

MARKETPLACE BUSSINESS PLAN DAY 4

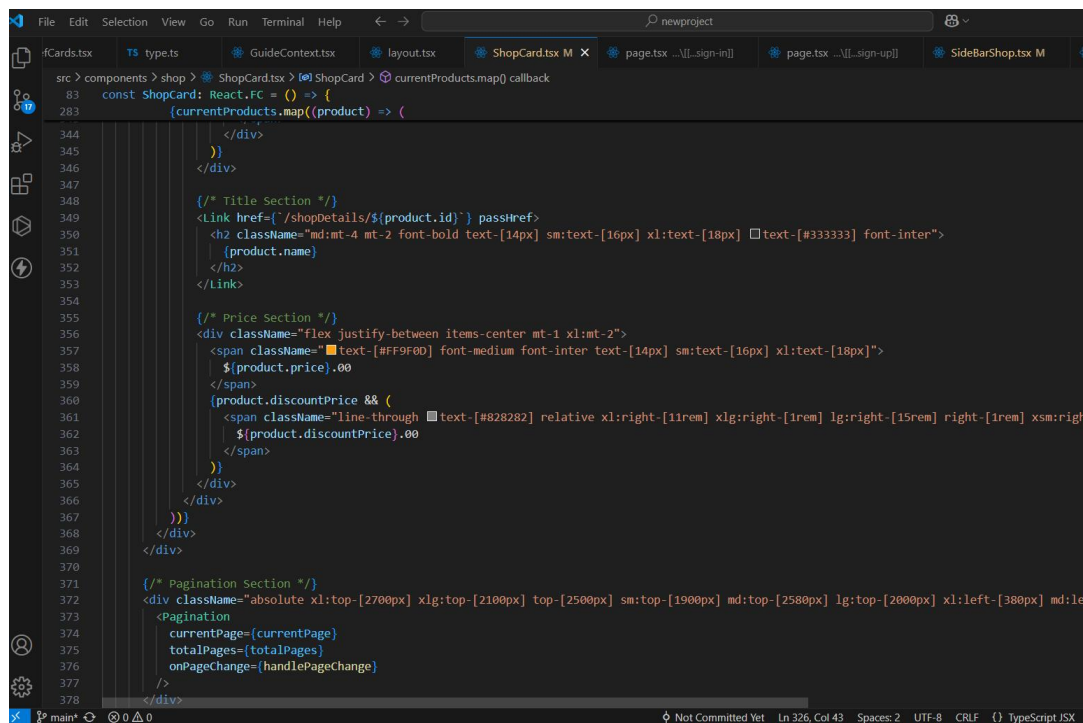
Building Dynamic Frontend Components for my Marketplace – Foodtuck

In last 3 days, I've focused on how we can plan our marketplace and how we can start our Q-commerce business after that all concepts I'm going to create my hackathon in dynamic way how we can do dynamic routing , product listing and many other components. For this I used dynamic and modular components which is reusable. So here is my essential components for my hackathon . I'll explain how I created components and I'll also share my code snippets. Here is breakdown of my components;

