Just In

Nike Air Force 1 PLT.AF.ORM

Women's Shoes

Colors: White

MRP : ₹8695

Just In

Nike Air Force 1 React

Men's Shoes

Colors: White

MRP : ₹13295

Just In

Nike Standard Issue Basketball Jersey

Women's Basketball Jersey

Colors: White

MRP : ₹2895

Promo Exclusion

Nike Dunk Low Retro SE

Men's Shoes

Colors: Beige

Sustainable Materials

Nike Dri-FIT UV Hyverse

Men's Short-Sleeve Graphic Fitness Top

Just In

Nike Court Vision Low

Men's Shoes

Colors: White

Find a Store | Help | Join Us | Login

News & Featured Men Women Kids Sale SNKRS

Nike Air Force

Just In

Nike Air Force 1 PLT.AF.ORM

Women's Shoes

Colors: White

MRP : ₹8695

Just In

Nike Air Force 1 React

Men's Shoes

Colors: White

MRP : ₹13295

Just In

Nike Air Force 1 Mid '07

Men's Shoes

Colors: White

MRP : ₹10795

FIND A STORE	GET HELP	ABOUT NIKE
BECOME A MEMBER	Order Status	News
SIGN UP FOR EMAIL	Delivery	Careers
US FEEDBACK SEND	Returns	Investors
STUDENT DISCOUNTS	Payment Options	Sustainability
	Contact Us On Nike.com Inquiries	
	Contact Us On All Other Inquiries	

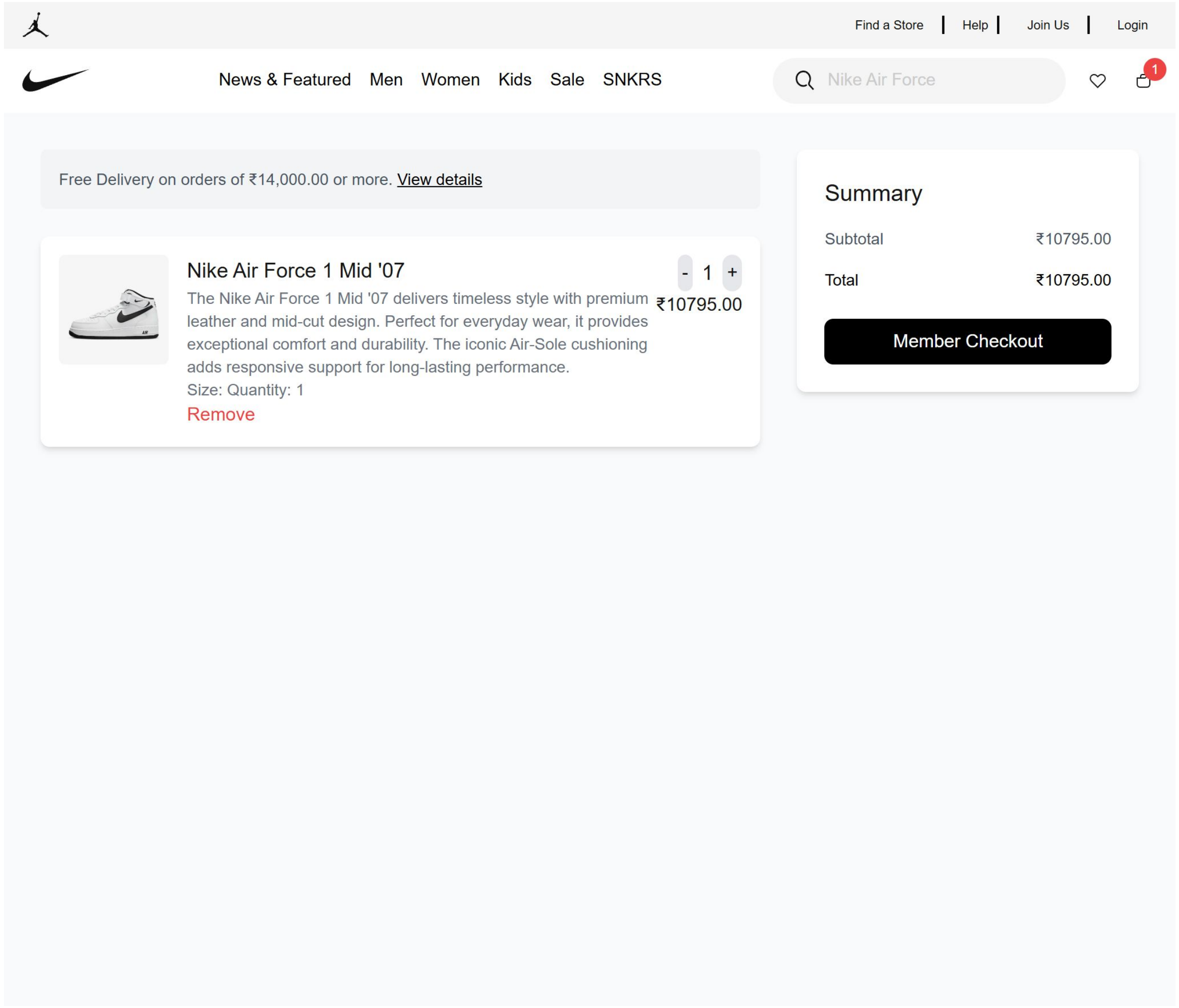
Guides Terms of Sale Terms of Use Nike Privacy Policy

