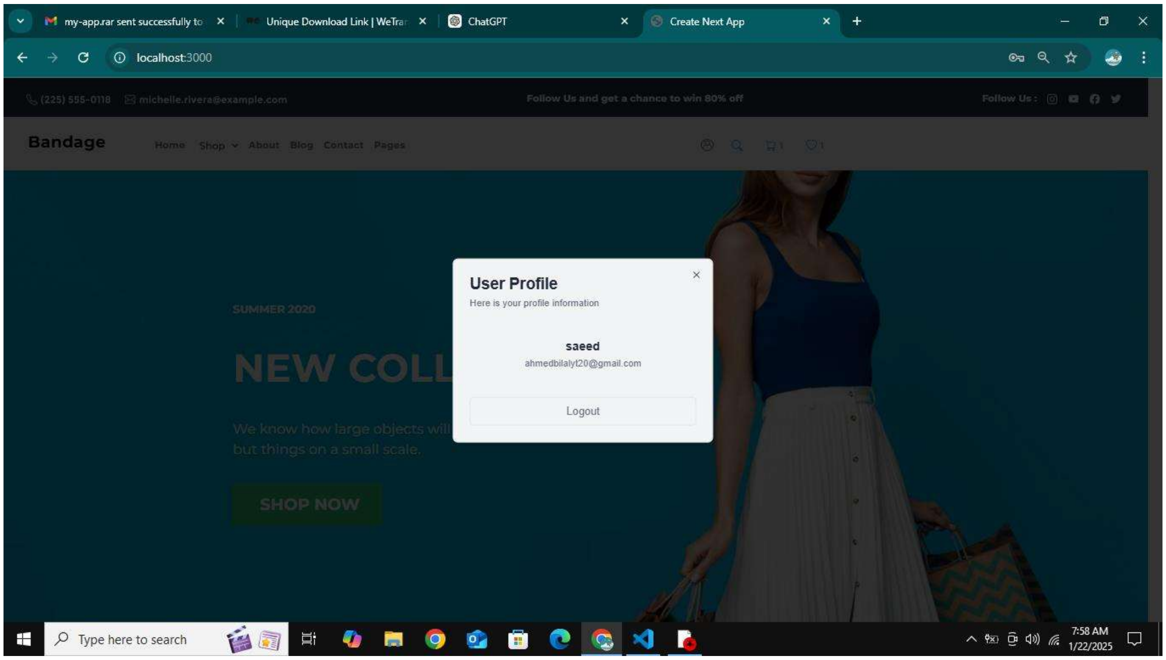
Taxes and shipping are calculated at checkout

User Authentication:

- I created a **Sign-Up Page** and a **Login Page** where users can register and log in.
- User can sign up using google account credentials.

- All of user's credentials and personal info is managed through the secure service from **firebase**
- I also made a **Profile Page** to display user details.




my-app.rar sent successfully to
Unique Download Link | WeTra
ChatGPT
Create Next App
localhost:3000
(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
SUMMER 2020
NEW COLL
We know how large objects will
but things on a small scale.
SHOP NOW
Type here to search
7:58 AM
1/22/2025

