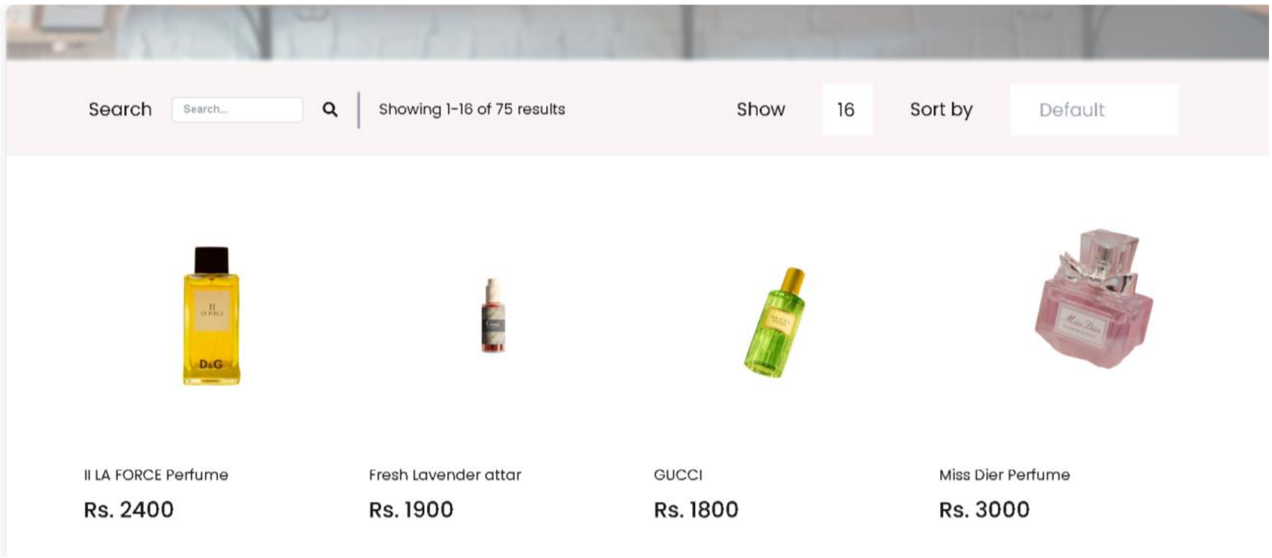
Day 4 - Dynamic Frontend Components- Aromas

Functional Deliverables

Screenshots

Product Listing



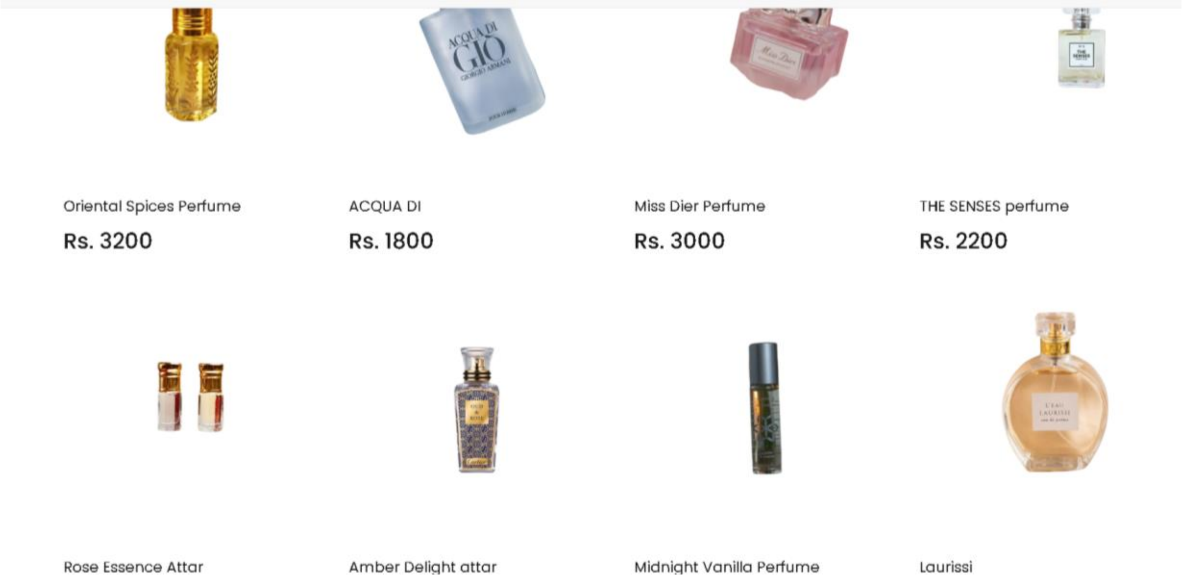
Search Search... Showing 1-16 of 75 results Show 16 Sort by Default

II LA FORCE Perfume
Rs. 2400

Fresh Lavender attar
Rs. 1900

GUCCI
Rs. 1800

Miss Dier Perfume
Rs. 3000



Oriental Spices Perfume
Rs. 3200

ACQUA DI
Rs. 1800

Miss Dier Perfume
Rs. 3000

THE SENSES perfume
Rs. 2200

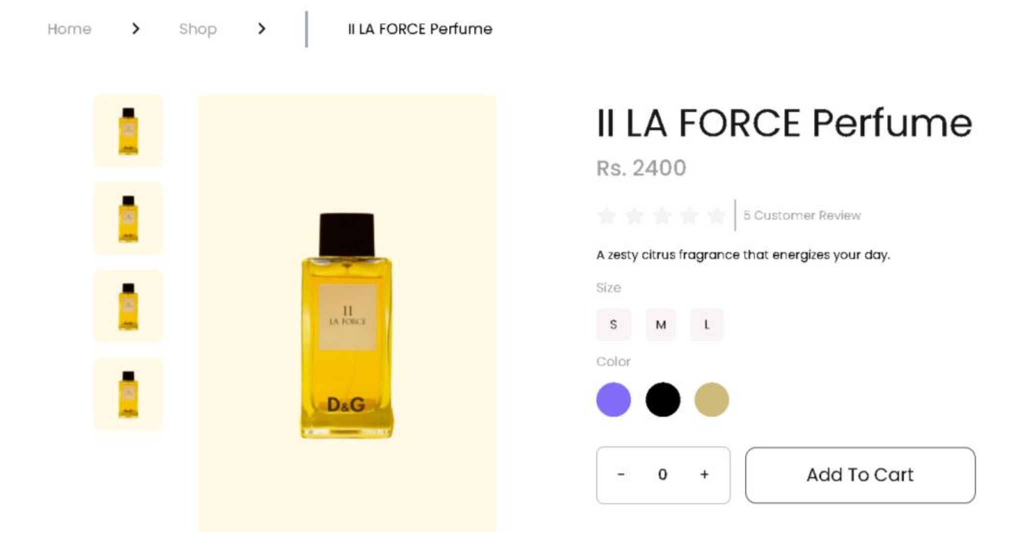
Rose Essence Attar

Amber Delight attar

Midnight Vanilla Perfume

Laurissi

Product detail dynamic



Home > Shop > | II LA FORCE Perfume

II LA FORCE Perfume

Rs. 2400

☆☆☆☆☆ | 5 Customer Review

A zesty citrus fragrance that energizes your day.

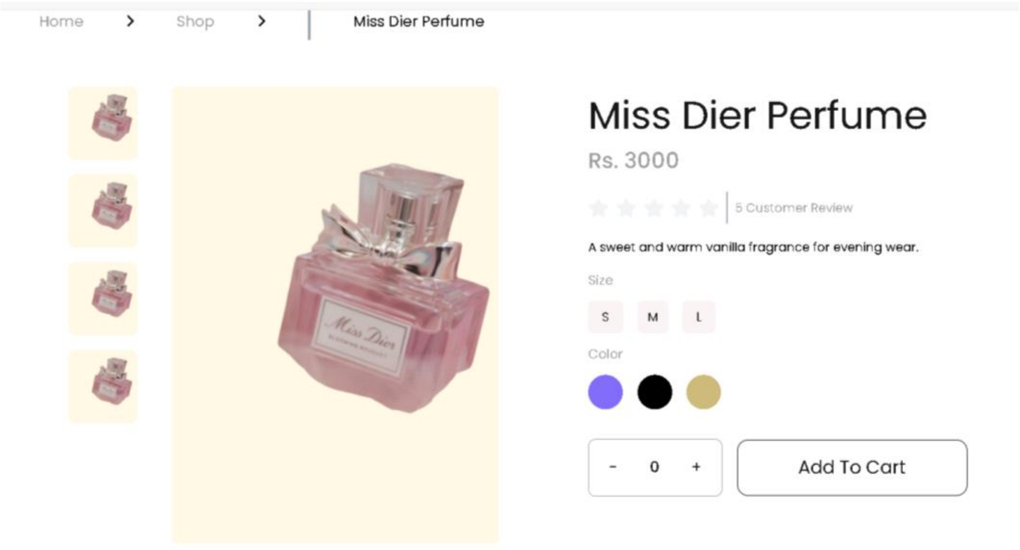
Size

S M L

Color

● ● ●

- 0 +   Add To Cart



Home > Shop > | Miss Dier Perfume

Miss Dier Perfume

Rs. 3000

☆☆☆☆☆ | 5 Customer Review

A sweet and warm vanilla fragrance for evening wear.