Nike Air Force 1 Mid '07

The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance.

₹10795

Add to Cart



Find a Store | Help | Join Us | Login

News & Featured Men Women Kids Sale SNKRS

Nike Air Force

1

Free Delivery on orders of ₹14,000.00 or more. View details

Nike Air Force 1 Mid '07

The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance.

Size: Quantity: 1

Remove

- 1 +

₹10795.00

Summary

Subtotal	₹10795.00
Total	₹10795.00

Member Checkout

How would you like to get your order?

Custom regulations for India require a copy of the recipient's KYC. The address on the KYC needs to match the shipping address. Our courier will contact you via SMS/email to obtain a copy of your KYC. The KYC will be stored securely and used solely for the purpose of clearing customs (including sharing it with customs officials for all orders and returns). If your KYC does not match your shipping address, please click the link for more information.Learn More

Enter your name and address:

First Name

Last Name

Address Line 1

Address Line 2

Address Line 3

Order Summary

Subtotal: ₹10795

Delivery/Shipping: Free

Total: ₹10795

Arrives: Thu Jan 23 2025

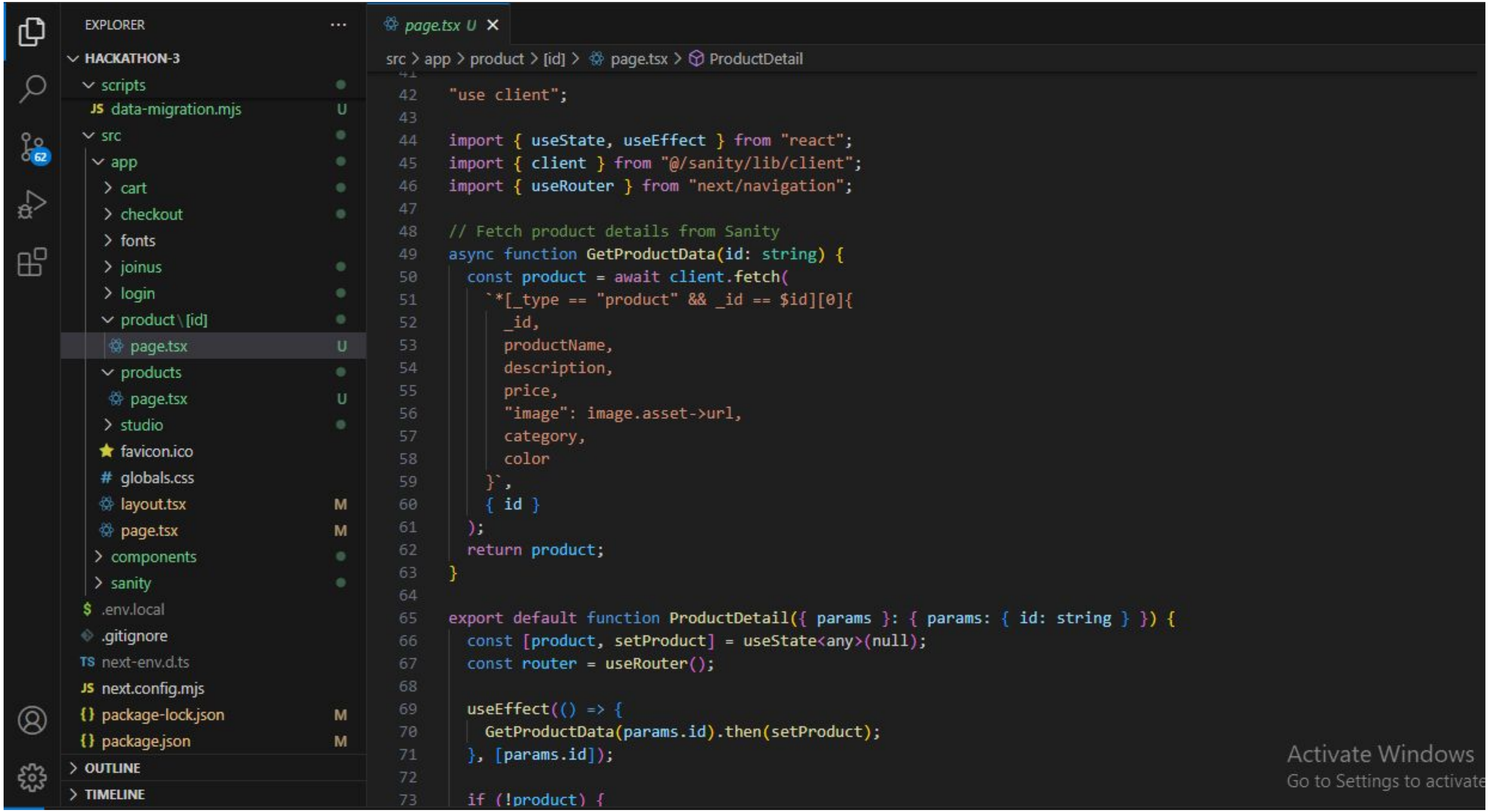
Nike Air Force 1 Mid '07

Men's Shoes

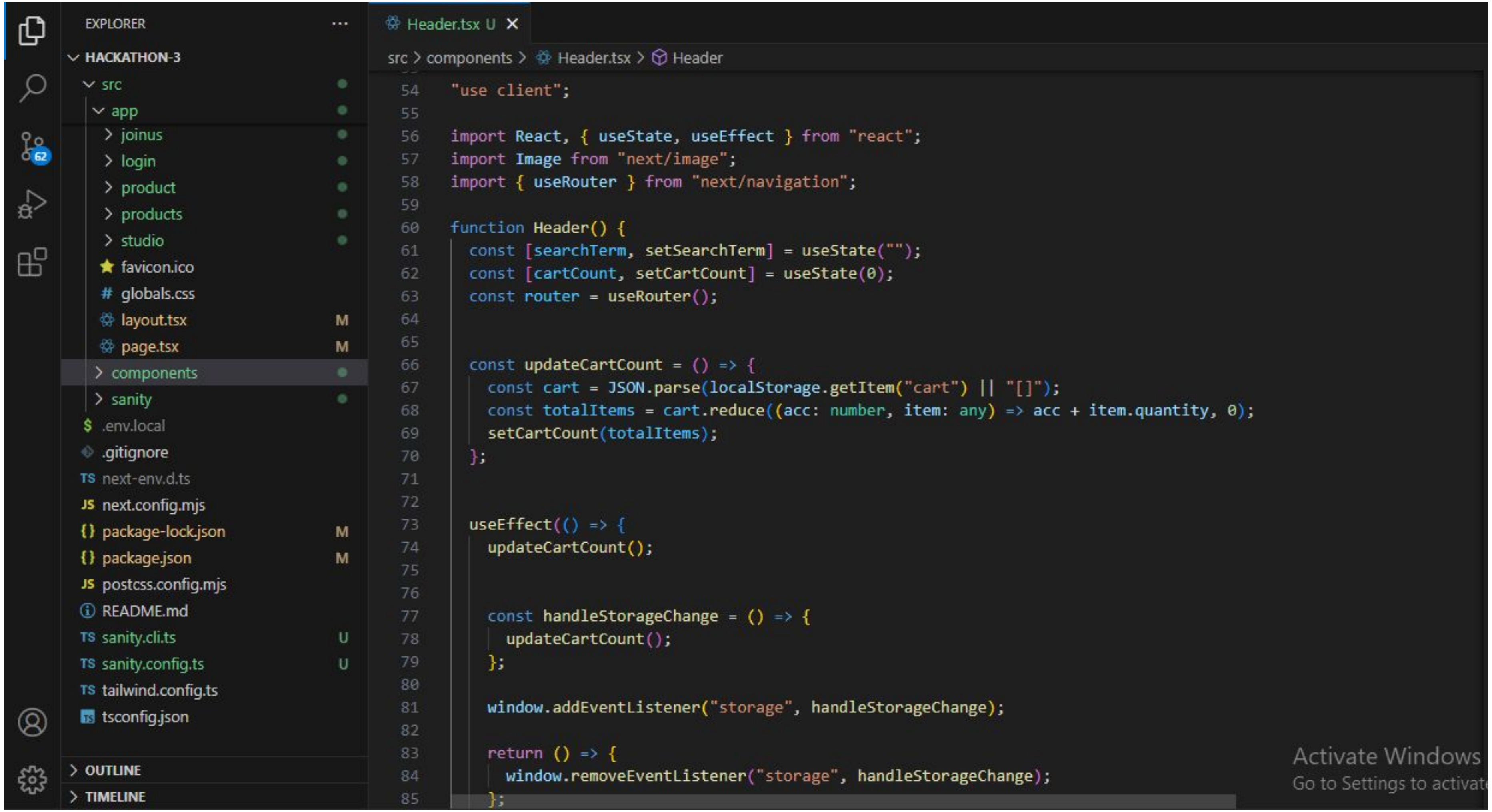
Color:

Quantity: 1

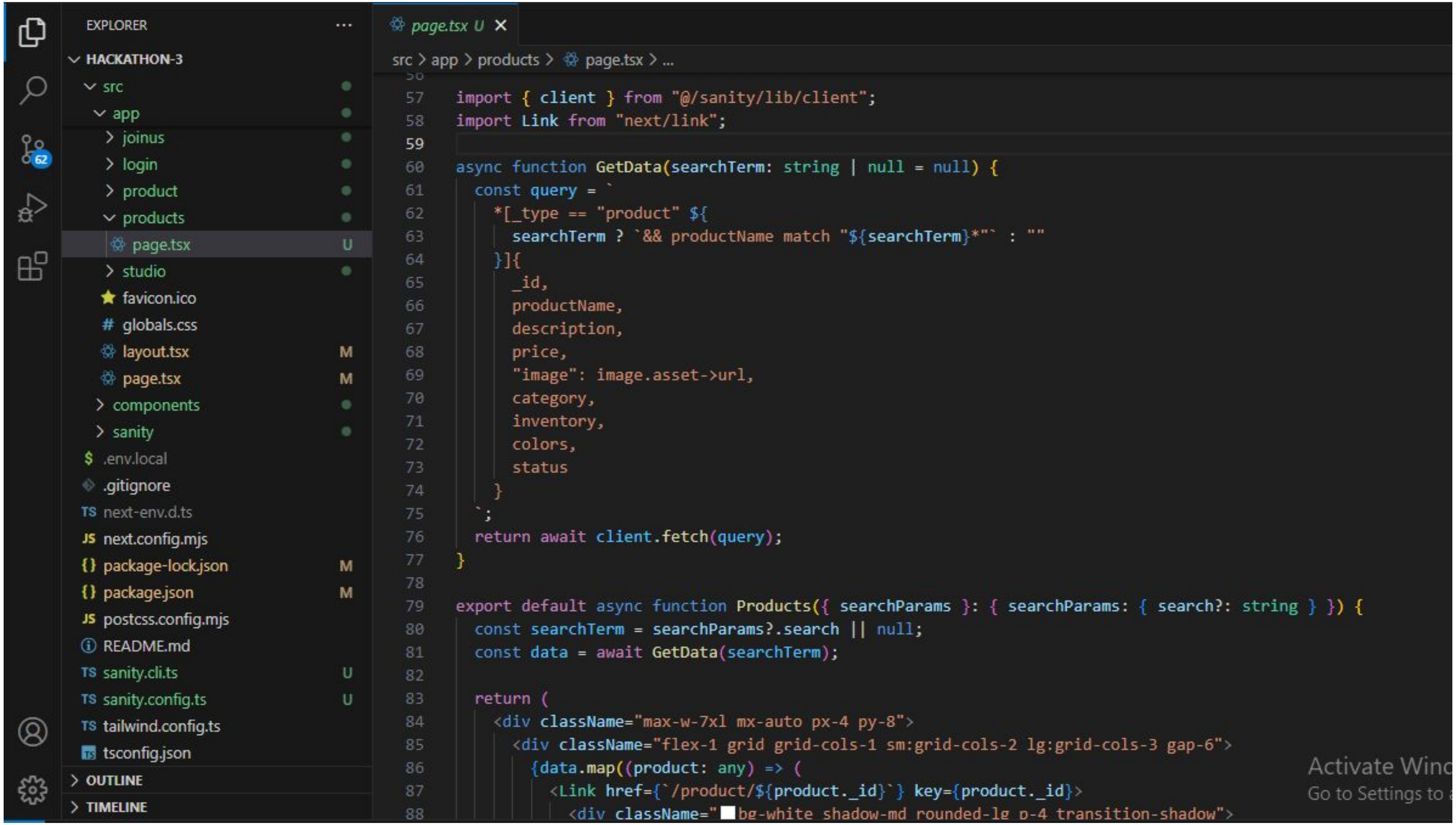
₹10795



```tsx
"use client";

import { useState, useEffect } from "react";
import { client } from "@/sanity/lib/client";
import { useRouter } from "next/navigation";

// Fetch product details from Sanity
async function GetProductData(id: string) {
  const product = await client.fetch(
    `*[_type == "product" && _id == $id][0]{
      _id,
      productName,
      description,
      price,
      "image": image.asset->url,
      category,
      color
    }`,
    { id }
  );
  return product;
}

export default function ProductDetail({ params }: { params: { id: string } }) {
  const [product, setProduct] = useState<any>(null);
  const router = useRouter();

  useEffect(() => {
    GetProductData(params.id).then(setProduct);
  }, [params.id]);

  if (!product) {
```



```tsx
"use client";

import React, { useState, useEffect } from "react";
import Image from "next/image";
import { useRouter } from "next/navigation";

function Header() {
  const [searchTerm, setSearchTerm] = useState("");
  const [cartCount, setCartCount] = useState(0);
  const router = useRouter();


  const updateCartCount = () => {
    const cart = JSON.parse(localStorage.getItem("cart") || "[]");
    const totalItems = cart.reduce((acc: number, item: any) => acc + item.quantity, 0);
    setCartCount(totalItems);
  };


  useEffect(() => {
    updateCartCount();


    const handleStorageChange = () => {
      updateCartCount();
    };

    window.addEventListener("storage", handleStorageChange);

    return () => {
      window.removeEventListener("storage", handleStorageChange);
    };
```



```tsx
import { client } from "@/sanity/lib/client";
import Link from "next/link";

async function GetData(searchTerm: string | null = null) {
  const query = `
    *[_type == "product" ${
      searchTerm ? `&& productName match "${searchTerm}*"` : ""
    }]{
      _id,
      productName,
      description,
      price,
      "image": image.asset->url,
      category,
      inventory,
      colors,
      status
    }
  `;
  return await client.fetch(query);
}

export default async function Products({ searchParams }: { searchParams: { search?: string } }) {
  const searchTerm = searchParams?.search || null;
  const data = await GetData(searchTerm);

  return (
    <div className="max-w-7xl mx-auto px-4 py-8">
      <div className="flex-1 grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
        {data.map((product: any) => (
          <Link href={`/product/${product._id}`} key={product._id}>
            <div className="bg-white shadow-md rounded-lg p-4 transition-shadow">
```