Building key Components

1. Product Listing Page:

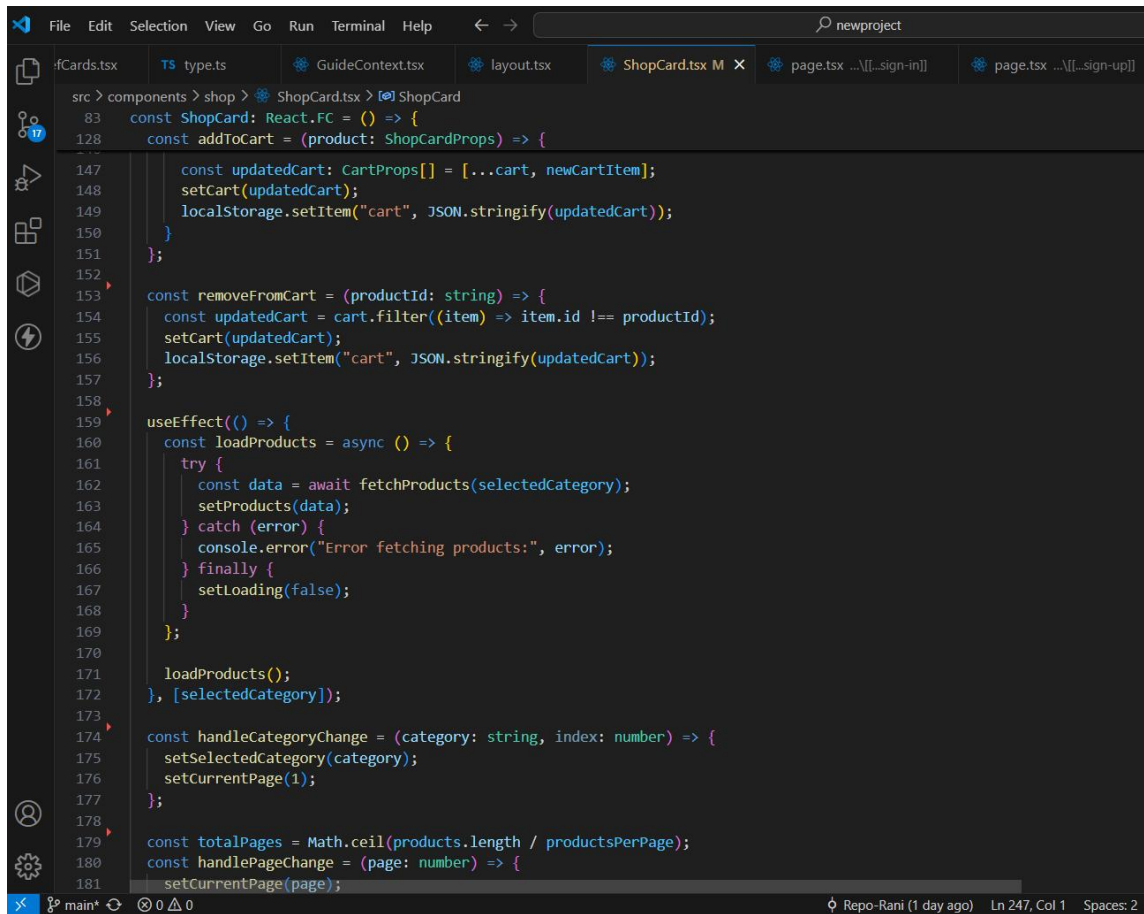
I've build product listing page dynamically where I've used grid layout for the professional work flow;



```
src > components > shop > ShopCard.tsx > ShopCard > currentProducts.map() callback
83  const ShopCard: React.FC = () => {
283  (currentProducts.map((product) => {
344      </div>
345  })
346  </div>
347
348  /* Title Section */
349  <Link href={`/${shopDetails}/${product.id}`} passHref>
350  <h2 className="md:mt-4 mt-2 font-bold text-[14px] sm:text-[16px] xl:text-[18px] text-[#333333] font-inter">
351  {product.name}
352  </h2>
353  </Link>
354
355  /* Price Section */
356  <div className="flex justify-between items-center mt-1 xl:mt-2">
357  <span className="text-[#FF9F00] font-medium font-inter text-[14px] sm:text-[16px] xl:text-[18px]">
358  ${product.price}.00
359  </span>
360  {product.discountPrice && (
361  <span className="line-through text-[#828282] relative xl:right-[11rem] xlg:right-[1rem] lg:right-[15rem] right-[1rem] xsm:right-[11rem] font-medium font-inter text-[14px] sm:text-[16px] xl:text-[18px]">
362  ${product.discountPrice}.00
363  </span>
364  )}
365  </div>
366  </div>
367  ))
368  </div>
369  </div>
370
371  /* Pagination Section */
372  <div className="absolute xl:top-[2700px] xlg:top-[2100px] top-[2500px] sm:top-[1900px] md:top-[2580px] lg:top-[2000px] xl:left-[380px] md:left-[380px] font-medium font-inter text-[14px] sm:text-[16px] xl:text-[18px]">
373  <Pagination
374  currentPage={currentPage}
375  totalPages={totalPages}
376  onPageChange={handlePageChange}
377  />
378  </div>
```

In **card** component I've design my card style where I've style my image, name, price etc...