Size

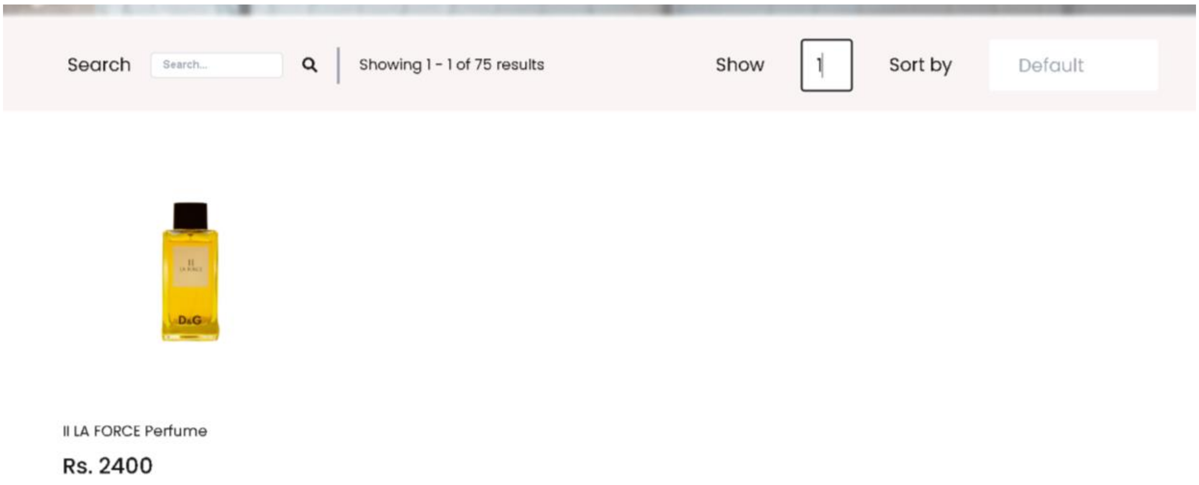
S M L

Color

● ● ●

- 0 +   Add To Cart

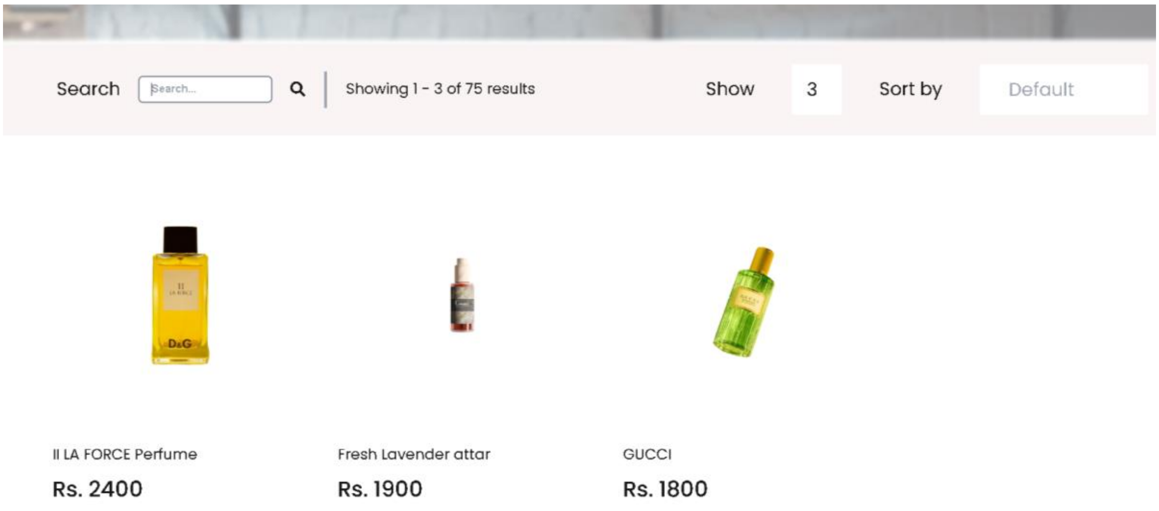
Show products on page > e.g. 1



Search Search... | Showing 1 - 1 of 75 results   Show 1   Sort by Default

II LA FORCE Perfume

Rs. 2400



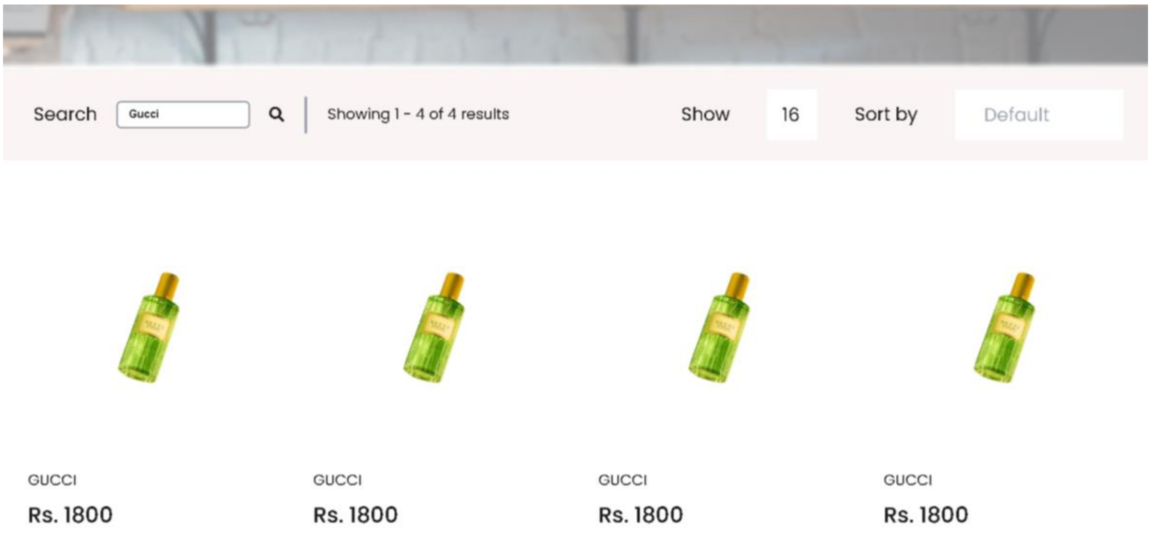
Search Search... Showing 1 - 3 of 75 results Show 3 Sort by Default

II LA FORCE Perfume
Rs. 2400

Fresh Lavender attar
Rs. 1900

GUCCI
Rs. 1800

Search bar result



Search Gucci Showing 1 - 4 of 4 results Show 16 Sort by Default

GUCCI
Rs. 1800

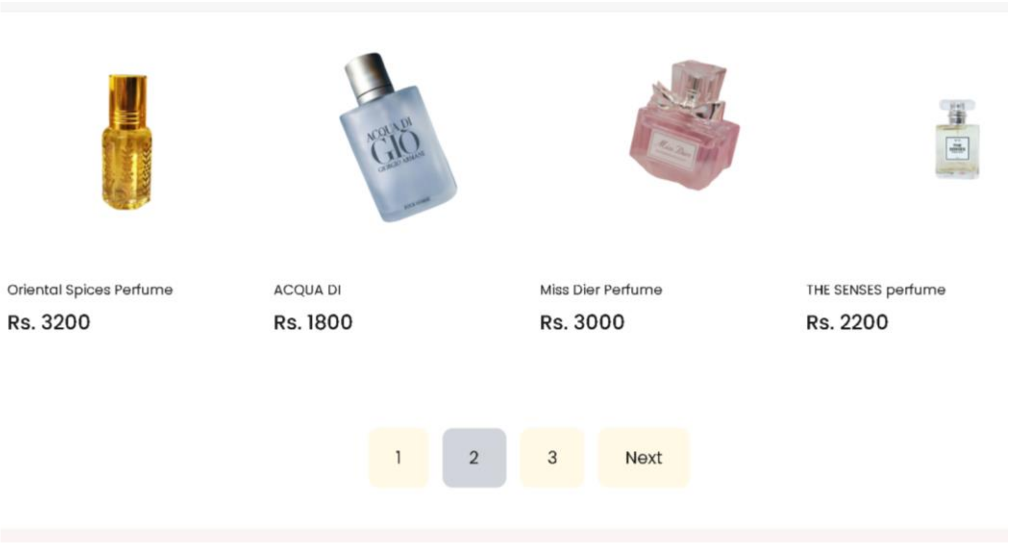
GUCCI
Rs. 1800

GUCCI
Rs. 1800

GUCCI
Rs. 1800

Pagination

Second page



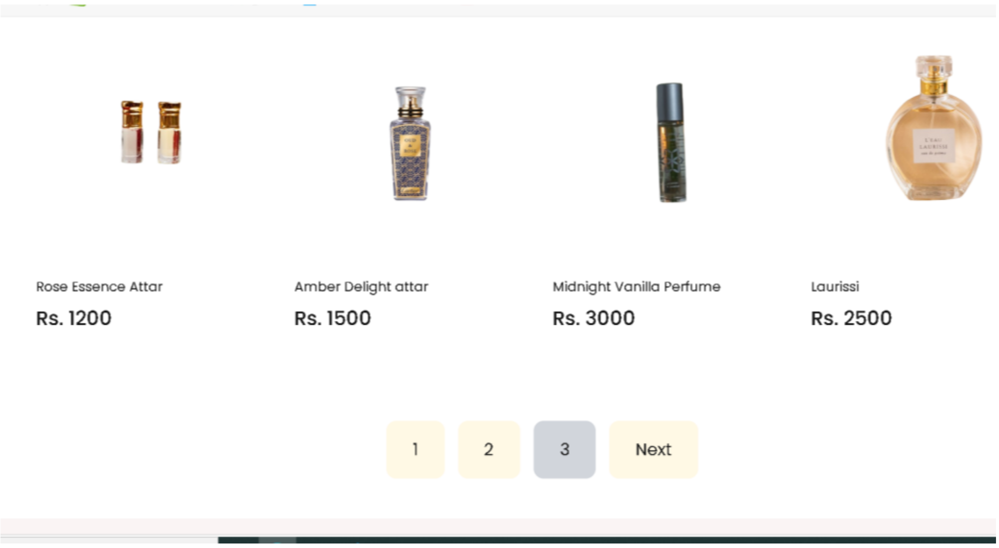
Oriental Spices Perfume
Rs. 3200

ACQUA DI
Rs. 1800

Miss Dier Perfume
Rs. 3000

THE SENSES perfume
Rs. 2200

Third page



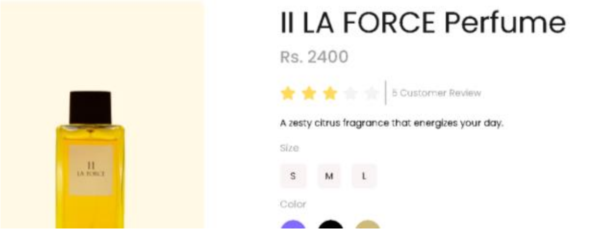
Rose Essence Attar
Rs. 1200

Amber Delight attar
Rs. 1500

Midnight Vanilla Perfume
Rs. 3000

Laurissi
Rs. 2500

Rating of product



II LA FORCE Perfume

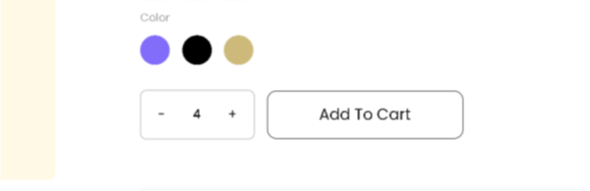
Rs. 2400

★★★☆☆ | 5 Customer Review

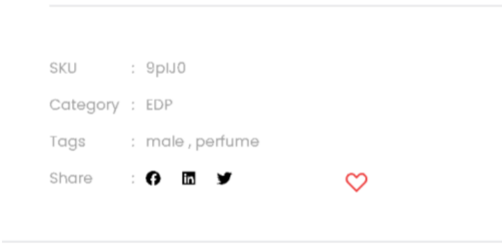
A zesty citrus fragrance that energizes your day.

