

**Day 4 - Dynamic Frontend Components-
Aromas**

Asaad Hussain

00414030

GIAIC, Quarter 2

20th January 2025

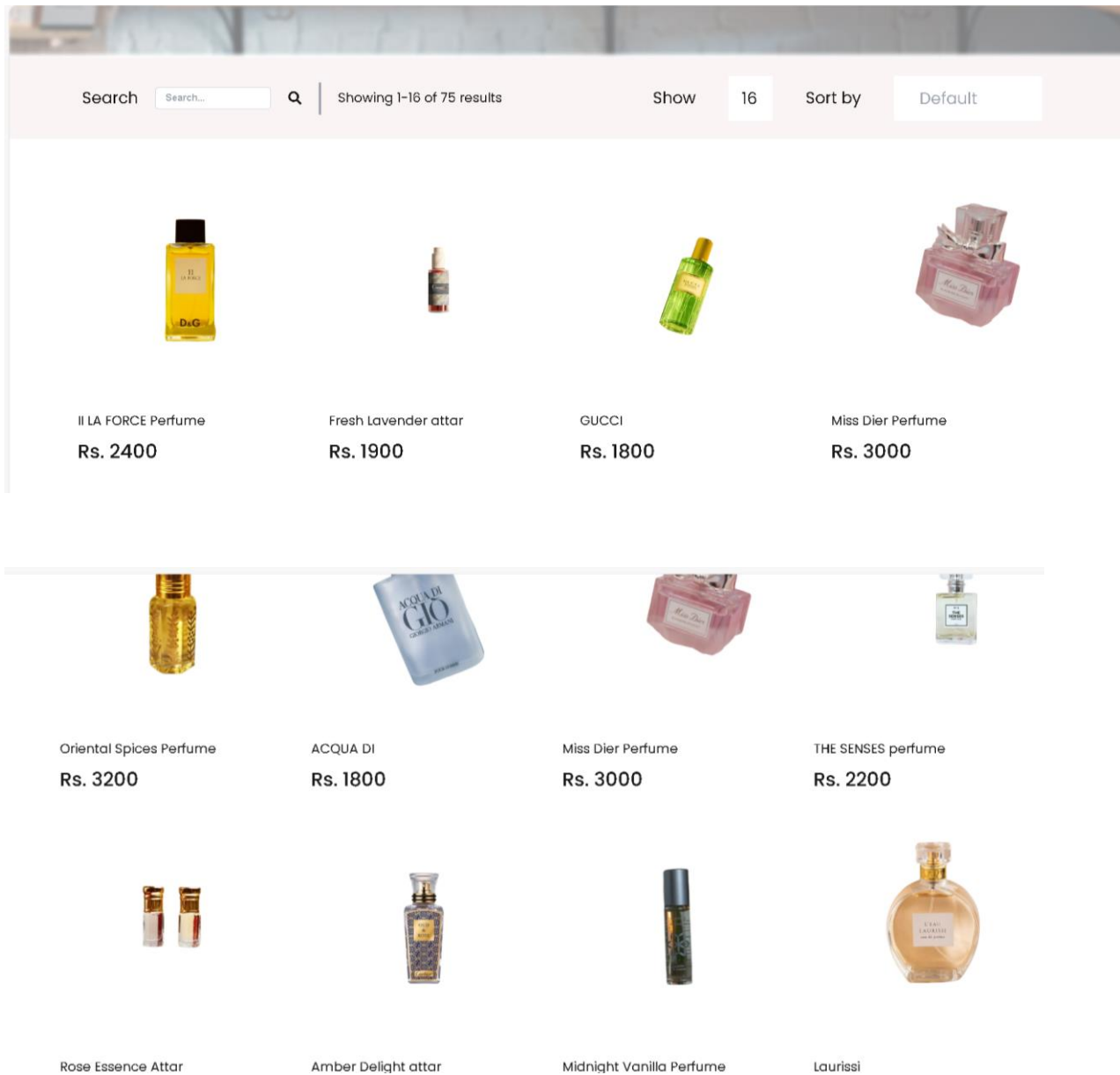
Table of Content

Functional Deliverables	3
Code Deliverables	7

Functional Deliverables

Screenshots

Product Listing



Product detail dynamic



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



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
Q

Showing 1 - 1 of 75 results

Show

1

Sort by
Default



II LA FORCE Perfume

Rs. 2400

e.g. 3

Search

Q

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

Q

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 Dior Perfume
Rs. 3000



THE SENSES perfume
Rs. 2200

1

2

3

Next

Third page



Rose Essence Attar
Rs. 1200



Amber Delight attar
Rs. 1500



Midnight Vanilla Perfume
Rs. 3000



Laurissi
Rs. 2500

1

2

3

Next

Rating of product



II LA FORCE Perfume

Rs. 2400

★★★★☆ 5 Customer Review

A zesty citrus fragrance that energizes your day.