User Profile

Here is your profile information

saeed

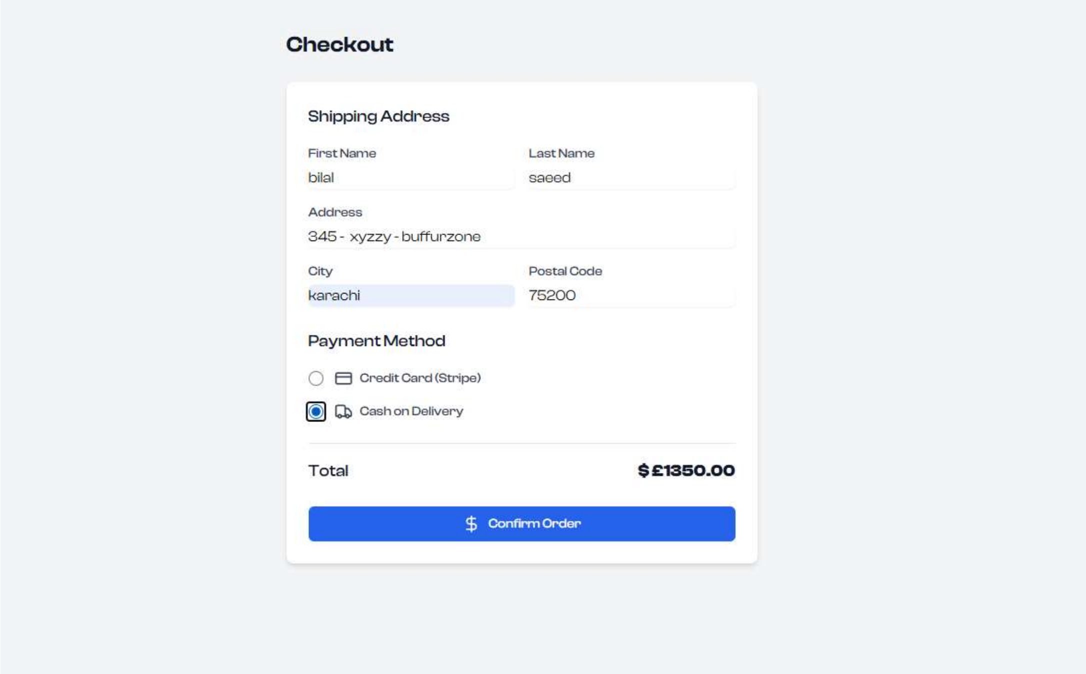
ahmedbilalyt20@gmail.com

Logout

Checkout Flow Component:

- Added a checkout flow component page to the website to streamline the purchasing process.
- The component collects user information such as:
  1. First name
  2. Last name
  3. Address
  4. Cart items (automatically form the users cart)
- User data is securely stored on Firebase for reliability and scalability.
- Implemented an order confirmation page that:
  1. Fetches stored user data from Firebase.
  2. Displays a personalized thank-you message with the user's name.
  3. Shows the user's entered details and a summary of the ordered products.
- This flow simplifies the checkout process and adds a personal touch to enhance user experience.

Checkout page:



Checkout

Shipping Address

First Name: bilal

Last Name: saeed

Address: 345 - xyzzy - buffurzone

City: karachi

Postal Code: 75200

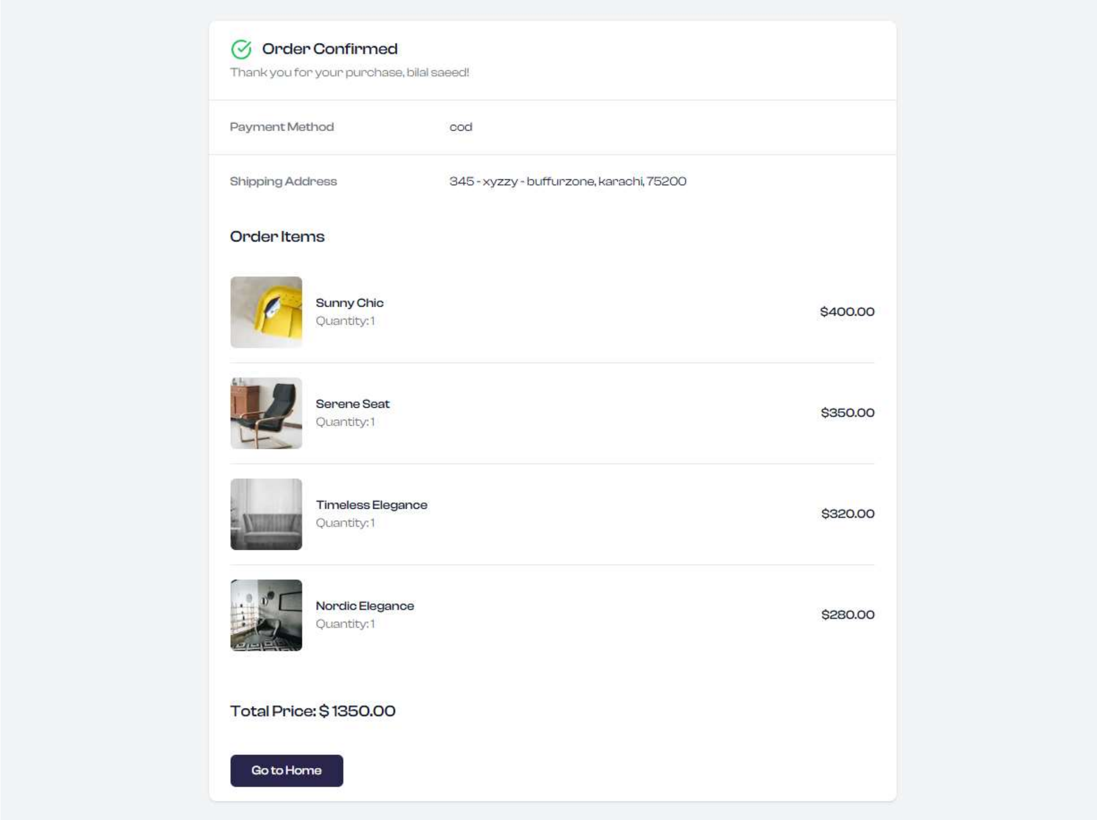
Payment Method

○ Credit Card (Stripe)

● Cash on Delivery

Total **$£1350.00**

Order confirmation page:



✓ Order Confirmed

Thank you for your purchase, bilal saeed!

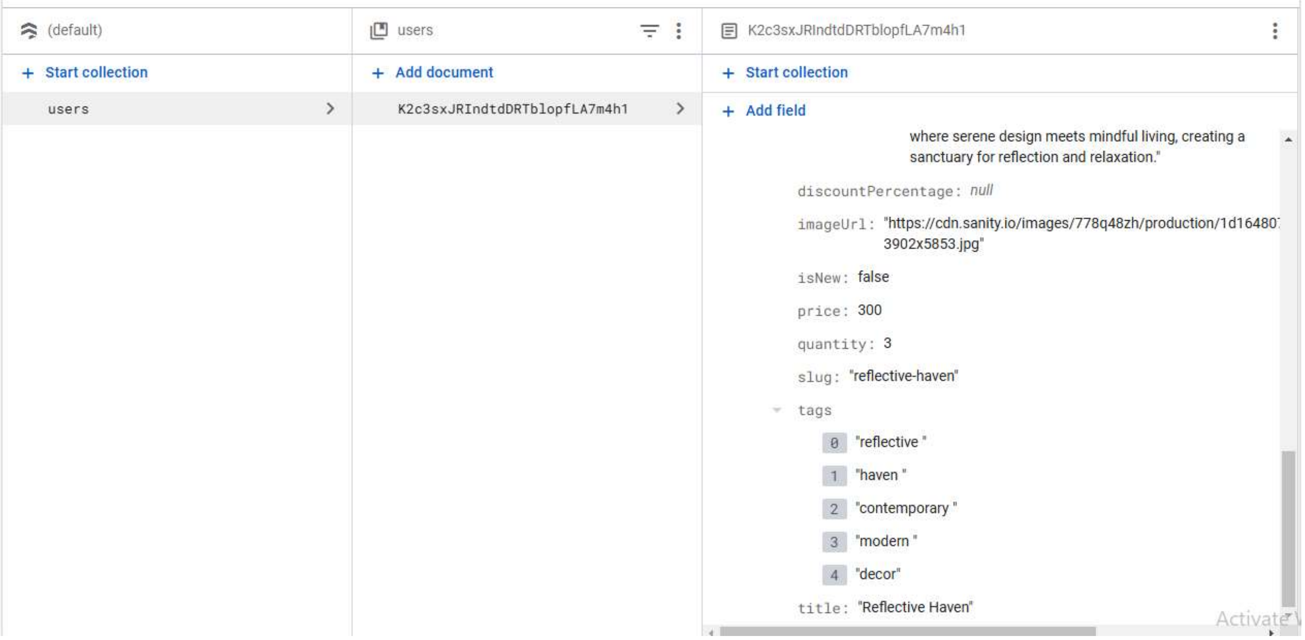
Payment Method	cod
Shipping Address	345 - xyzzy - buffurzone, karachi, 75200

Order Items

Sunny Chic Quantity: 1	$400.00
Serene Seat Quantity: 1	$350.00
Timeless Elegance Quantity: 1	$320.00
Nordic Elegance Quantity: 1	$280.00