Size

Increasing product quantity



Like Product

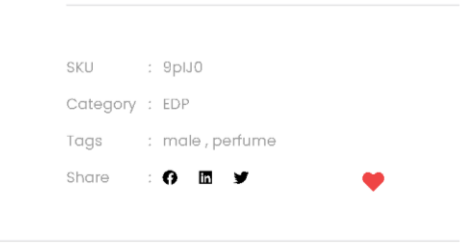


SKU : 9plJ0

Category : EDP

Tags : male , perfume

Share :



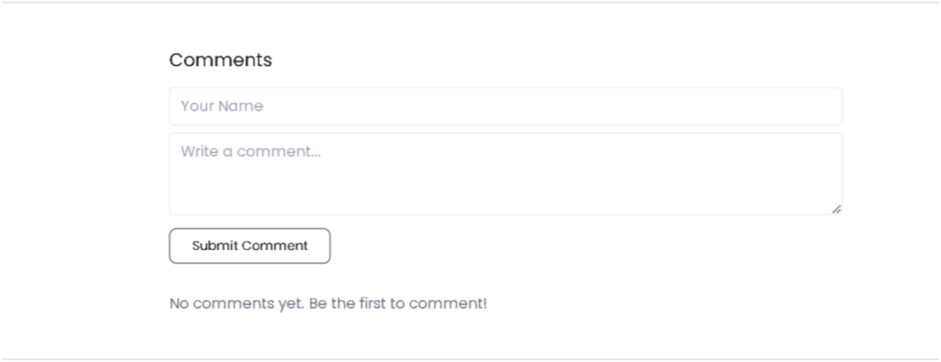
SKU : 9plJ0

Category : EDP

Tags : male , perfume

Share :

Comments

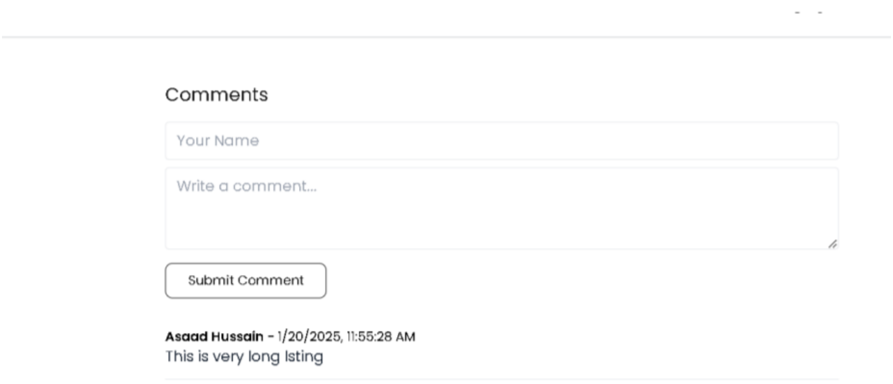


Comments

Your Name

Write a comment...

No comments yet. Be the first to comment!



Comments

Your Name

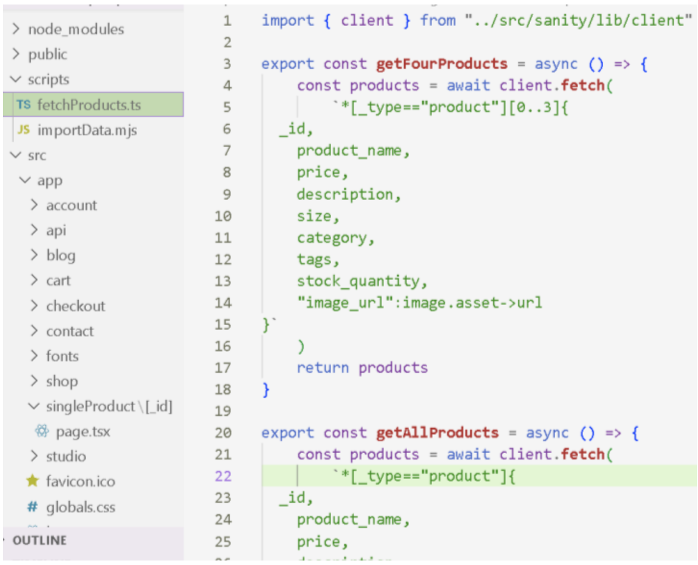
Write a comment...

Asaad Hussain - 1/20/2025, 11:55:28 AM
This is very long lsting

Code Deliverables

Snippets and Documentation

Created a asynchronous function to fetch the data from Sanity using GROQ

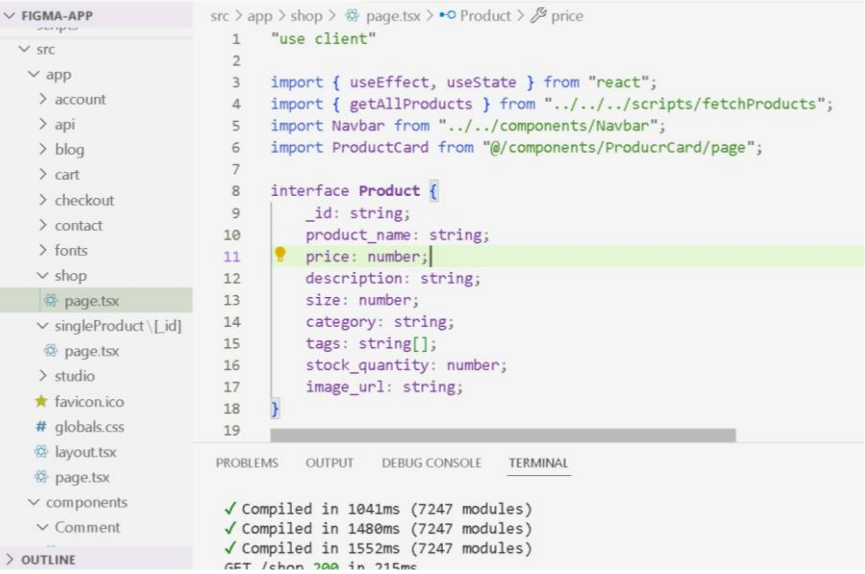


```
import { client } from "../src/sanity/lib/client"

export const getFourProducts = async () => {
    const products = await client.fetch(
        `*[_type=="product"][0..3]{
  _id,
    product_name,
    price,
    description,
    size,
    category,
    tags,
    stock_quantity,
    "image_url":image.asset->url
}`
    )
    return products
}

export const getAllProducts = async () => {
    const products = await client.fetch(
        `*[_type=="product"]{
  _id,
    product_name,
    price,
```

To get the products data in the shop page for listing, ensured type check, GET the data from the getProducts function in useEffect and saved it in the state created, also implemented the loading.

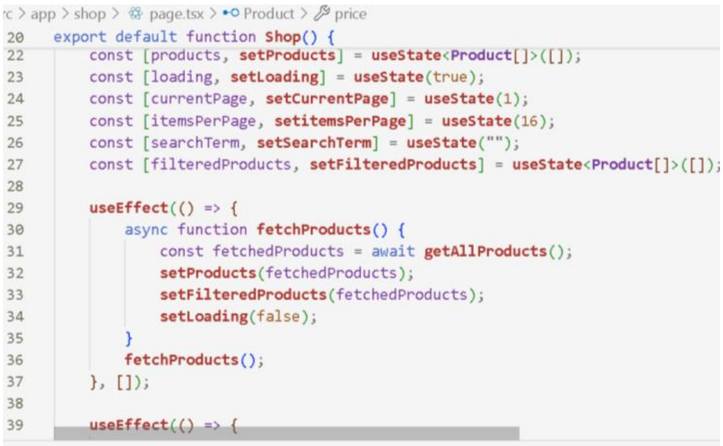
Other states are created to implement filtering on search, show items per page and pagination

Product Listing

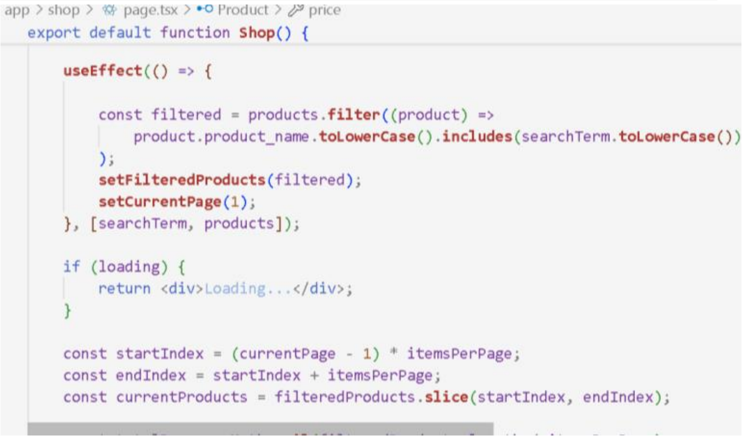


```
"use client"

import { useEffect, useState } from "react";
import { getAllProducts } from "../../../scripts/fetchProducts";
import Navbar from "../../components/Navbar";
import ProductCard from "@/components/ProducrCard/page";

interface Product {
    _id: string;
    product_name: string;
    price: number;
    description: string;
    size: number;
    category: string;
    tags: string[];
    stock_quantity: number;
    image_url: string;
}

```

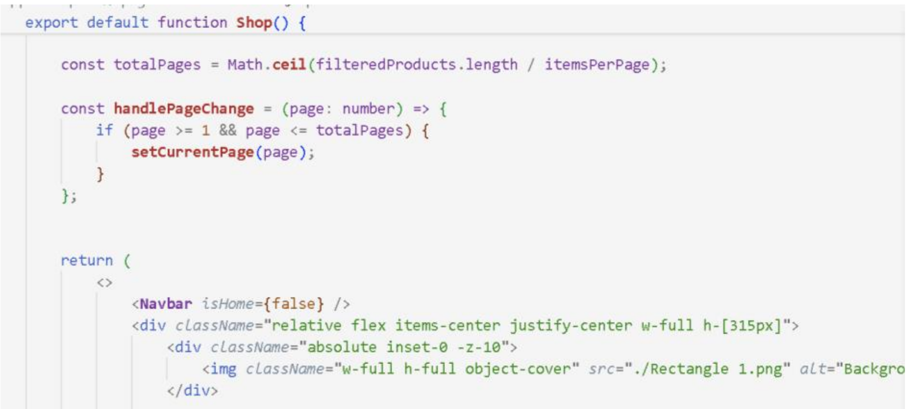
```
✓ Compiled in 1041ms (7247 modules)
✓ Compiled in 1480ms (7247 modules)
✓ Compiled in 1552ms (7247 modules)
GET /shop 200 in 215ms
```



