

# MUHAMMAD RAMZAN

## (00367450)

## Day 4 Implementation Report

Fetch Products on shop page and in different components

By sanity



Trenton modular sofa\_3

**24,998.00**



Granite dining table with dining chair

**25,000.00**



Outdoor bar table and stool

**24,999.00**



Plain console with teak mirror

**25,000.00**



Grain coffee table

**15,000.00**



### Top Picks For You

Find a bright ideal to suit your taste with our great selection of suspension, floor and table lights.



Outdoor bar table and stool

**24,999.00**



Reclaimed teak Sideboard

**19,999.00**



Granite square side table

**258,798.00**

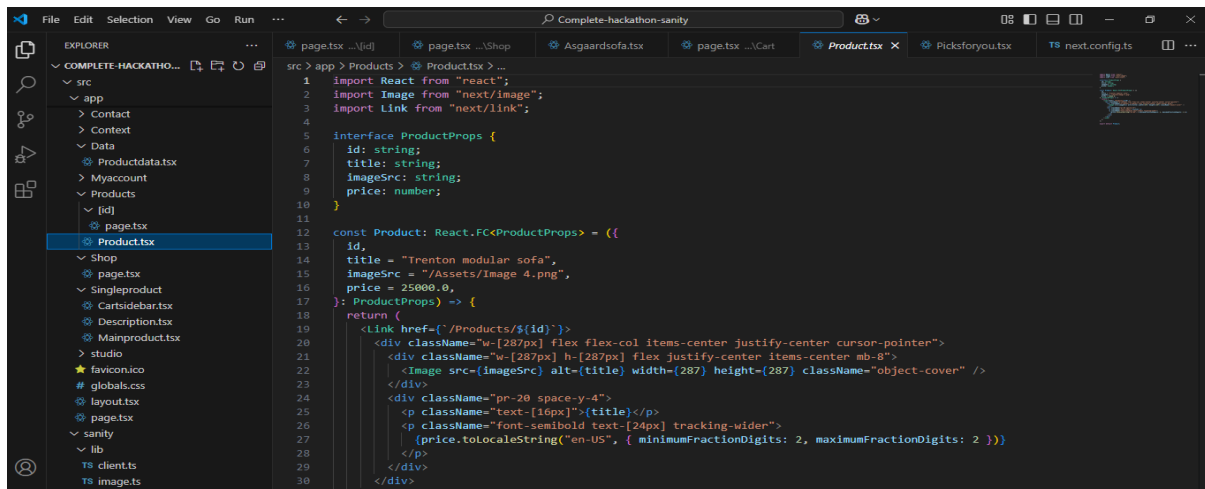


Cosmo Sofa

**30,000.00**

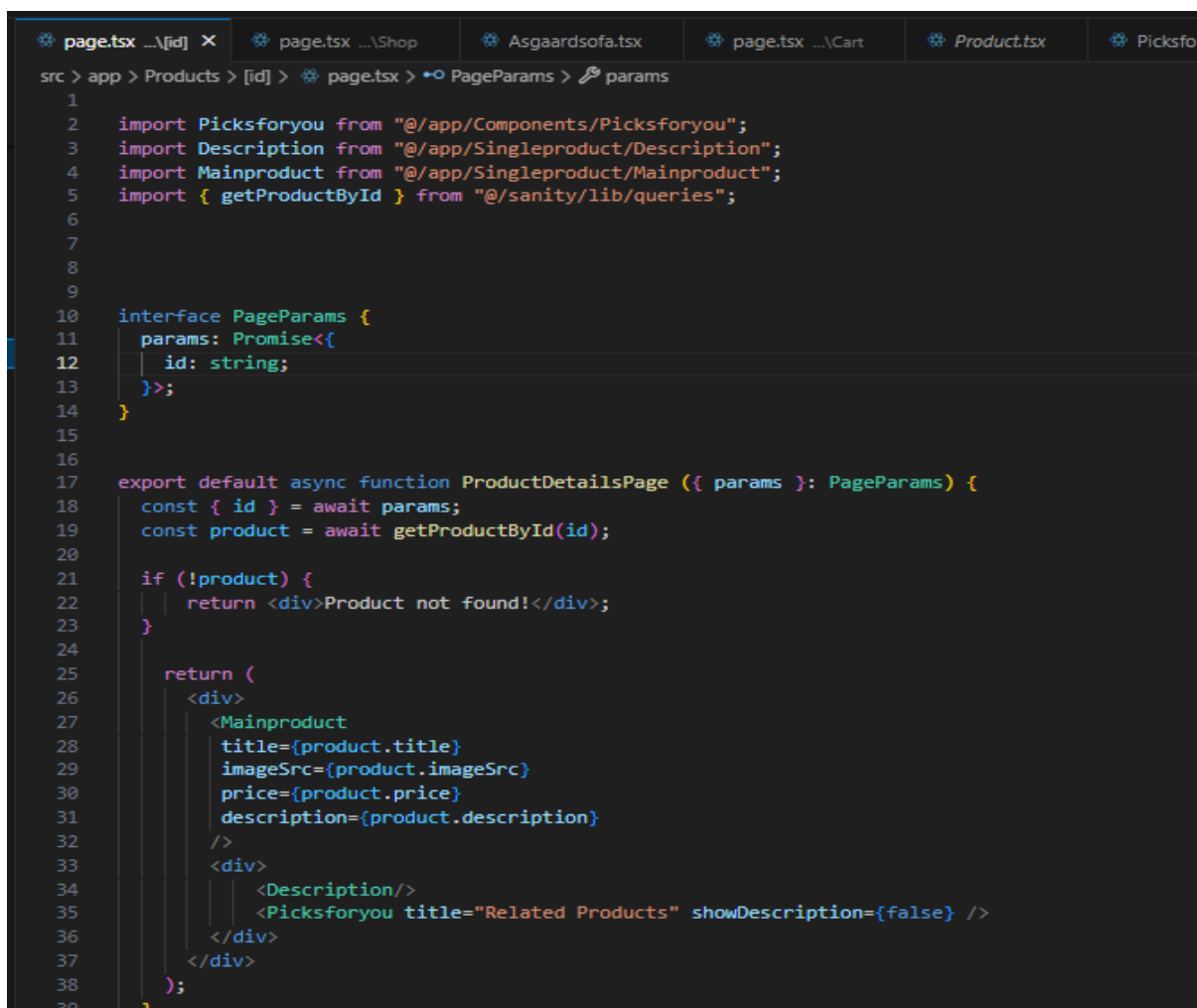
[View more](#)

this my product code that is dynamically rendering in my complete project with the help of dynamic route