Total Price: $ 1350.00

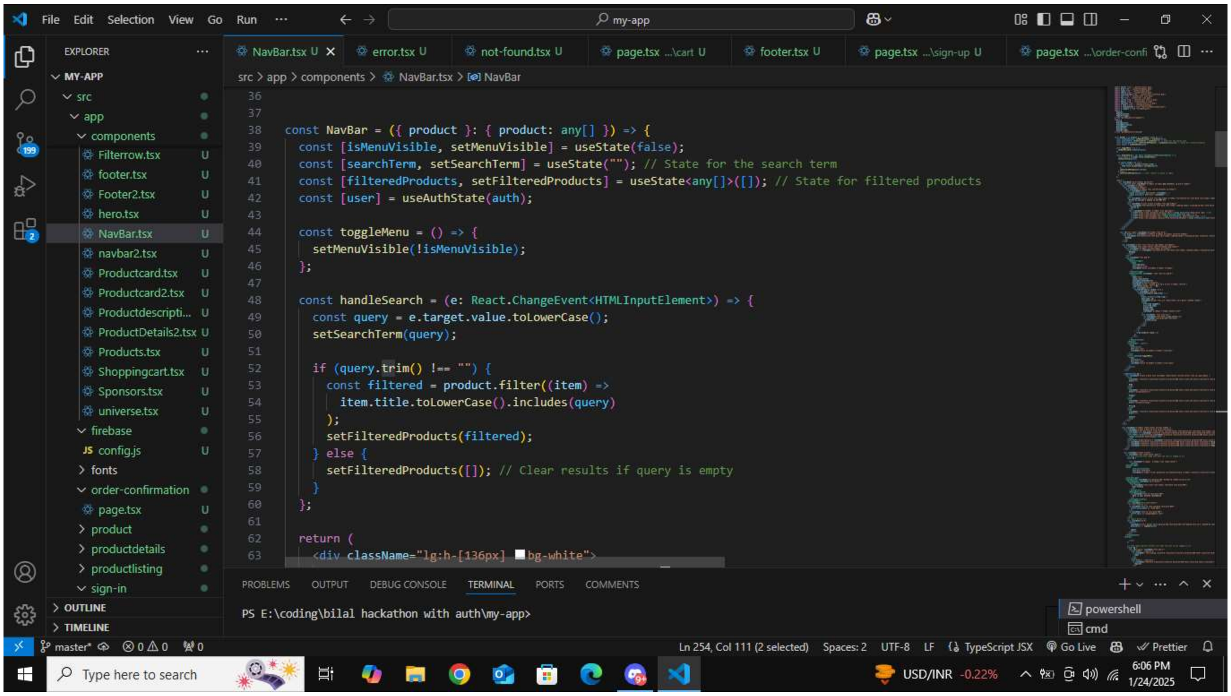
Order on firebase database:



```
                 where serene design meets mindful living, creating a
                 sanctuary for reflection and relaxation."
discountPercentage: null
imageUrl: "https://cdn.sanity.io/images/778q48zh/production/1d16480
          3902x5853.jpg"
isNew: false
price: 300
quantity: 3
slug: "reflective-haven"
tags
   0 "reflective "
   1 "haven "
   2 "contemporary "
   3 "modern "
   4 "decor"
title: "Reflective Haven"
```