In Shop Card component I've used Groq query for fetching data from sanity CMS and it is displaying dynamically.

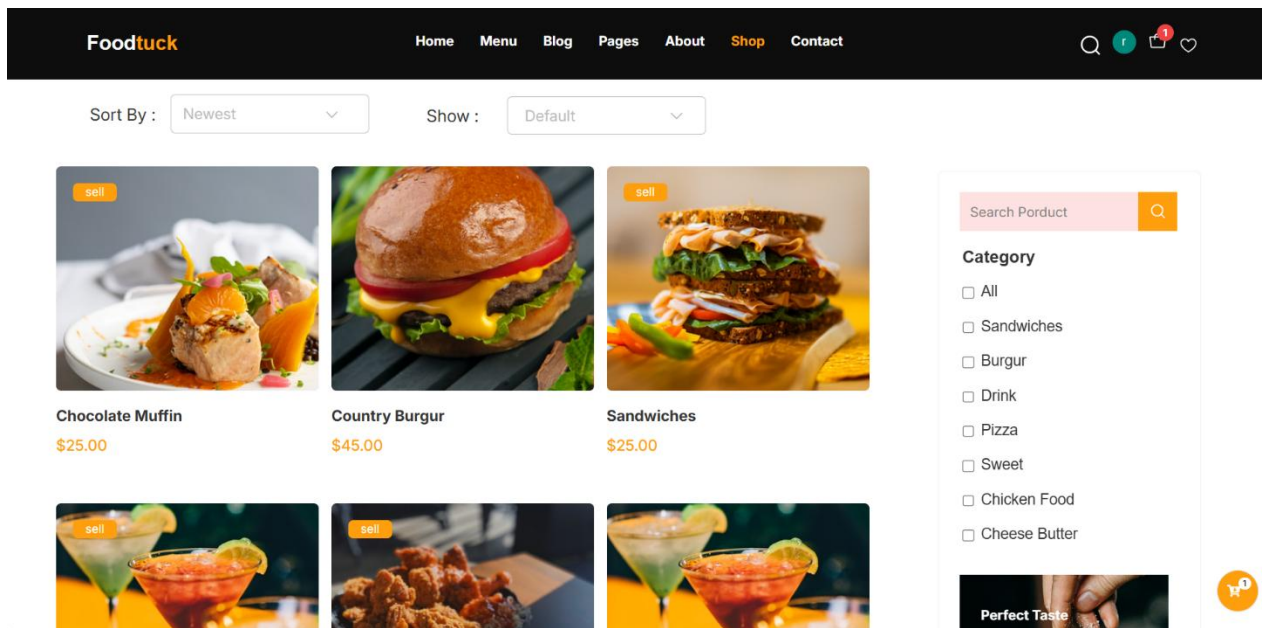


```
src > components > shop > ShopCard.tsx > ShopCard
83  const ShopCard: React.FC = () => {
128  const addToCart = (product: ShopCardProps) => {
147      const updatedCart: CartProps[] = [...cart, newCartItem];
148      setCart(updatedCart);
149      localStorage.setItem("cart", JSON.stringify(updatedCart));
150  }
151  };
152
153  const removeFromCart = (productId: string) => {
154      const updatedCart = cart.filter((item) => item.id !== productId);
155      setCart(updatedCart);
156      localStorage.setItem("cart", JSON.stringify(updatedCart));
157  };
158
159  useEffect(() => {
160      const loadProducts = async () => {
161          try {
162              const data = await fetchProducts(selectedCategory);
163              setProducts(data);
164          } catch (error) {
165              console.error("Error fetching products:", error);
166          } finally {
167              setLoading(false);
168          }
169      };
170
171      loadProducts();
172  }, [selectedCategory]);
173
174  const handleCategoryChange = (category: string, index: number) => {
175      setSelectedCategory(category);
176      setCurrentPage(1);
177  };
178
179  const totalPages = Math.ceil(products.length / productsPerPage);
180  const handlePageChange = (page: number) => {
181      setCurrentPage(page);
182  };
183  };
```