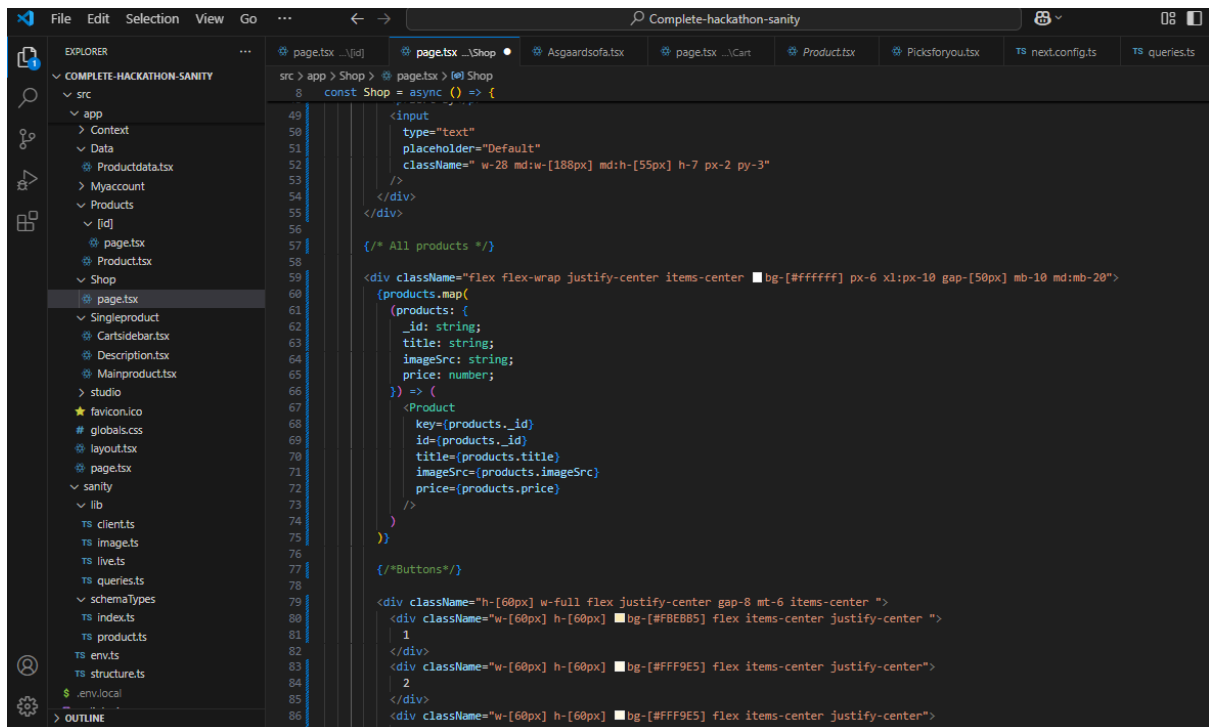
```
1 import React from "react";
2 import Image from "next/image";
3 import Link from "next/link";
4
5 interface ProductProps {
6   id: string;
7   title: string;
8   imageSrc: string;
9   price: number;
10 }
11
12 const Product: React.FC<ProductProps> = ({
13   id,
14   title = "Trenton modular sofa",
15   imageSrc = "/Assets/Image 4.png",
16   price = 25000.0,
17 }: ProductProps) => {
18   return (
19     <Link href={`/${Products}/${id}`}>
20       <div className="w-[287px] flex flex-col items-center justify-center cursor-pointer">
21         <div className="w-[287px] h-[287px] flex justify-center items-center mb-8">
22           <Image src={imageSrc} alt={title} width={287} height={287} className="object-cover" />
23         </div>
24         <div className="pr-20 space-y-4">
25           <p className="text-[16px]">{title}</p>
26           <p className="font-semibold text-[24px] tracking-wider">
27             {price.toLocaleString("en-US", { minimumFractionDigits: 2, maximumFractionDigits: 2 })}
28           </p>
29         </div>
30       </div>
31     </Link>
32   );
33 }
```

This is the dynamic route page. With this route, all my products are displayed dynamically on the shop page and in various components.