Size

S

M

L

Color

Blue

Black

Gold

Increasing product quantity



Color

Blue

Black



Gold

- 4 +

Add To Cart

Like Product

SKU : 9piJ0
Category : EDP
Tags : male , perfume
Share :    

SKU : 9piJ0
Category : EDP
Tags : male , perfume
Share :    

Comments

Comments

No comments yet. Be the first to comment!

Comments

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

Code Deliverables

Snippets and Documentation

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

```

1  import { client } from "../src/sanity/lib/client"
2
3  export const getFourProducts = async () => {
4    const products = await client.fetch(
5      `*[_type=="product"]{0..3}{
6        _id,
7        product_name,
8        price,
9        description,
10       size,
11       category,
12       tags,
13       stock_quantity,
14       "image_url":image.asset->url
15     }`
16   )
17   return products
18 }
19
20 export const getAllProducts = async () => {
21   const products = await client.fetch(
22     `*[_type=="product"]{
23       _id,
24       product_name,
25       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

```

src > app > shop > page.tsx > Product > price
1  "use client"
2
3  import { useEffect, useState } from "react";
4  import { getAllProducts } from "../../scripts/fetchProducts";
5  import Navbar from "../../components/Navbar";
6  import ProductCard from "@components/ProducrCard/page";
7
8  interface Product {
9    _id: string;
10   product_name: string;
11   price: number;
12   description: string;
13   size: number;
14   category: string;
15   tags: string[];
16   stock_quantity: number;
17   image_url: string;
18 }
19

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

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

```



```

c > app > shop > page.tsx > Product > price
20 export default function Shop() {
21   const [products, setProducts] = useState<Product[]>([]);
22   const [loading, setLoading] = useState(true);
23   const [currentPage, setCurrentPage] = useState(1);
24   const [itemsPerPage, setItemsPerPage] = useState(16);
25   const [searchTerm, setSearchTerm] = useState("");
26   const [filteredProducts, setFilteredProducts] = useState<Product[]>([]);
27
28   useEffect(() => {
29     async function fetchProducts() {
30       const fetchedProducts = await getAllProducts();
31       setProducts(fetchedProducts);
32       setFilteredProducts(fetchedProducts);
33       setLoading(false);
34     }
35     fetchProducts();
36   }, []);
37
38   useEffect(() => {

```

```

app > shop > page.tsx > Product > price
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">
          

```

```

export default function Shop() {

  <div className="flex flex-col justify-center items-center p-0 m-0 w-[35%]">
    
    <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 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}
          </div>
        </div>
        <i className="fa-solid fa-magnifying-glass"></i>
      </div>
      <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" >
      <div>
        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);
        }
      </div>
    </div>
  </div>
  <p>Sort by</p>

```

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


```

src > components > ProductCard > page.tsx > Product > description
1  "use client"
2
3
4  import Link from "next/link";
5  import React from "react";
6
7  interface Product {
8    _id: string;
9    product_name: string;
10   price: number;
11   description: string;
12   size: number;
13   category: string;
14   tags: string[];
15   stock_quantity: number;
16   image_url: string;
17 }
18
19 interface CardProp {
20   product: Product
21 }
22
23 export default function ProductCard({ product }: CardProp) {
24
25   return (

```

```

    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={product.image_url} alt="Product Image" />
          </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>
    )
  }
}

```

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

```

app / singleproduct / [id] / page.jsx / 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>
      </>
    )
  }
}

```

```

<div>
  <p className="text-black">{product.product_name}</p>
</div>

<div className="py-4 flex items-start justify-center w-full">
  <div className="flex items-start justify-center w-[43%]>
    <div className="flex flex-col items-center justify-center w-[20%]>
      <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]>
        <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>
    <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={product.image_url} alt="" />
      </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>

```