So in short in this step I've fetch data dynamically form sanity CMS and using groq query and grid layout for responsiveness.

```
File Edit Selection View Go Run Terminal Help newproject
src > components > shop > ShopCard > currentProducts.map() callback

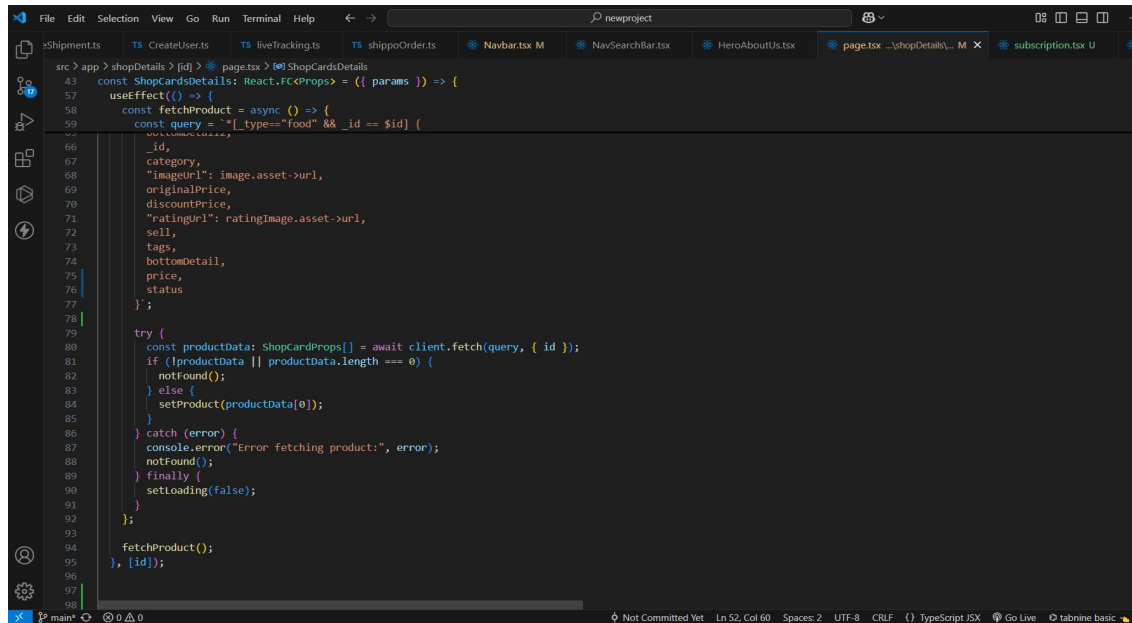
28
29 const fetchProducts = async (category: string | null): Promise<ShopCardProps[]> => {
30   const query = category
31   ? `[_type="food" && category == "${category}"] {
32     ratingNum,
33     description,
34     reviews,
35     name,
36     text,
37     bottomDetail,
38     _id,
39     category,
40     "imageUrl": image.asset->url,
41     originalPrice,
42     discountPrice,
43     "ratingUrl": ratingImage.asset->url,
44     sell,
45     bottomDetail,
46     price
47   }`
48   : `[_type="food"] {
49     ratingNum,
50     description,
51     reviews,
52     name,
53     text,
54     bottomDetail,
55     _id,
56     category,
57     "imageUrl": image.asset->url,
58     originalPrice,
59     discountPrice,
60     "ratingUrl": ratingImage.asset->url,
61     sell,
62     bottomDetail,
63     price
64   }`;
65 }
```



This is my frontend output these all products I've fetched from sanity and displayed on the frontend.