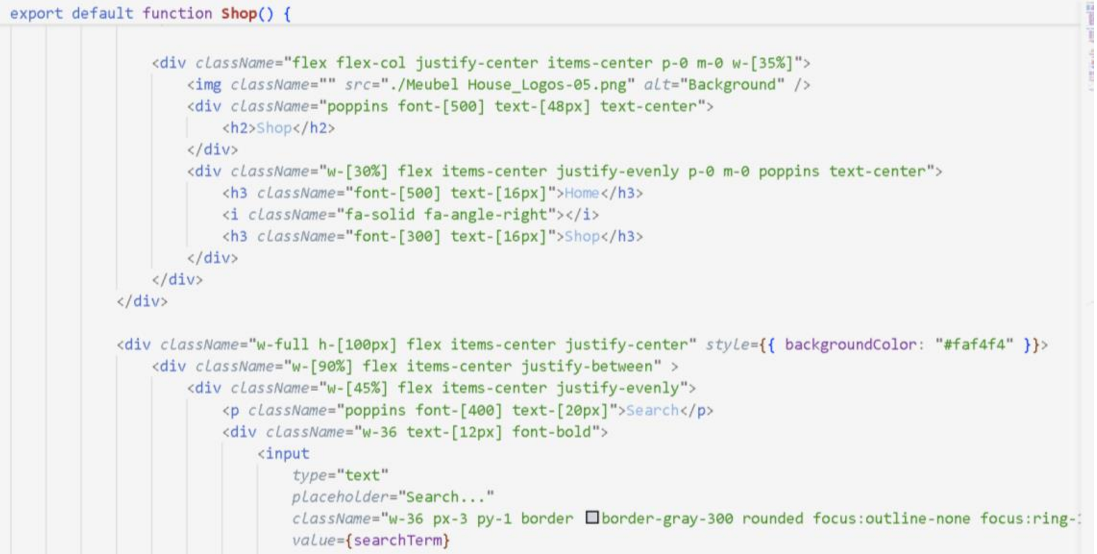
```
export default function Shop() {
    const [products, setProducts] = useState<Product[]>([]);
    const [loading, setLoading] = useState(true);
    const [currentPage, setCurrentPage] = useState(1);
    const [itemsPerPage, setitemsPerPage] = useState(16);
    const [searchTerm, setSearchTerm] = useState("");
    const [filteredProducts, setFilteredProducts] = useState<Product[]>([]);

    useEffect(() => {
        async function fetchProducts() {
            const fetchedProducts = await getAllProducts();
            setProducts(fetchedProducts);
            setFilteredProducts(fetchedProducts);
            setLoading(false);
        }
        fetchProducts();
    }, []);

    useEffect(() => {
```



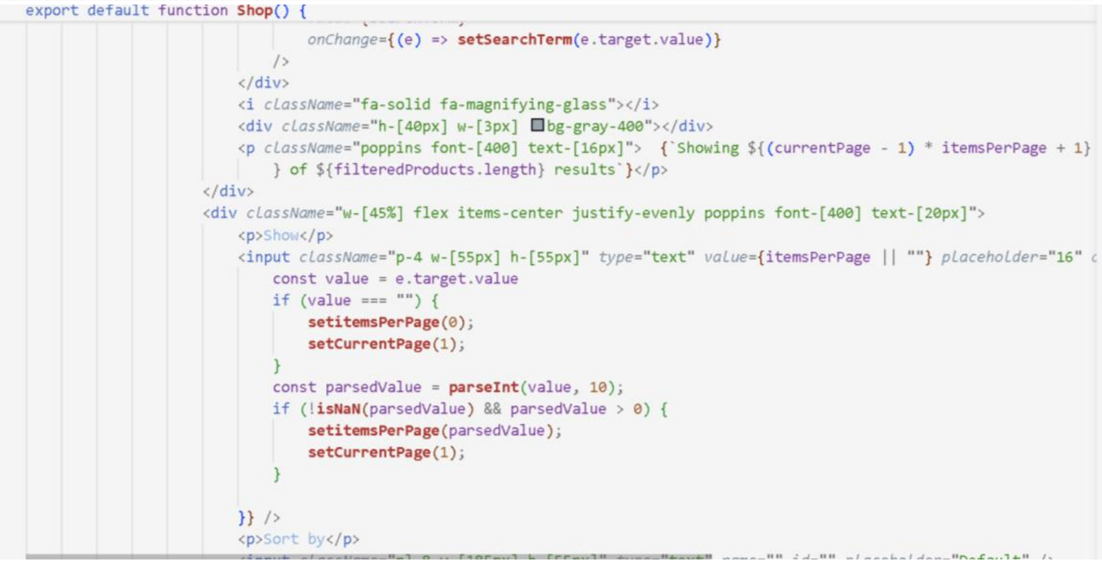
```
export default function Shop() {

    useEffect(() => {

        const filtered = products.filter((product) =>
            product.product_name.toLowerCase().includes(searchTerm.toLowerCase())
        );
        setFilteredProducts(filtered);
        setCurrentPage(1);
    }, [searchTerm, products]);

    if (loading) {
        return <div>Loading...</div>;
    }

    const startIndex = (currentPage - 1) * itemsPerPage;
    const endIndex = startIndex + itemsPerPage;
    const currentProducts = filteredProducts.slice(startIndex, endIndex);
```



```
export default function Shop() {

    const totalPages = Math.ceil(filteredProducts.length / itemsPerPage);

    const handlePageChange = (page: number) => {
        if (page >= 1 && page <= totalPages) {
            setCurrentPage(page);
        }
    };


    return (
        <>
            <Navbar isHome={false} />
            <div className="relative flex items-center justify-center w-full h-[315px]">
                <div className="absolute inset-0 -z-10">
                    <img className="w-full h-full object-cover" src="./Rectangle 1.png" alt="Backgro
                </div>
```

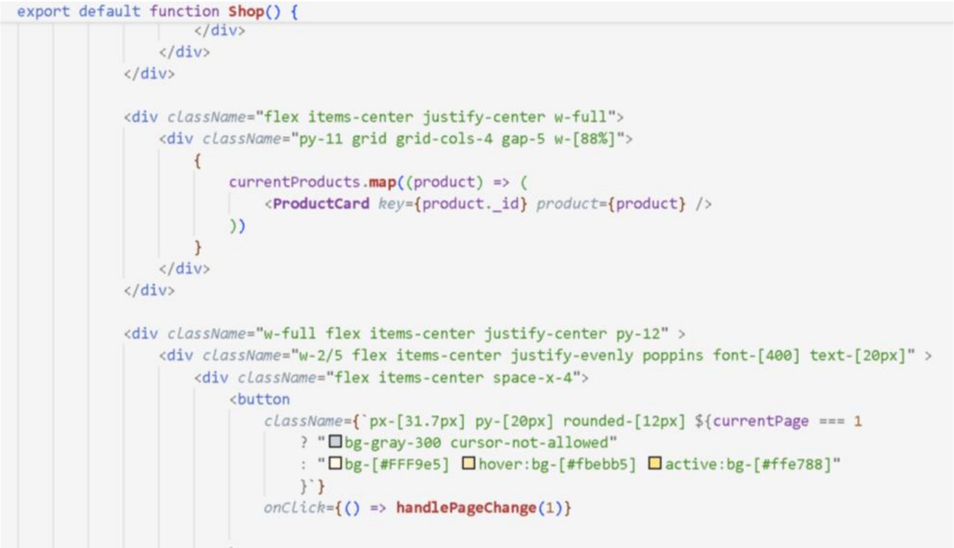


```
export default function Shop() {

                <div className="flex flex-col justify-center items-center p-0 m-0 w-[35%]">
                    <img className="" src="./Meubel House_Logos-05.png" alt="Background" />
                    <div className="poppins font-[500] text-[48px] text-center">
                        <h2>Shop</h2>
                    </div>
                    <div className="w-[30%] flex items-center justify-evenly p-0 m-0 poppins text-center">
                        <h3 className="font-[500] text-[16px]">Home</h3>
                        <i className="fa-solid fa-angle-right"></i>
                        <h3 className="font-[300] text-[16px]">Shop</h3>
                    </div>
                </div>
            </div>

            <div className="w-full h-[100px] flex items-center justify-center" style={{ backgroundColor: "#faf4f4" }}>
                <div className="w-[90%] flex items-center justify-between" >
                    <div className="w-[45%] flex items-center justify-evenly">
                        <p className="poppins font-[400] text-[20px]">Search</p>
                        <div className="w-36 text-[12px] font-bold">
                            <input
                                type="text"
                                placeholder="Search..."
                                className="w-36 px-3 py-1 border □border-gray-300 rounded focus:outline-none focus:ring-
                                value={searchTerm}
```