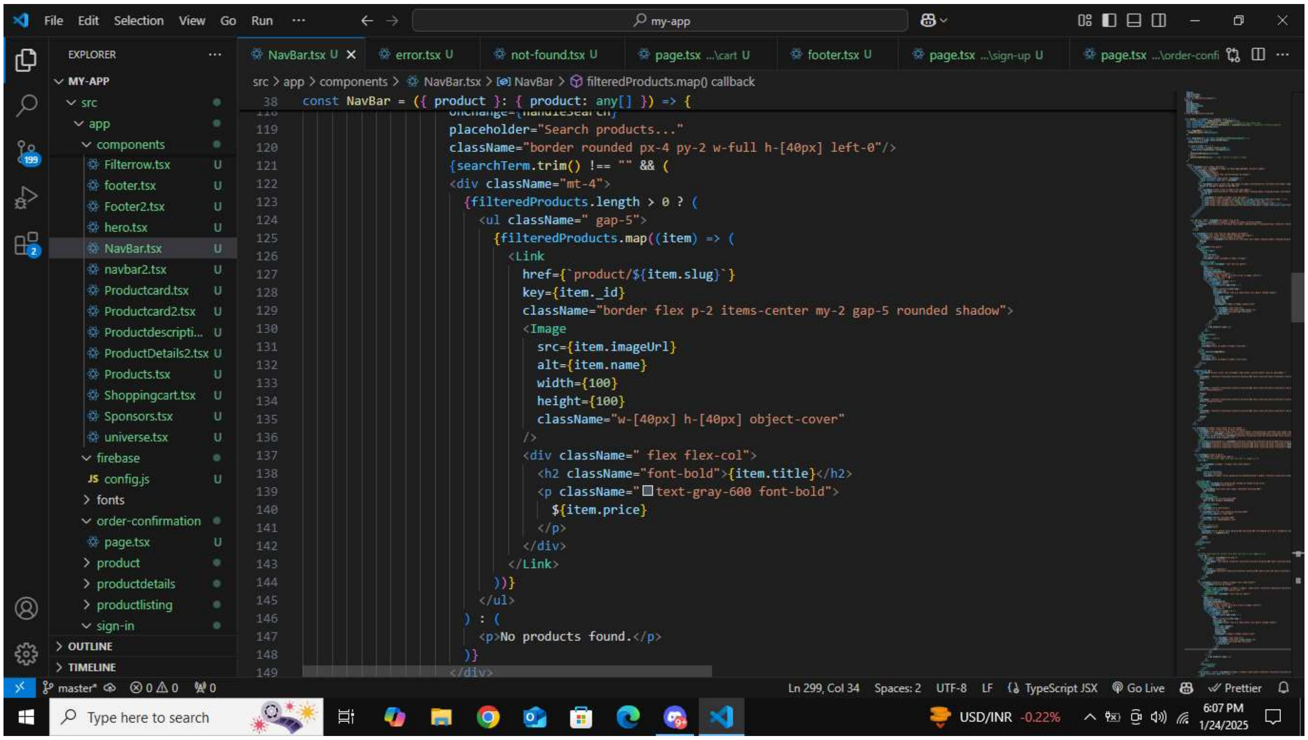
Snippets of the code

Search engine code:



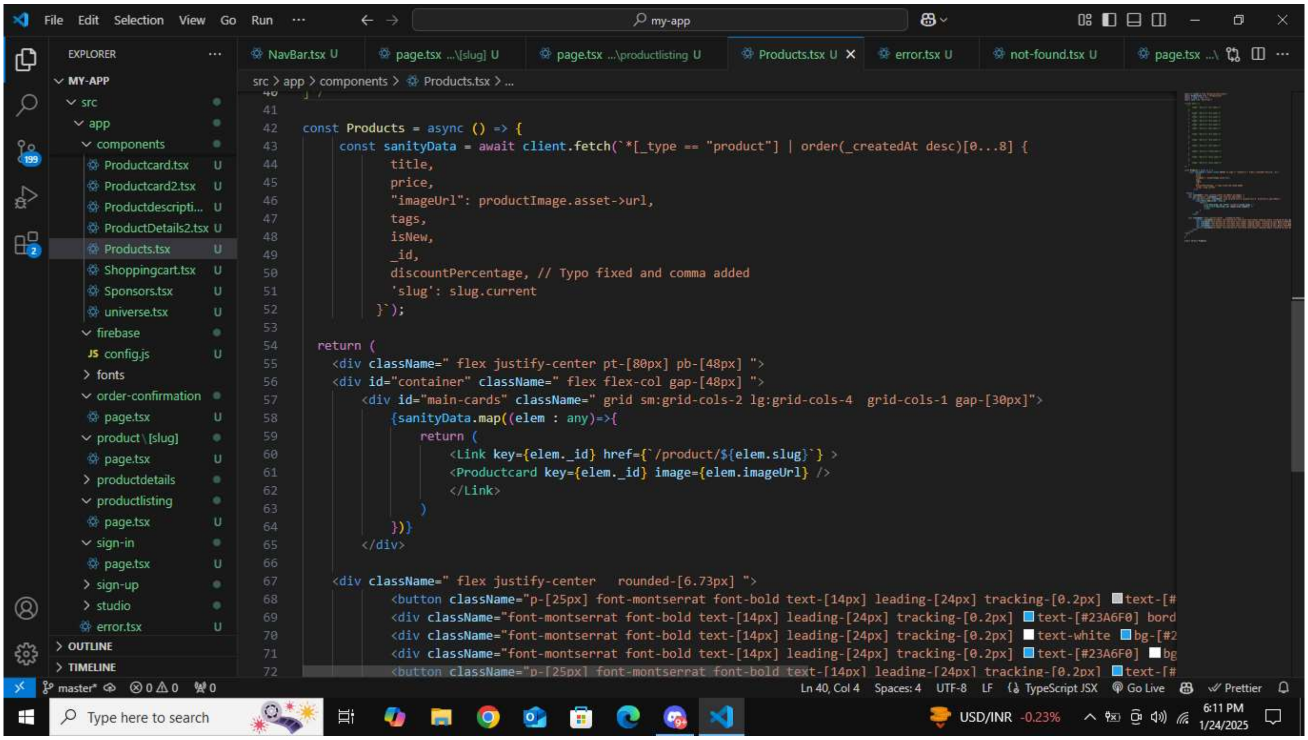
```
const NavBar = ({ product }: { product: any[] }) => {
  const [isMenuVisible, setMenuVisible] = useState(false);
  const [searchTerm, setSearchTerm] = useState(""); // State for the search term
  const [filteredProducts, setFilteredProducts] = useState<any[]>([]); // State for filtered products
  const [user] = useAuthState(auth);

  const toggleMenu = () => {
    setMenuVisible(!isMenuVisible);
  };

  const handleSearch = (e: React.ChangeEvent<HTMLInputElement>) => {
    const query = e.target.value.toLowerCase();
    setSearchTerm(query);

    if (query.trim() !== "") {
      const filtered = product.filter((item) =>
        item.title.toLowerCase().includes(query)
      );
      setFilteredProducts(filtered);
    } else {
      setFilteredProducts([]); // Clear results if query is empty
    }
  };

  return (
    <div className="lg:h-[136px] bg-white">
```

```
PS E:\coding\bilal hackathon with auth\my-app>
```



```
const NavBar = ({ product }: { product: any[] }) => {
            placeholder="Search products..."
            className="border rounded px-4 py-2 w-full h-[40px] left-0"/>
            {searchTerm.trim() !== "" && (
            <div className="mt-4">
              {filteredProducts.length > 0 ? (
                <ul className=" gap-5">
                  {filteredProducts.map((item) => (
                    <Link
                      href={`product/${item.slug}`}
                      key={item._id}
                      className="border flex p-2 items-center my-2 gap-5 rounded shadow">
                      <Image
                        src={item.imageUrl}
                        alt={item.name}
                        width={100}
                        height={100}
                        className="w-[40px] h-[40px] object-cover"
                      />
                      <div className=" flex flex-col">
                        <h2 className="font-bold">{item.title}</h2>
                        <p className="text-gray-600 font-bold">
                          ${item.price}
                        </p>
                      </div>
                    </Link>
                  ))}
                </ul>
              ) : (
                <p>No products found.</p>
              )}
            </div>
```

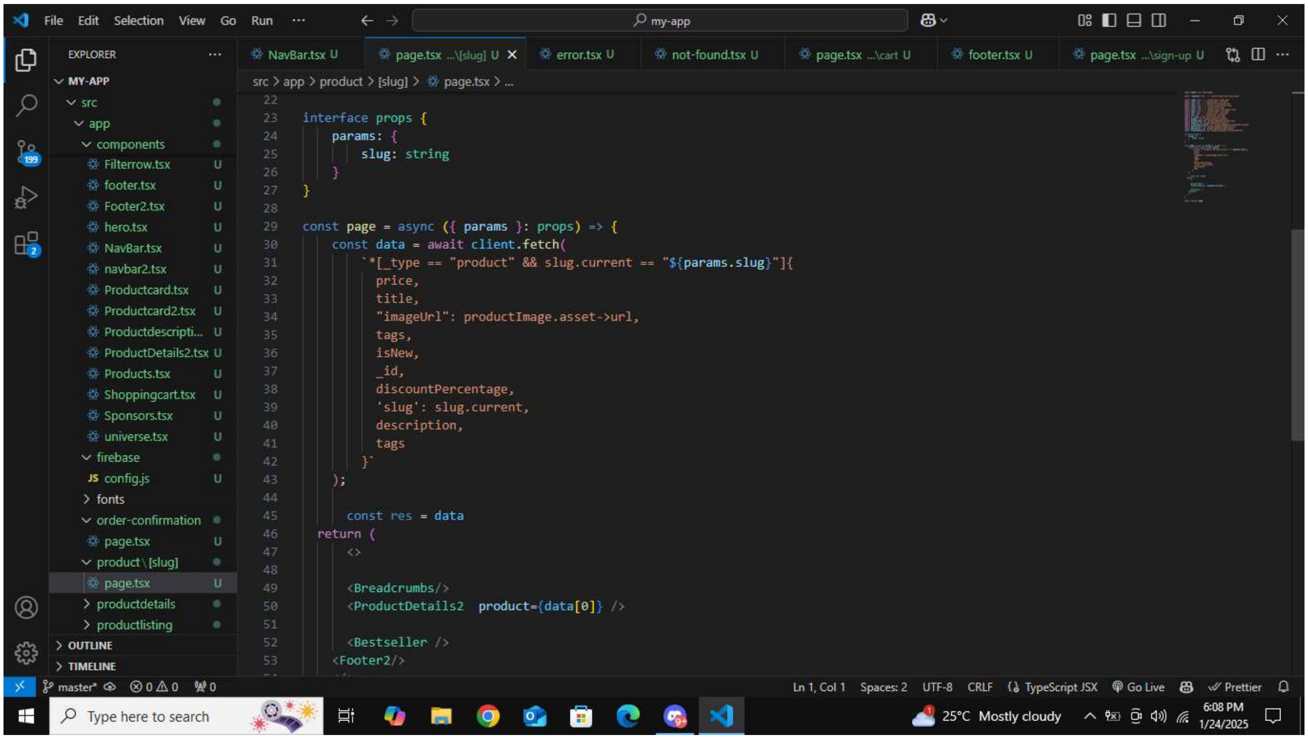
Product list code:


my-app
NavBar.tsx U
page.tsx ...\[slug] U
page.tsx ...\productlisting U
Products.tsx U
error.tsx U
not-found.tsx U
src > app > components > Products.tsx > ...
Ln 40, Col 4 Spaces: 4 UTF-8 LF TypeScript JSX Go Live Prettier
USD/INR -0.23%
6:11 PM
1/24/2025

```
const Products = async () => {
    const sanityData = await client.fetch(`*[_type == "product"] | order(_createdAt desc)[0...8] {
          title,
          price,
          "imageUrl": productImage.asset->url,
          tags,
          isNew,
          _id,
          discountPercentage, // Typo fixed and comma added
          'slug': slug.current
        }`);

  return (
    <div className=" flex justify-center pt-[80px] pb-[48px] ">
    <div id="container" className=" flex flex-col gap-[48px] ">
        <div id="main-cards" className=" grid sm:grid-cols-2 lg:grid-cols-4  grid-cols-1 gap-[30px]">
            {sanityData.map((elem : any)=>{
                return (
                    <Link key={elem._id} href={`/product/${elem.slug}`} >
                    <Productcard key={elem._id} image={elem.imageUrl} />
                    </Link>
                )
            })}
        </div>

    <div className=" flex justify-center   rounded-[6.73px] ">
        <button className="p-[25px] font-montserrat font-bold text-[14px] leading-[24px] tracking-[0.2px] text-[#
        <div className="font-montserrat font-bold text-[14px] leading-[24px] tracking-[0.2px] text-[#23A6F0] bord
        <div className="font-montserrat font-bold text-[14px] leading-[24px] tracking-[0.2px] text-white bg-[#2
        <div className="font-montserrat font-bold text-[14px] leading-[24px] tracking-[0.2px] text-[#23A6F0] bg
        <button className="p-[25px] font-montserrat font-bold text-[14px] leading-[24px] tracking-[0.2px] text-[#
```