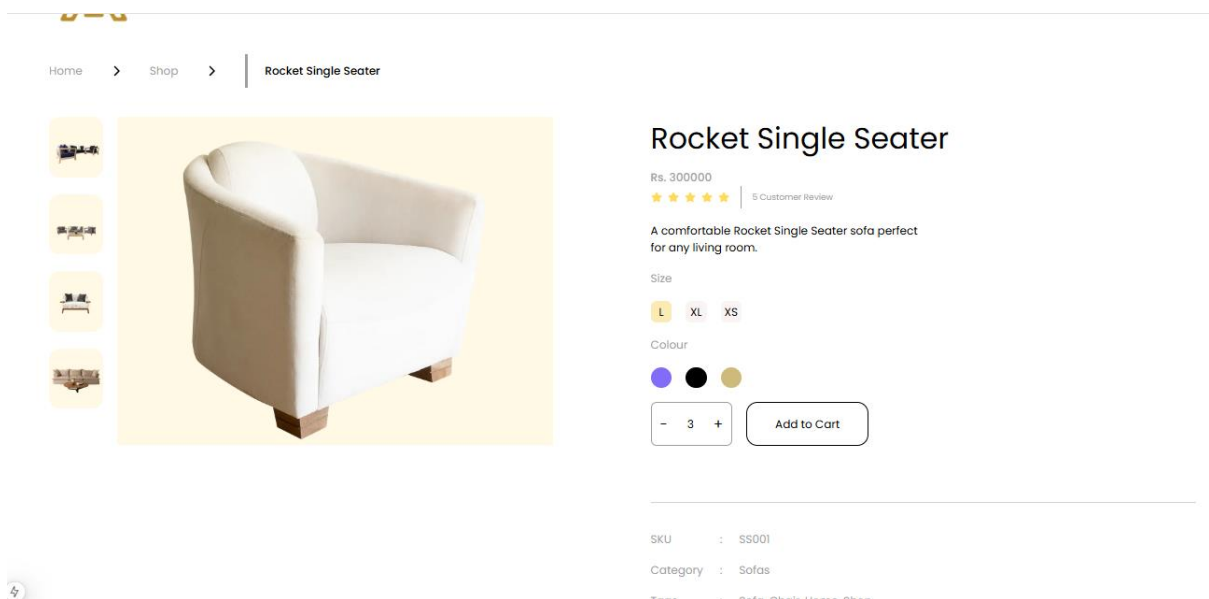
```
1
2 import Picksforyou from "@app/Components/Picksforyou";
3 import Description from "@app/Singleproduct/Description";
4 import Mainproduct from "@app/Singleproduct/Mainproduct";
5 import { getProductById } from "@sanity/lib/queries";
6
7
8
9
10 interface PageParams {
11   params: Promise<{
12     id: string;
13   }>;
14 }
15
16
17 export default async function ProductDetailsPage ({ params }: PageParams) {
18   const { id } = await params;
19   const product = await getProductById(id);
20
21   if (!product) {
22     return <div>Product not found!</div>;
23   }
24
25   return (
26     <div>
27       <Mainproduct
28         title={product.title}
29         imageSrc={product.imageSrc}
30         price={product.price}
31         description={product.description}
32       />
33       <div>
34         <Description/>
35         <Picksforyou title="Related Products" showDescription={false} />
36       </div>
37     </div>
38   );
39 }
```

This is my shop page there my all products are rendering from sanity with the help of api



```
src > app > Shop > page.tsx > Shop
const Shop = async () => {
  49
  50   <input
  51     type="text"
  52     placeholder="Default"
  53     className="w-28 md:w-[188px] md:h-[55px] h-7 px-2 py-3"
  54   />
  55 </div>
  56 </div>
  57
  58 /* All products */
  59
  60 <div className="flex flex-wrap justify-center items-center bg-[#ffffff] px-6 xl:px-10 gap-[50px] mb-10 md:mb-20">
  61   {products.map(
  62     (products: {
  63       _id: string;
  64       title: string;
  65       imageSrc: string;
  66       price: number;
  67     }) => (
  68       <Product
  69         key={products._id}
  70         id={products._id}
  71         title={products.title}
  72         imageSrc={products.imageSrc}
  73         price={products.price}
  74       />
  75     )
  76   )}
  77
  78 /*Buttons*/
  79
  80 <div className="h-[60px] w-full flex justify-center gap-8 mt-6 items-center">
  81   <div className="w-[60px] h-[60px] bg-[#FBE885] flex items-center justify-center">
  82     1
  83   </div>
  84   <div className="w-[60px] h-[60px] bg-[#FFF9E5] flex items-center justify-center">
  85     2
  86   </div>
  87   <div className="w-[60px] h-[60px] bg-[#FFF9E5] flex items-center justify-center">
```

On this Mainproduct page i add a functionality “add to cart” all products will render on this page by the product id and then i can add to cart