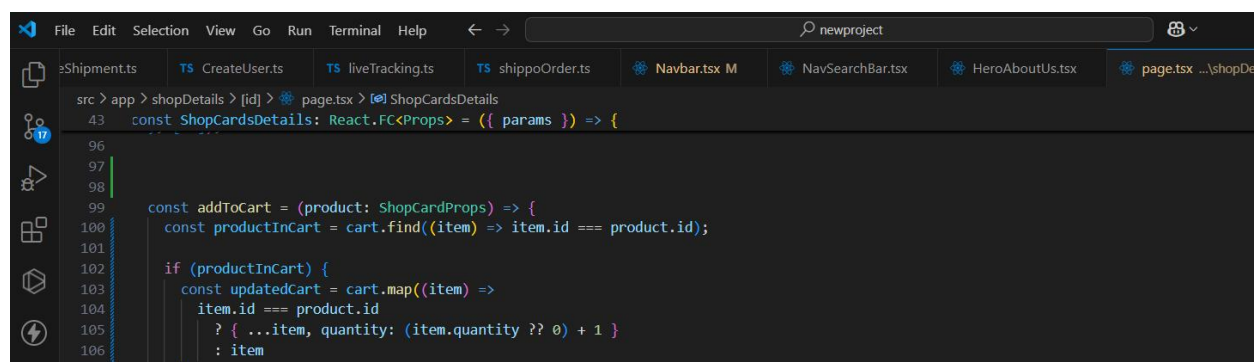
2. Product Detailed Component:

After fetching multiple products from sanity CMS, now I'm going to create single product page by using dynamic routing where user can visit further detail of product like product description, size, color, reviews etc...



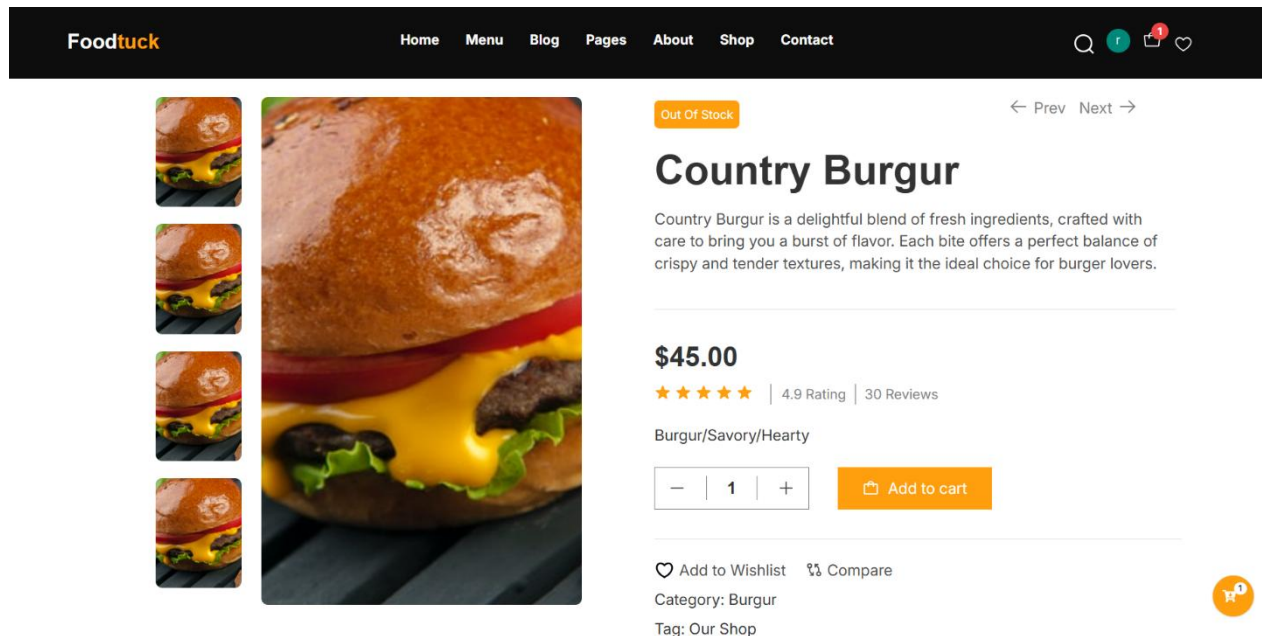
```
src > app > shopDetails > [id] > page.tsx > ShopCardsDetails
43 const ShopCardsDetails: React.FC<Props> = ({ params }) => {
57   useEffect(() => {
58     const fetchProduct = async () => {
59       const query = `*[_type=="food" && _id == $id] {
60         _id,
61         category,
62         "imageUrl": image.asset->url,
63         originalPrice,
64         discountPrice,
65         "ratingUrl": ratingImage.asset->url,
66         sell,
67         tags,
68         bottomDetail,
69         price,
70         status
71       }`;
72       try {
73         const productData: ShopCardProps[] = await client.fetch(query, { id });
74         if (!productData || productData.length === 0) {
75           notFound();
76         } else {
77           setProduct(productData[0]);
78         }
79       } catch (error) {
80         console.error("Error fetching product:", error);
81         notFound();
82       } finally {
83         setLoading(false);
84       }
85     };
86     fetchProduct();
87   }, [id]);
88 }
```