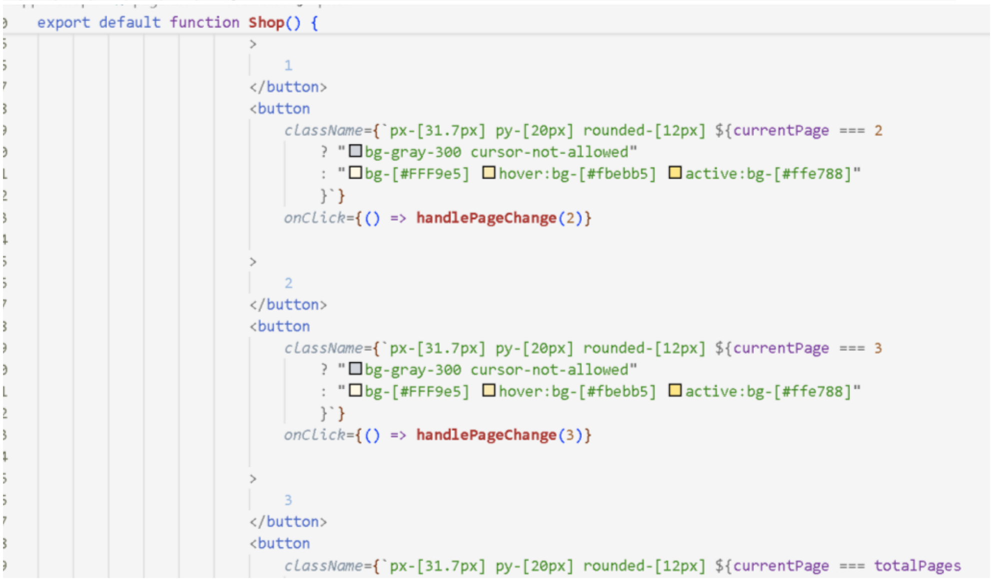


```
export default function Shop() {
                                onChange={(e) => setSearchTerm(e.target.value)}
                            />
                        </div>
                        <i className="fa-solid fa-magnifying-glass"></i>
                        <div className="h-[40px] w-[3px] ■bg-gray-400"></div>
                        <p className="poppins font-[400] text-[16px]">  {`Showing ${(currentPage - 1) * itemsPerPage + 1}
                            } of ${filteredProducts.length} results`}</p>
                    </div>
                    <div className="w-[45%] flex items-center justify-evenly poppins font-[400] text-[20px]">
                        <p>Show</p>
                        <input className="p-4 w-[55px] h-[55px]" type="text" value={itemsPerPage || ""} placeholder="16" c
                            const value = e.target.value
                            if (value === "") {
                                setitemsPerPage(0);
                                setCurrentPage(1);
                            }
                            const parsedValue = parseInt(value, 10);
                            if (!isNaN(parsedValue) && parsedValue > 0) {
                                setitemsPerPage(parsedValue);
                                setCurrentPage(1);
                            }

                        }} />
                        <p>Sort by</p>
```

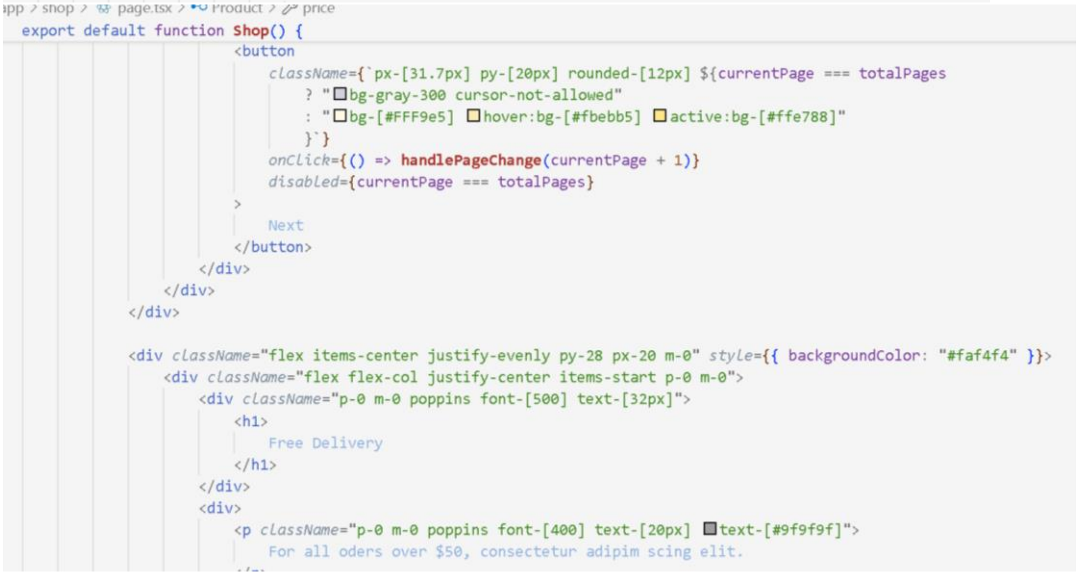
The filtered products state sliced and saved in currentProduct variable which was mapped



```
export default function Shop() {
                    </div>
                </div>
            </div>

            <div className="flex items-center justify-center w-full">
                <div className="py-11 grid grid-cols-4 gap-5 w-[88%]">
                    {
                        currentProducts.map((product) => (
                            <ProductCard key={product._id} product={product} />
                        ))
                    }
                </div>
            </div>

            <div className="w-full flex items-center justify-center py-12" >
                <div className="w-2/5 flex items-center justify-evenly poppins font-[400] text-[20px]" >
                    <div className="flex items-center space-x-4">
                        <button
                            className={`px-[31.7px] py-[20px] rounded-[12px] ${currentPage === 1
                                ? "■bg-gray-300 cursor-not-allowed"
                                : "□bg-[#FFF9e5] □hover:bg-[#fbebb5] □active:bg-[#ffe788]"
                                }`}
                            onClick={() => handlePageChange(1)}
```



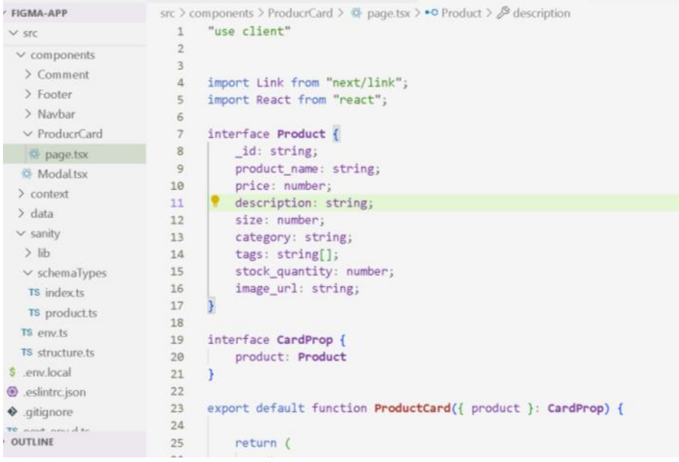
```
export default function Shop() {
                        >
                            1
                        </button>
                        <button
                            className={`px-[31.7px] py-[20px] rounded-[12px] ${currentPage === 2
                                ? "■bg-gray-300 cursor-not-allowed"
                                : "□bg-[#FFF9e5] □hover:bg-[#fbebb5] □active:bg-[#ffe788]"
                                }`}
                            onClick={() => handlePageChange(2)}

                        >
                            2
                        </button>
                        <button
                            className={`px-[31.7px] py-[20px] rounded-[12px] ${currentPage === 3
                                ? "■bg-gray-300 cursor-not-allowed"
                                : "□bg-[#FFF9e5] □hover:bg-[#fbebb5] □active:bg-[#ffe788]"
                                }`}
                            onClick={() => handlePageChange(3)}

                        >
                            3
                        </button>
                        <button
                            className={`px-[31.7px] py-[20px] rounded-[12px] ${currentPage === totalPages
```



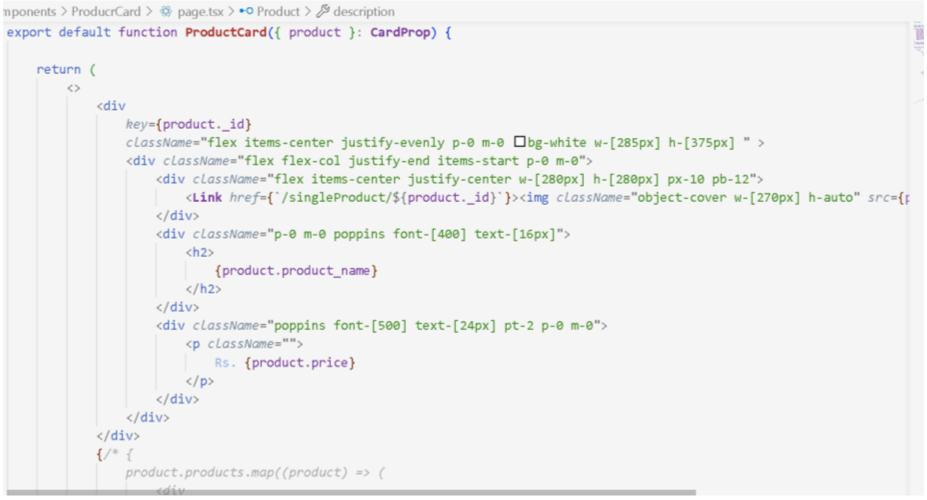
```
export default function Shop() {
                        <button
                            className={`px-[31.7px] py-[20px] rounded-[12px] ${currentPage === totalPages
                                ? "■bg-gray-300 cursor-not-allowed"
                                : "□bg-[#FFF9e5] □hover:bg-[#fbebb5] □active:bg-[#ffe788]"
                                }`}
                            onClick={() => handlePageChange(currentPage + 1)}
                            disabled={currentPage === totalPages}
                        >
                            Next
                        </button>
                    </div>
                </div>
            </div>

            <div className="flex items-center justify-evenly py-28 px-20 m-0" style={{ backgroundColor: "#faf4f4" }}>
                <div className="flex flex-col justify-center items-start p-0 m-0">
                    <div className="p-0 m-0 poppins font-[500] text-[32px]">
                        <h1>
                            Free Delivery
                        </h1>
                    </div>
                    <div>
                        <p className="p-0 m-0 poppins font-[400] text-[20px] ■text-[#9f9f9f]">
                            For all oders over $50, consectetur adipim scing elit.
```

The card component which takes the product data as a prop and shows the data dynamically.