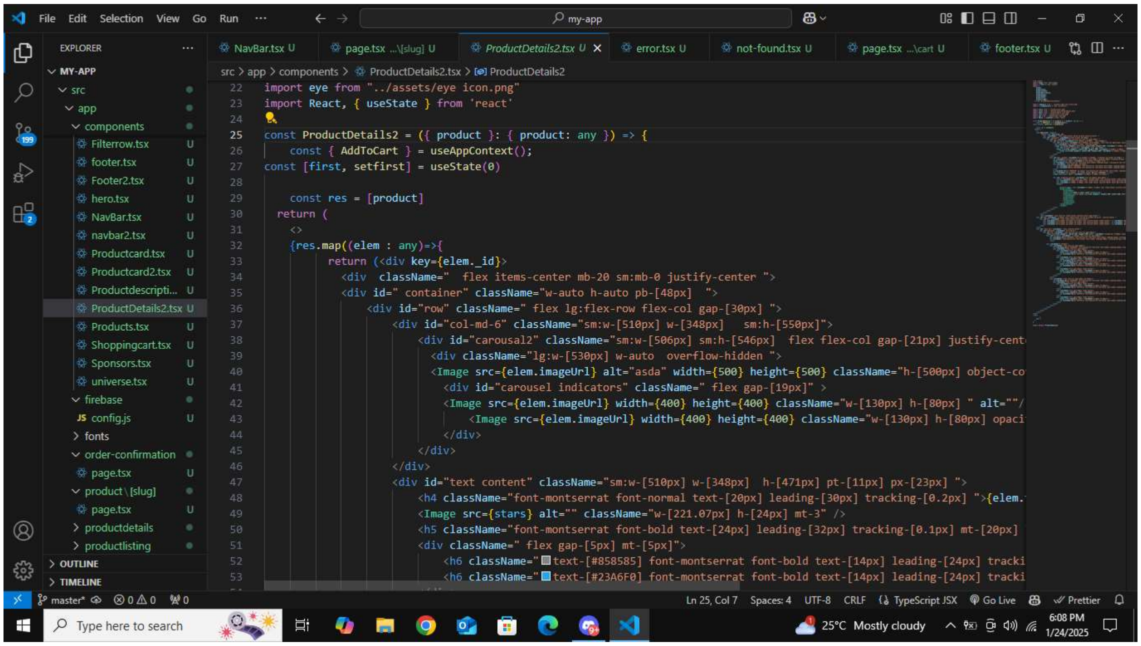
Dynamic Product details slug code:

Data is being fetched and passed to the product component.



```tsx

interface props {
    params: {
        slug: string
    }
}

const page = async ({ params }: props) => {
    const data = await client.fetch(
        `*[_type == "product" && slug.current == "${params.slug}"]{
          price,
          title,
          "imageUrl": productImage.asset->url,
          tags,
          isNew,
          _id,
          discountPercentage,
          'slug': slug.current,
          description,
          tags
        }`
    );

      const res = data
   return (
      <>

      <Breadcrumbs/>
      <ProductDetails2  product={data[0]} />

      <Bestseller />
    <Footer2/>
```

Product component code:



src > app > components > ProductDetails2.tsx > ProductDetails2

```tsx
import eye from "../assets/eye icon.png"
import React, { useState } from 'react'

const ProductDetails2 = ({ product }: { product: any }) => {
    const { AddToCart } = useAppContext();
const [first, setfirst] = useState(0)

    const res = [product]
  return (
    <>
    {res.map((elem : any)=>{
          return (<div key={elem._id}>
            <div  className="  flex items-center mb-20 sm:mb-0 justify-center ">
            <div id=" container" className="w-auto h-auto pb-[48px]  ">
                <div id="row" className=" flex lg:flex-row flex-col gap-[30px] ">
                    <div id="col-md-6" className="sm:w-[510px] w-[348px]   sm:h-[550px]">
                        <div id="carousal2" className="sm:w-[506px] sm:h-[546px]  flex flex-col gap-[21px] justify-cent
                          <div className="lg:w-[530px] w-auto  overflow-hidden ">
                          <Image src={elem.imageUrl} alt="asda" width={500} height={500} className="h-[500px] object-co
                            <div id="carousel indicators" className=" flex gap-[19px]" >
                            <Image src={elem.imageUrl} width={400} height={400} className="w-[130px] h-[80px] " alt=""/
                                <Image src={elem.imageUrl} width={400} height={400} className="w-[130px] h-[80px] opaci
                            </div>
                        </div>
                    </div>
                    <div id="text content" className="sm:w-[510px] w-[348px]  h-[471px] pt-[11px] px-[23px] ">
                        <h4 className="font-montserrat font-normal text-[20px] leading-[30px] tracking-[0.2px] ">{elem.
                        <Image src={stars} alt="" className="w-[221.07px] h-[24px] mt-3" />
                        <h5 className="font-montserrat font-bold text-[24px] leading-[32px] tracking-[0.1px] mt-[20px]
                        <div className=" flex gap-[5px] mt-[5px]">
                            <h6 className="text-[#858585] font-montserrat font-bold text-[14px] leading-[24px] tracki
                            <h6 className="text-[#23A6F0] font-montserrat font-bold text-[14px] leading-[24px] tracki
```