So firstly in my card component I used link component for navigation I've passed **shopDetails** on the basis of **id**.



```
src > app > shopDetails > [id] > page.tsx > ShopCardsDetails
96
97
98
99 const addToCart = (product: ShopCardProps) => {
100   const productInCart = cart.find((item) => item.id === product.id);
101
102   if (productInCart) {
103     const updatedCart = cart.map((item) =>
104       item.id === product.id
105       ? { ...item, quantity: (item.quantity ?? 0) + 1 }
106       : item
107     );
108   }
109 }
```

Here is my dynamic routing route where user can visit detailed for product I've used id for the dynamic routing on the basis of id I've used dynamic routing.



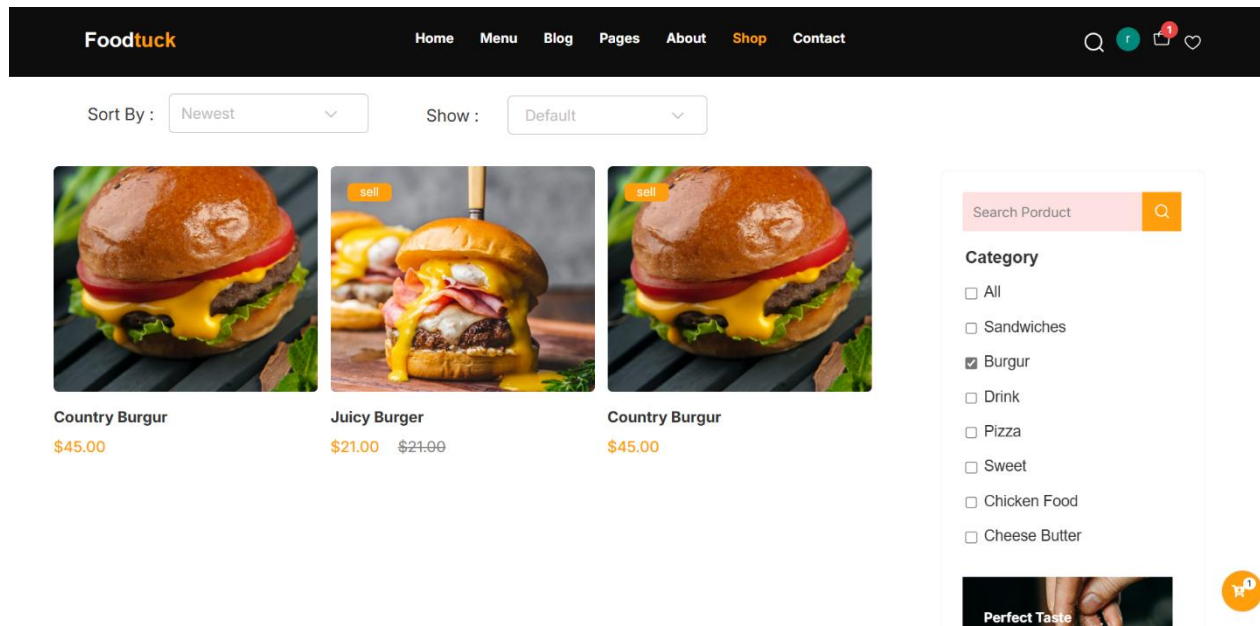
And here I've used try and catch for error handling if user buy product which is available in my stock then my try method will implemented otherwise the error with through because of unavailability of product.