In the link the dynamic path is passed to href to show that particular product

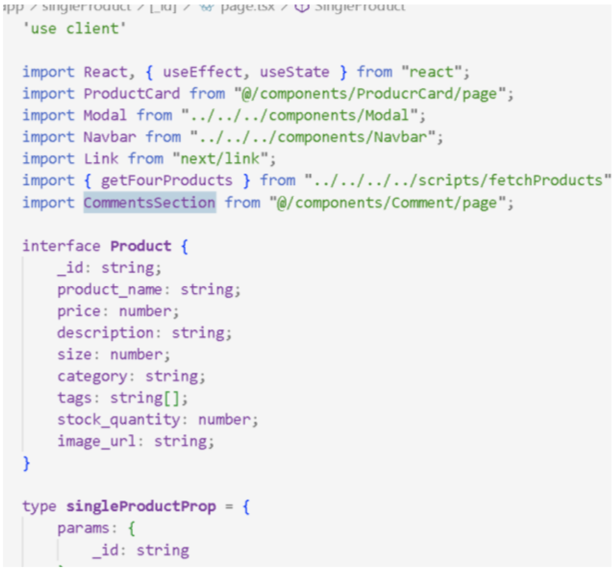


```
"use client"


import Link from "next/link";
import React from "react";

interface Product {
    _id: string;
    product_name: string;
    price: number;
    description: string;
    size: number;
    category: string;
    tags: string[];
    stock_quantity: number;
    image_url: string;
}

interface CardProp {
    product: Product
}

export default function ProductCard({ product }: CardProp) {

    return (
```

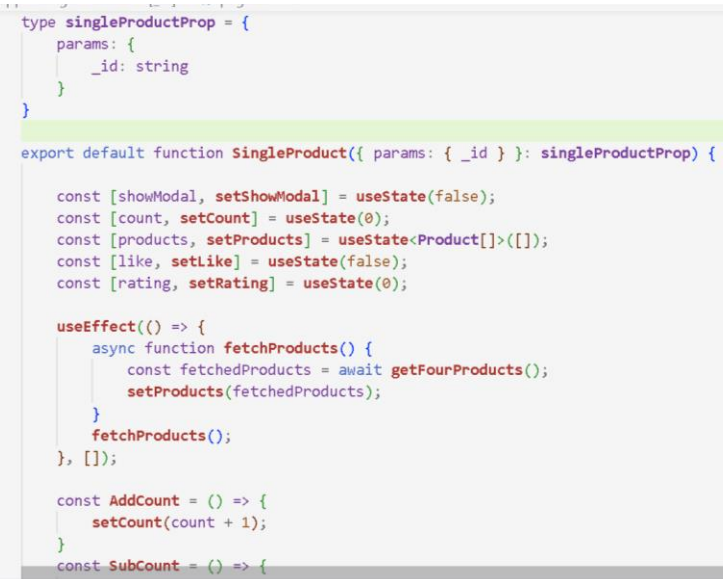


```
export default function ProductCard({ product }: CardProp) {

    return (
        <>
            <div
                key={product._id}
                className="flex items-center justify-evenly p-0 m-0 □bg-white w-[285px] h-[375px] " >
                <div className="flex flex-col justify-end items-start p-0 m-0">
                    <div className="flex items-center justify-center w-[280px] h-[280px] px-10 pb-12">
                        <Link href={`/singleProduct/${product._id}`}><img className="object-cover w-[270px] h-auto" src={p
                    </div>
                    <div className="p-0 m-0 poppins font-[400] text-[16px]">
                        <h2>
                            {product.product_name}
                        </h2>
                    </div>
                    <div className="poppins font-[500] text-[24px] pt-2 p-0 m-0">
                        <p className="">
                            Rs. {product.price}
                        </p>
                    </div>
                </div>
            </div>
            {/* {
                product.products.map((product) => (
                    <div
```

To show the details of single product page the id was taken from the product and was set to dynamic routing, to show the particular page the id was passed as prop to the singleProduct component

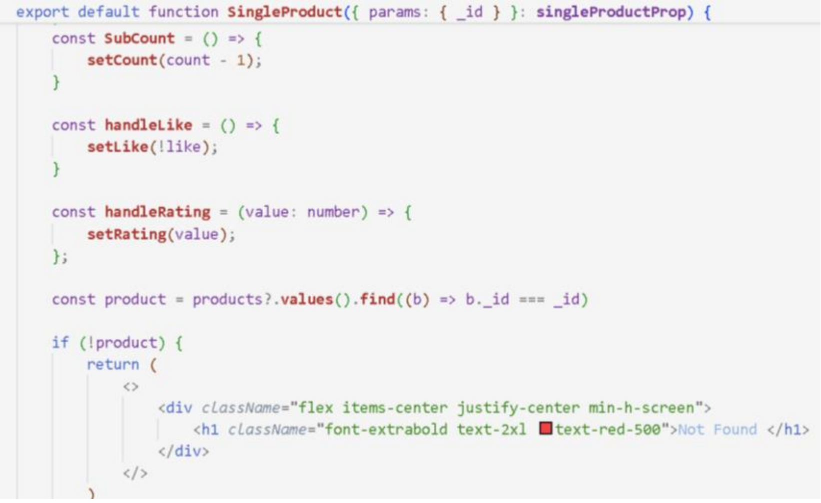


```
pp > singleProduct > [_id] > page.tsx > SingleProduct
'use client'

import React, { useEffect, useState } from "react";
import ProductCard from "@/components/ProducrCard/page";
import Modal from "../../../components/Modal";
import Navbar from "../../../components/Navbar";
import Link from "next/link";
import { getFourProducts } from "../../../../scripts/fetchProducts"
import CommentsSection from "@/components/Comment/page";

interface Product {
    _id: string;
    product_name: string;
    price: number;
    description: string;
    size: number;
    category: string;
    tags: string[];
    stock_quantity: number;
    image_url: string;
}

type singleProductProp = {
    params: {
        _id: string
```

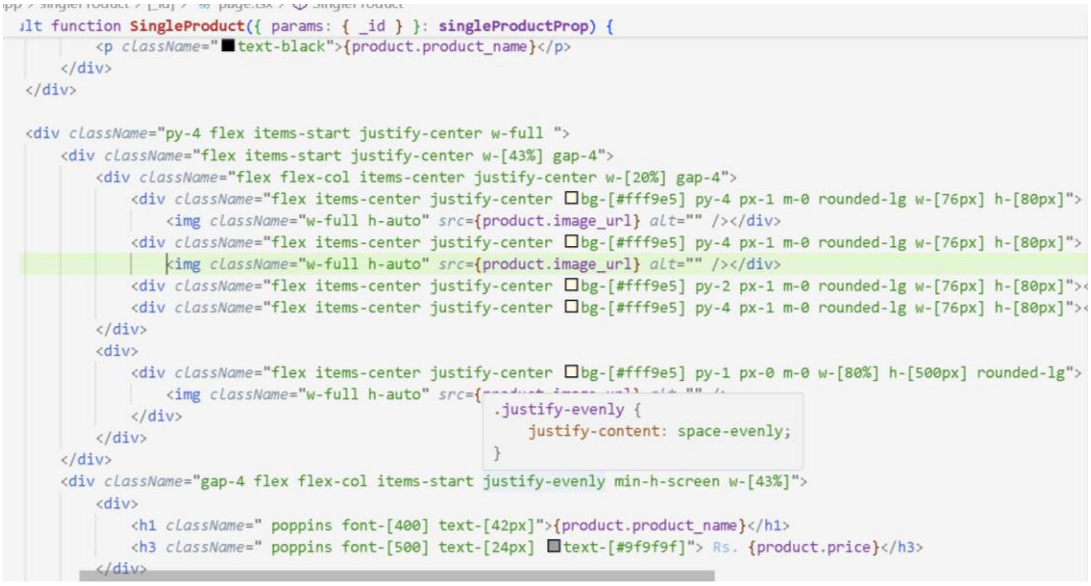


```
type singleProductProp = {
    params: {
        _id: string
    }
}

export default function SingleProduct({ params: { _id } }: singleProductProp) {

    const [showModal, setShowModal] = useState(false);
    const [count, setCount] = useState(0);
    const [products, setProducts] = useState<Product[]>([]);
    const [like, setLike] = useState(false);
    const [rating, setRating] = useState(0);

    useEffect(() => {
        async function fetchProducts() {
            const fetchedProducts = await getFourProducts();
            setProducts(fetchedProducts);
        }
        fetchProducts();
    }, []);

    const AddCount = () => {
        setCount(count + 1);
    }
    const SubCount = () => {
```

States were defined to implement star rating, like button, count to update quantity, the data was fetched from the function in useEffect

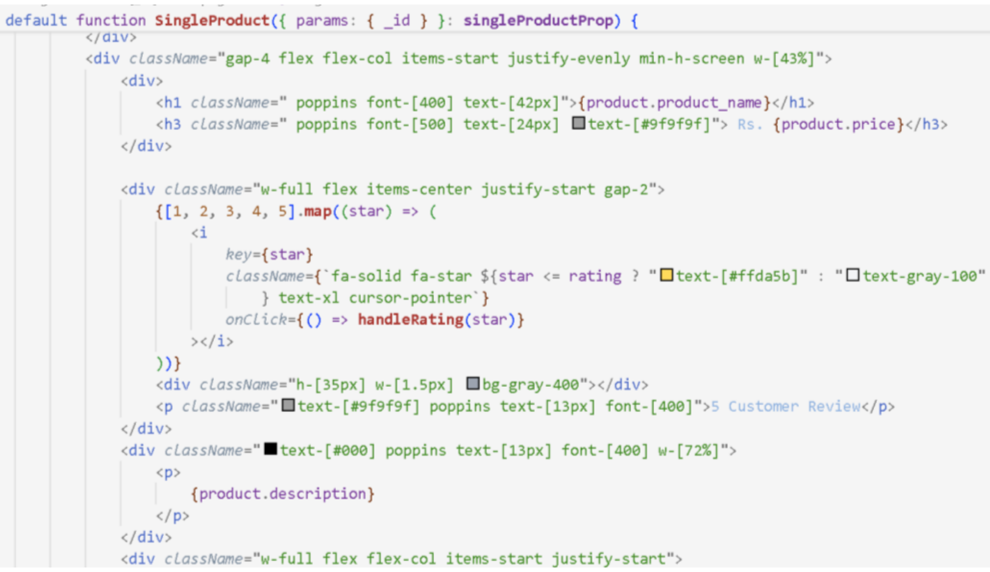


```
export default function SingleProduct({ params: { _id } }: singleProductProp) {
    const SubCount = () => {
        setCount(count - 1);
    }

    const handleLike = () => {
        setLike(!like);
    }

    const handleRating = (value: number) => {
        setRating(value);
    };

    const product = products?.values().find((b) => b._id === _id)

    if (!product) {
        return (
            <>
                <div className="flex items-center justify-center min-h-screen">
                    <h1 className="font-extrabold text-2xl 🟥text-red-500">Not Found </h1>
                </div>
            </>
        )
```