My main product page code there i implement add to cart functionality

```
5 Customer Review
</div>
</div>
<p className="w-[424px] pr-5 my-5">{description}</p>
<p className="text-[#9F9F9F] mt-3">Size</p>

<div className="flex gap-5 my-5">
  <div className="w-[30px] h-[30px] bg-[#FBE8B5] rounded-lg flex justify-center items-center ">
    L
  </div>
  <div className="w-[30px] h-[30px] bg-[#FAF4F4] rounded-lg flex justify-center items-center ">
    XL
  </div>
  <div className="w-[30px] h-[30px] bg-[#FAF4F4] rounded-lg flex justify-center items-center ">
    XS
  </div>
</div>
<p className="text-[#9F9F9F] mt-3">Colour</p>

<div className="flex gap-5 my-5">
  <div className="w-[30px] h-[30px] bg-[#816DFA] rounded-full"></div>
  <div className="w-[30px] h-[30px] bg-black rounded-full"></div>
  <div className="w-[30px] h-[30px] bg-[#CDBA7B] rounded-full"></div>
</div>

<div className="flex gap-5">
  <div className="flex justify-center items-center border-2 border-[#9F9F9F] rounded-lg py-4 px-3 gap-7 ">
    <button onClick={handleDecrement} className="text-xl">
      -
    </button>

    { /* Display the dynamic quantity */ }
    <span>{localQuantity}</span>
    <button onClick={handleIncrement} className="text-xl">
      +
    </button>
  </div>
  <button
    onClick={handleAddToCart}
    className="text-black py-4 px-10 flex justify-center items-center border-black rounded-2xl border-2"
  >
    Add to Cart
  </button>
</div>

<hr className="mt-20 w-full border-t-2 border-gray-300" />
```



And this is my cart context page for add to cart functionality

```

12
13 interface CartContextType {
14   cart: CartItem[];
15   addToCart: (item: CartItem) => void;
16   removeFromCart: (id: string) => void;
17   updateCartQuantity: (id: string, quantity: number) => void;
18 }
19
20
21 const CartContext = createContext<CartContextType | undefined>(undefined);
22
23 export const CartProvider = ({ children }: { children: ReactNode }) => {
24   const [cart, setCart] = useState<CartItem[]>([]);
25
26   const addToCart = (product: CartItem) => {
27     setCart((prevCart) => {
28       const existingProduct = prevCart.find((item) => item.id === product.id);
29
30       if (existingProduct) {
31         // If the product exists, update its quantity
32         return prevCart.map((item) =>
33           item.id === product.id
34             ? { ...item, quantity: item.quantity + product.quantity }
35             : item
36         );
37       } else {
38         // If the product doesn't exist, add it to the cart
39         return [...prevCart, product];
40       }
41     });
42   }
43
44   const removeFromCart = (id: string) => {
45     setCart((prevCart) => prevCart.filter((item) => item.id !== id));
46   };
47
48   const updateCartQuantity = (id: string, quantity: number) => {
49     setCart((prevCart) =>
50       prevCart.map((item) =>
51         item.id === id ? { ...item, quantity } : item
52       )
53     );
54   };
55 }

```

This is my cart page there product will show if anyone click on add to cart button i add a complete functionality you can remove product add quantity, increase and decrease quantity on cart page.

Product					
	Product	Price	Quantity	Subtotal	
	Rocket Single Seater	Rs. 300,000	<input type="text" value="3"/>	Rs. 900,000	Remove
	Outdoor bar table and stool	Rs. 24,999	<input type="text" value="2"/>	Rs. 49,998	Remove

**Cart Totals**

Subtotal Rs. 949,998

Total **Rs. 949,998**

Check Out

On **Day 4**, I implemented the functionality to fetch products dynamically on the shop page using Sanity's API. I used dynamic routes to display the products across different components. On the **Mainproduct page**, I added an "add to cart" feature, allowing users to add products to their cart. In the **Cart Context**, I set up the necessary functionality for adding and removing items, as well as adjusting their quantities. The **Cart page** shows the added products, with options to remove or modify the quantity of each item. Below, I've provided the relevant code and images.