So when user click on the card, user will navigate the single product details page on the basis of unique id. So now in step 2 I've implemented my dynamic routing successfully.

3. Category Component:

After dynamic routing I've to implement **Category** component. Firstly I've to used sanity groq query because the data is stored in sanity then I've to fetch category by using groq query. For this I've use State for selected category and then I've used groq query

Here is my code snippet you can see how functionality work.



```

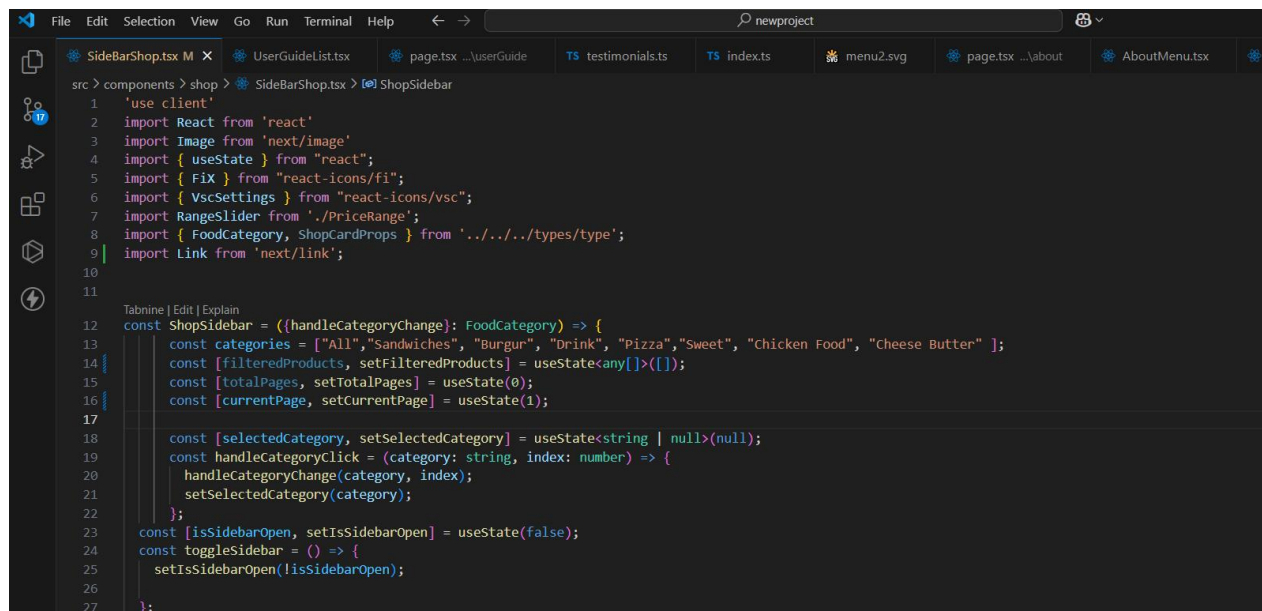
File Edit Selection View Go Run Terminal Help newproject
SideBarShop.tsx M X UserGuideList.tsx page.tsx .../UserGuide TS testimonials.ts TS index.ts menu2.svg page.tsx .../about AboutMenu.tsx Hero.tsx

src > components > shop > SideBarShop.tsx > ShopSidebar
1 'use client'
2 import React from 'react'
3 import Image from 'next/image'
4 import { useState } from 'react';
5 import { fix } from 'react-icons/fi';
6 import { VscSettings } from 'react-icons/vsc';
7 import RangeSlider from './PriceRange';
8 import { FoodCategory, ShopCardProps } from '../../types/type';
9 import Link from 'next/link';
10
11
12 Tabnine | Edit | Explain
13 const ShopSidebar = ({handleCategoryChange}: FoodCategory) => {
14   const categories = ["All", "Sandwiches", "Burgur", "Drink", "Pizza", "Sweet", "Chicken Food", "Cheese Butter"];
15   const [filteredProducts, setFilteredProducts] = useState<any[]>([]);
16   const [totalPages, setTotalPages] = useState(0);
17   const [currentPage, setCurrentPage] = useState(1);
18
19   const [selectedCategory, setSelectedCategory] = useState<string | null>(null);
20   const handleCategoryClick = (category: string, index: number) => {
21     handleCategoryChange(category, index);
22     setSelectedCategory(category);
23   };
24   const [isSidebarOpen, setIsSidebarOpen] = useState(false);
25   const toggleSidebar = () => {
26     setIsSidebarOpen(!isSidebarOpen);
27   };
28   return (
29     <>
30     /* For desktop screens */
31     <aside className='md:flex hidden sidebar'>
32       <div className='lg:w-[312px] h-[1590px] w-[290px] relative top-[150px] xxxl:left-[1300px] xl:left-[1050px] lg:left-[700px] xlg:left-[850px] md:left-[450px]'>
33         <div className='w-[248px] h-[46px] p-3 relative lg:left-[25px] left-[10px] top-[1.5rem] bg-red-100 '>
34           <input
35             type="text"
36             placeholder="Search Porduct"