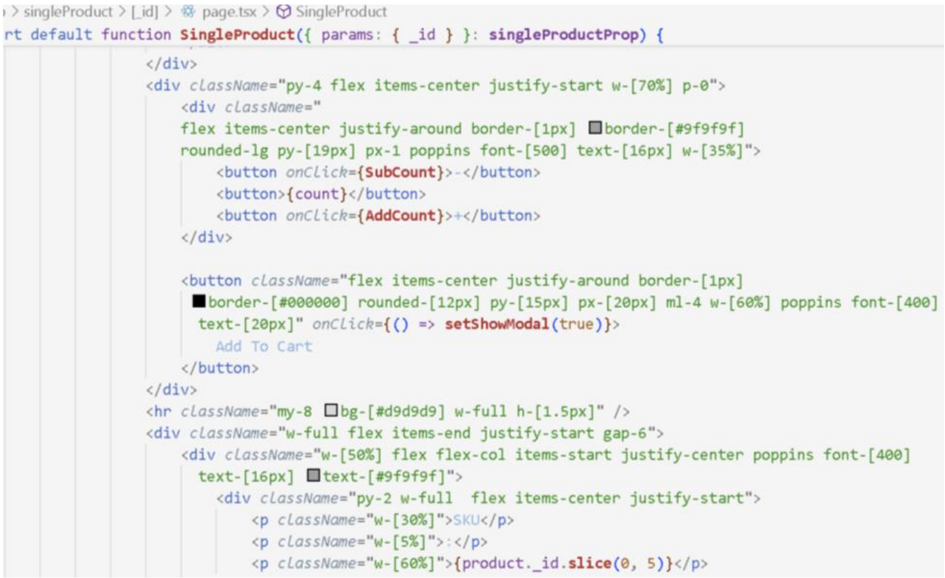


```
ult function SingleProduct({ params: { _id } }: singleProductProp) {
            <p className="■text-black">{product.product_name}</p>
        </div>
    </div>

    <div className="py-4 flex items-start justify-center w-full ">
        <div className="flex items-start justify-center w-[43%] gap-4">
            <div className="flex flex-col items-center justify-center w-[20%] gap-4">
                <div className="flex items-center justify-center □bg-[#fff9e5] py-4 px-1 m-0 rounded-lg w-[76px] h-[80px]">
                    <img className="w-full h-auto" src={product.image_url} alt="" /></div>
                <div className="flex items-center justify-center □bg-[#fff9e5] py-4 px-1 m-0 rounded-lg w-[76px] h-[80px]">
                    <img className="w-full h-auto" src={product.image_url} alt="" /></div>
                <div className="flex items-center justify-center □bg-[#fff9e5] py-2 px-1 m-0 rounded-lg w-[76px] h-[80px]"><
                <div className="flex items-center justify-center □bg-[#fff9e5] py-4 px-1 m-0 rounded-lg w-[76px] h-[80px]"><
            </div>
            <div>
                <div className="flex items-center justify-center □bg-[#fff9e5] py-1 px-0 m-0 w-[80%] h-[500px] rounded-lg">
                    <img className="w-full h-auto" src={
                </div>
            </div>
        </div>
        <div className="gap-4 flex flex-col items-start justify-evenly min-h-screen w-[43%]">
            <div>
                <h1 className=" poppins font-[400] text-[42px]">{product.product_name}</h1>
                <h3 className=" poppins font-[500] text-[24px] ■text-[#9f9f9f]"> Rs. {product.price}</h3>
            </div>
```

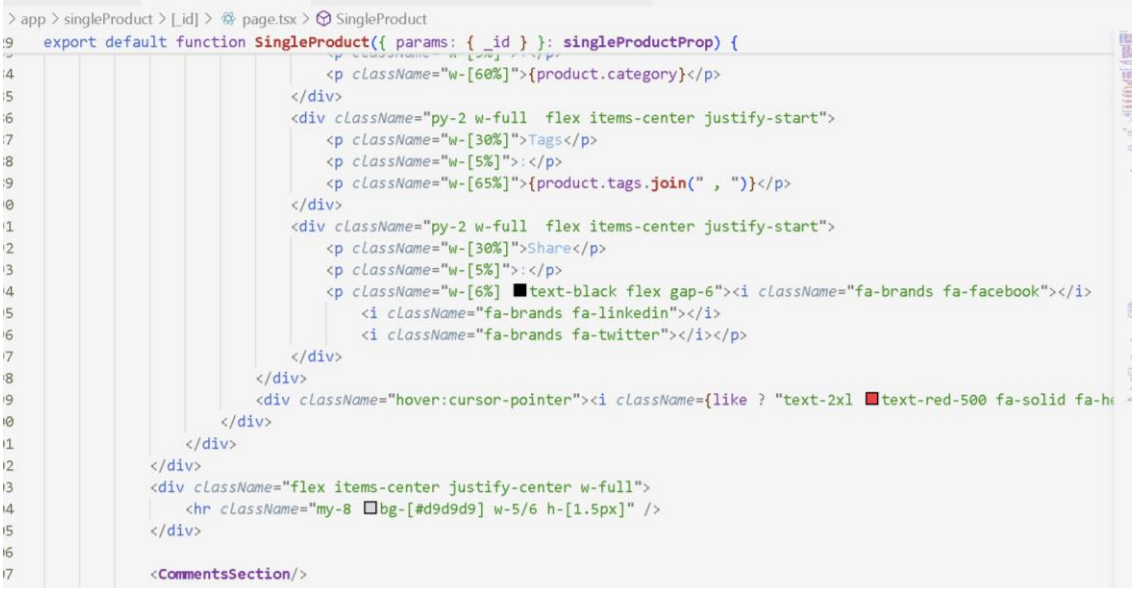
```
.justify-evenly {
    justify-content: space-evenly;
}
```



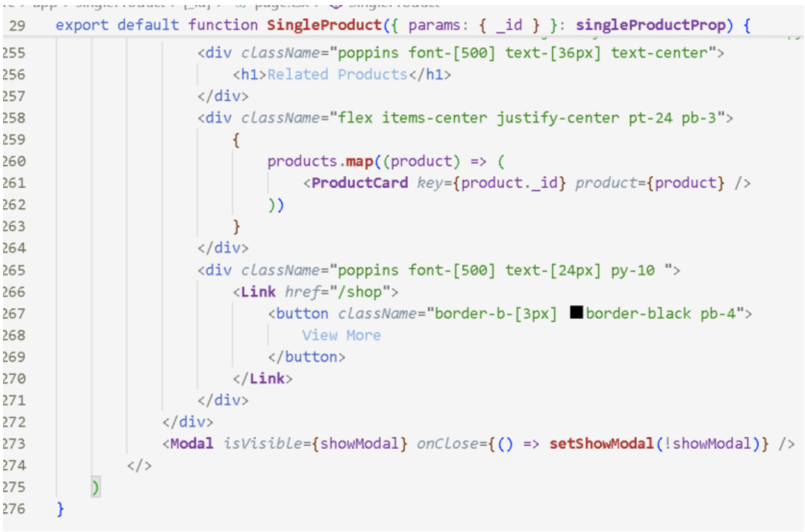
```
default function SingleProduct({ params: { _id } }: singleProductProp) {
        </div>
        <div className="gap-4 flex flex-col items-start justify-evenly min-h-screen w-[43%]">
            <div>
                <h1 className=" poppins font-[400] text-[42px]">{product.product_name}</h1>
                <h3 className=" poppins font-[500] text-[24px] ■text-[#9f9f9f]"> Rs. {product.price}</h3>
            </div>

            <div className="w-full flex items-center justify-start gap-2">
                {[1, 2, 3, 4, 5].map((star) => (
                    <i
                        key={star}
                        className={`fa-solid fa-star ${star <= rating ? "🟨text-[#ffda5b]" : "□text-gray-100"
                            } text-xl cursor-pointer`}
                        onClick={() => handleRating(star)}
                    ></i>
                ))}
                <div className="h-[35px] w-[1.5px] ■bg-gray-400"></div>
                <p className="■text-[#9f9f9f] poppins text-[13px] font-[400]">5 Customer Review</p>
            </div>
            <div className="■text-[#000] poppins text-[13px] font-[400] w-[72%]">
                <p>
                    {product.description}
                </p>
            </div>
            <div className="w-full flex flex-col items-start justify-start">
```



```
› > singleProduct > [_id] > page.tsx > SingleProduct
rt default function SingleProduct({ params: { _id } }: singleProductProp) {
                </div>
                <div className="py-4 flex items-center justify-start w-[70%] p-0">
                    <div className="
                    flex items-center justify-around border-[1px] ■border-[#9f9f9f]
                    rounded-lg py-[19px] px-1 poppins font-[500] text-[16px] w-[35%]">
                        <button onClick={SubCount}>-</button>
                        <button>{count}</button>
                        <button onClick={AddCount}>+</button>
                    </div>

                    <button className="flex items-center justify-around border-[1px]
                     ■border-[#000000] rounded-[12px] py-[15px] px-[20px] ml-4 w-[60%] poppins font-[400]
                      text-[20px]" onClick={() => setShowModal(true)}>
                        Add To Cart
                    </button>
                </div>
                <hr className="my-8 □bg-[#d9d9d9] w-full h-[1.5px]" />
                <div className="w-full flex items-end justify-start gap-6">
                    <div className="w-[50%] flex flex-col items-start justify-center poppins font-[400]
                      text-[16px] ■text-[#9f9f9f]">
                        <div className="py-2 w-full  flex items-center justify-start">
                            <p className="w-[30%]">SKU</p>
                            <p className="w-[5%]">:</p>
                            <p className="w-[60%]">{product._id.slice(0, 5)}</p>
```

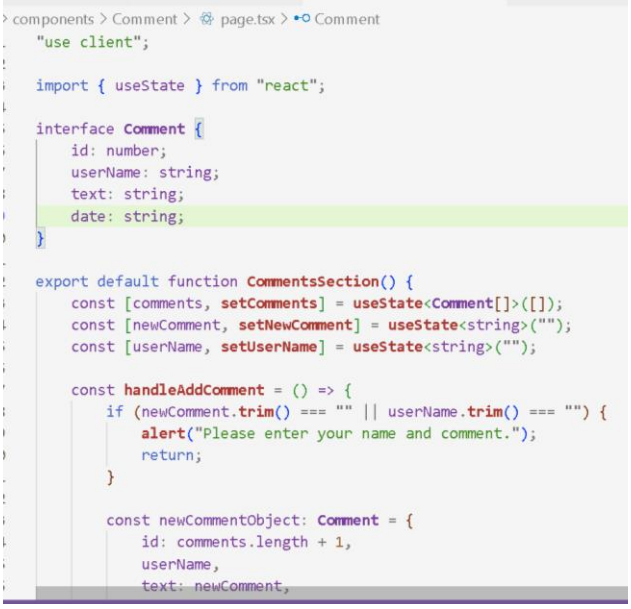


```
› app > singleProduct > [_id] > page.tsx > SingleProduct
   export default function SingleProduct({ params: { _id } }: singleProductProp) {
                                <p className="w-[60%]">{product.category}</p>
                            </div>
                            <div className="py-2 w-full  flex items-center justify-start">
                                <p className="w-[30%]">Tags</p>
                                <p className="w-[5%]">:</p>
                                <p className="w-[65%]">{product.tags.join(" , ")}</p>
                            </div>
                            <div className="py-2 w-full  flex items-center justify-start">
                                <p className="w-[30%]">Share</p>
                                <p className="w-[5%]">:</p>
                                <p className="w-[6%] ■text-black flex gap-6"><i className="fa-brands fa-facebook"></i>
                                    <i className="fa-brands fa-linkedin"></i>
                                    <i className="fa-brands fa-twitter"></i></p>
                            </div>
                        </div>
                        <div className="hover:cursor-pointer"><i className={like ? "text-2xl ■text-red-500 fa-solid fa-he
                    </div>
                </div>
            </div>
            <div className="flex items-center justify-center w-full">
                <hr className="my-8 □bg-[#d9d9d9] w-5/6 h-[1.5px]" />
            </div>

            <CommentsSection/>
```