```

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>
  <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
9 export default function SingleProduct({ params: { _id } }: singleProductProp) {
14   <p className="w-[60%]">{product.category}</p>
15   </div>
16   <div className="py-2 w-full flex items-center justify-start">
17     <p className="w-[30%]">Tags</p>
18     <p className="w-[5%]">:</p>
19     <p className="w-[65%]">{product.tags.join(" , ")}</p>
20   </div>
21   <div className="py-2 w-full flex items-center justify-start">
22     <p className="w-[30%]">Share</p>
23     <p className="w-[5%]">:</p>
24     <p className="w-[6%]"> <span text-black flex gap-6><i className="fa-brands fa-facebook"></i>
25       <i className="fa-brands fa-linkedin"></i>
26       <i className="fa-brands fa-twitter"></i></p>
27   </div>
28   </div>
29   <div className="hover:cursor-pointer"><i className={like ? "text-2xl text-red-500 fa-solid fa-he
30   </div>
31   </div>
32   <div className="flex items-center justify-center w-full">
33     <hr className="my-8 bg-[#d9d9d9] w-5/6 h-[1.5px]" />
34   </div>
35   <CommentsSection/>

```

```

29 export default function SingleProduct({ params: { _id } }: singleProductProp) {
255   <div className="poppins font-[500] text-[36px] text-center">
256     <h1>Related Products</h1>
257   </div>
258   <div className="flex items-center justify-center pt-24 pb-3">
259     {
260       products.map((product) => (
261         <ProductCard key={product._id} product={product} />
262       ))
263     }
264   </div>
265   <div className="poppins font-[500] text-[24px] py-10">
266     <Link href="/shop">
267       <button className="border-b-[3px] border-black pb-4">
268         View More
269       </button>
270     </Link>
271   </div>
272   </div>
273   <Modal isVisible={showModal} onClose={() => setShowModal(!showModal)} />
274 </>
275 ]
276 }

```

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,

```

```

page.tsx ...\snap page.tsx ...\idj page.tsx ...\Comment MODALSX
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 border rounded focus:outline-none focus:ring focus:ring-gray-300"
          value={userName}
          onChange={(e) => setUserName(e.target.value)}
        />
        <textarea

```

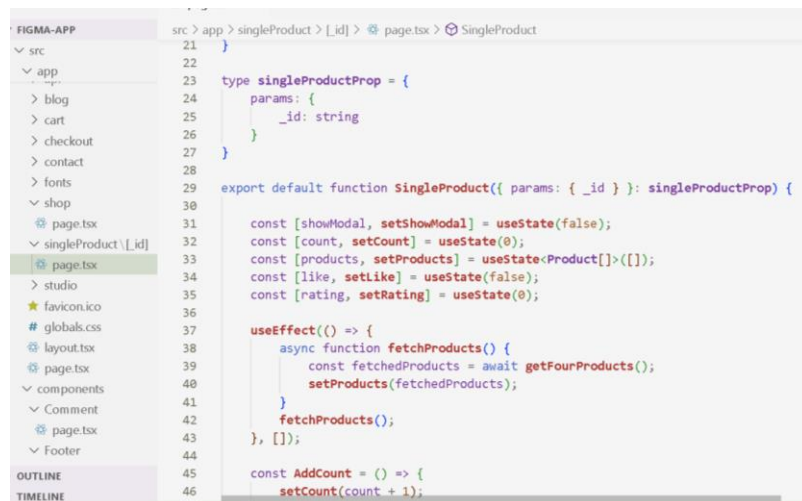
```

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



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'src' directory containing 'app', 'blog', 'cart', 'checkout', 'contact', 'fonts', 'shop', 'page.tsx', 'singleProduct', 'studio', 'favicon.ico', 'globals.css', 'layout.tsx', 'page.tsx', 'components', 'Comment', 'page.tsx', and 'Footer'. The 'singleProduct' directory is expanded, showing 'page.tsx' and 'SingleProduct'. The code editor shows the following code:

```
21 }
22
23 type singleProductProp = {
24   params: {
25     _id: string
26   }
27 }
28
29 export default function SingleProduct({ params: { _id } }: singleProductProp) {
30
31   const [showModal, setShowModal] = useState(false);
32   const [count, setCount] = useState(0);
33   const [products, setProducts] = useState<Product[]>([]);
34   const [like, setLike] = useState(false);
35   const [rating, setRating] = useState(0);
36
37   useEffect(() => {
38     async function fetchProducts() {
39       const fetchedProducts = await getFourProducts();
40       setProducts(fetchedProducts);
41     }
42     fetchProducts();
43   }, []);
44
45   const AddCount = () => {
46     setCount(count + 1);
47   }
48 }
```