```

For reusability I created **SideBar** component and then I've passed props.

This is my sidebar component where I used props an array for categories.



```
1 'use client'
2 import React from 'react'
3 import Image from 'next/image'
4 import { useState } from "react";
5 import { FiX } from "react-icons/fi";
6 import { VscSettings } from "react-icons/vsc";
7 import RangeSlider from './PriceRange';
8 import { FoodCategory, ShopCardProps } from '../types/type';
9 import Link from 'next/link';
10
11
12 Tabnine | Edit | Explain
13 const ShopSidebar = ({handleCategoryChange}: FoodCategory) => {
14   const categories = ["All", "Sandwiches", "Burgur", "Drink", "Pizza", "Sweet", "Chicken Food", "Cheese Butter" ];
15   const [filteredProducts, setFilteredProducts] = useState<any[]>([]);
16   const [totalPages, setTotalPages] = useState(0);
17   const [currentPage, setCurrentPage] = useState(1);
18
19   const [selectedCategory, setSelectedCategory] = useState<string | null>(null);
20   const handleClick = (category: string, index: number) => {
21     handleCategoryChange(category, index);
22     setSelectedCategory(category);
23   };
24   const [isSidebarOpen, setIsSidebarOpen] = useState(false);
25   const toggleSidebar = () => {
26     setIsSidebarOpen(!isSidebarOpen);
27   };
28 }
```

And then using map function I've displayed my all categories dynamically.
So here is my frontend where user can easily navigate through any category .

So I've completed my step 3 successfully. In this step I've created dynamic category component.