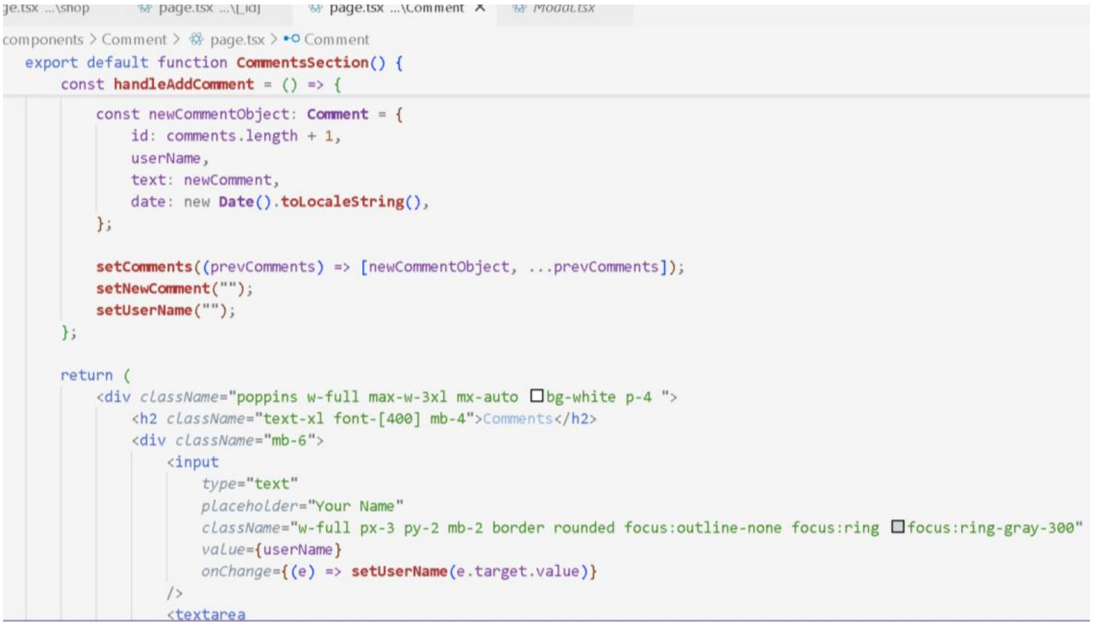
```
   export default function SingleProduct({ params: { _id } }: singleProductProp) {
                <div className="poppins font-[500] text-[36px] text-center">
                    <h1>Related Products</h1>
                </div>
                <div className="flex items-center justify-center pt-24 pb-3">
                    {
                        products.map((product) => (
                            <ProductCard key={product._id} product={product} />
                        ))
                    }
                </div>
                <div className="poppins font-[500] text-[24px] py-10 ">
                    <Link href="/shop">
                        <button className="border-b-[3px] ■border-black pb-4">
                            View More
                        </button>
                    </Link>
                </div>
            </div>
            <Modal isVisible={showModal} onClose={() => setShowModal(!showModal)} />
        </>
    )
}
```

The comments were set in a state, with user name and time of the comment



components > Comment > page.tsx > Comment

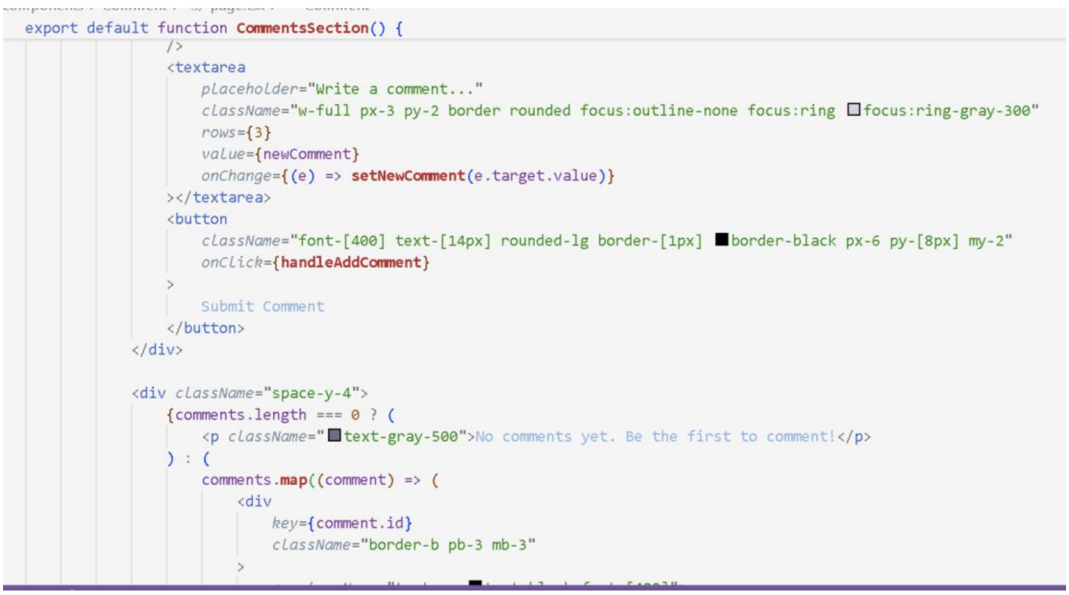
```
"use client";

import { useState } from "react";

interface Comment {
    id: number;
    userName: string;
    text: string;
    date: string;
}

export default function CommentsSection() {
    const [comments, setComments] = useState<Comment[]>([]);
    const [newComment, setNewComment] = useState<string>("");
    const [userName, setUserName] = useState<string>("");

    const handleAddComment = () => {
        if (newComment.trim() === "" || userName.trim() === "") {
            alert("Please enter your name and comment.");
            return;
        }

        const newCommentObject: Comment = {
            id: comments.length + 1,
            userName,
            text: newComment,
```



ge.tsx ...\shop | page.tsx ...\[_id] | page.tsx ...\Comment × | Modal.tsx

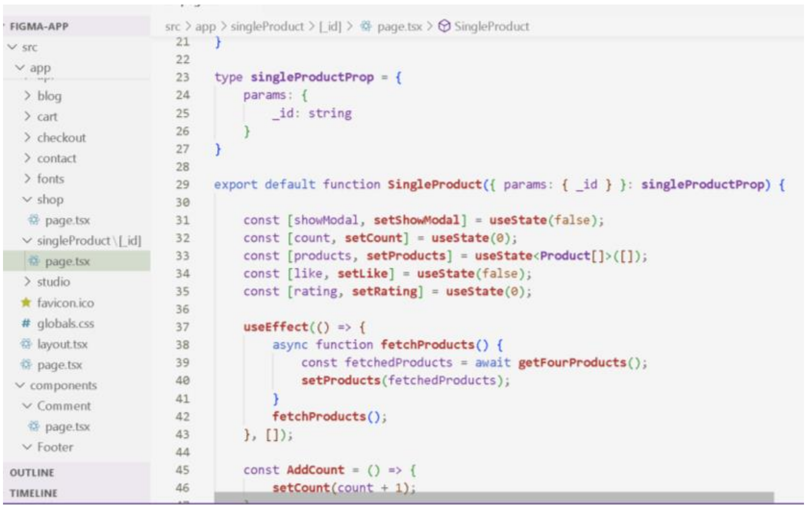
components > Comment > page.tsx > Comment

```
export default function CommentsSection() {
    const handleAddComment = () => {

        const newCommentObject: Comment = {
            id: comments.length + 1,
            userName,
            text: newComment,
            date: new Date().toLocaleString(),
        };

        setComments((prevComments) => [newCommentObject, ...prevComments]);
        setNewComment("");
        setUserName("");
    };

    return (
        <div className="poppins w-full max-w-3xl mx-auto □bg-white p-4 ">
            <h2 className="text-xl font-[400] mb-4">Comments</h2>
            <div className="mb-6">
                <input
                    type="text"
                    placeholder="Your Name"
                    className="w-full px-3 py-2 mb-2 border rounded focus:outline-none focus:ring □focus:ring-gray-300"
                    value={userName}
                    onChange={(e) => setUserName(e.target.value)}
                />
                <textarea
```



```
export default function CommentsSection() {
                />
                <textarea
                    placeholder="Write a comment..."
                    className="w-full px-3 py-2 border rounded focus:outline-none focus:ring □focus:ring-gray-300"
                    rows={3}
                    value={newComment}
                    onChange={(e) => setNewComment(e.target.value)}
                ></textarea>
                <button
                    className="font-[400] text-[14px] rounded-lg border-[1px] ■border-black px-6 py-[8px] my-2"
                    onClick={handleAddComment}
                >
                    Submit Comment
                </button>
            </div>

            <div className="space-y-4">
                {comments.length === 0 ? (
                    <p className="■text-gray-500">No comments yet. Be the first to comment!</p>
                ) : (
                    comments.map((comment) => (
                        <div
                            key={comment.id}
                            className="border-b pb-3 mb-3"
                        >
```

The dynamic route of the singleProduct page



```
}

type singleProductProp = {
    params: {
        _id: string
    }
}

export default function SingleProduct({ params: { _id } }: singleProductProp) {

    const [showModal, setShowModal] = useState(false);
    const [count, setCount] = useState(0);
    const [products, setProducts] = useState<Product[]>([]);
    const [like, setLike] = useState(false);
    const [rating, setRating] = useState(0);

    useEffect(() => {
        async function fetchProducts() {
            const fetchedProducts = await getFourProducts();
            setProducts(fetchedProducts);
        }
        fetchProducts();
    }, []);

    const AddCount = () => {
        setCount(count + 1);
```