1. Steps taken to build and integrate pages:

- **Step 1:** First, we created the main pages like **Product Page**, **Cart Page**, and **ProductDetail Page** inside the **App folder**.
  - Inside each folder, we created a `page.tsx` file for the respective page.
- **Step 2:** Each page was built to handle a specific task. For example:
  - **ProductPage** displays a list of products.
  - **CartPage** displays the products added to the cart.
  - **ProductDetailPage** shows detailed information about a single product when the user clicks on it.
- **Step 3:** After creating the pages, we used them to create the main flow of the application (like viewing products, adding them to the cart, and viewing the cart).
- **Step 4:** We connected the **CartPage** with **localStorage** to store cart data, so the cart persists even after page reloads.
- **Step 5:** When a user adds a product to the cart, it gets saved to **localStorage**, and the **CartPage** will display the updated cart data.

2. Challenges faced and solutions implemented:

- **Challenge 1: Displaying dynamic data on the CartPage** (like product images, names, prices, and quantities).
  - **Solution:** We fetched the product data using **Sanity** (or other APIs) and passed it to the **CartPage**. This allowed us to dynamically update the cart with images, prices, and other details based on what the user added.
- **Challenge 2: Persisting cart data after a page reload**.
  - **Solution:** We used **localStorage** to store the cart items. This way, even if the user reloads the page, the items in the cart will still be available.
- **Challenge 3: Handling quantity increase and decrease** in the cart.
  - **Solution:** We created functions in the **CartPage** to handle increasing or decreasing the quantity of items in the cart and recalculating the total price. When the user clicks on "+" or "-", the quantity updates, and the cart re-renders with the updated price.

3.Best practices followed during development:

- **Separation of Concerns:** Each page focuses on a specific task. For example:
  - **ProductPage** is only responsible for showing products.
  - **CartPage** manages cart items and checkout.
  - **ProductDetailPage** is responsible for displaying detailed product information.
- **State Management:** We used **React's useState** and **useEffect** hooks to manage the state of the cart on the **CartPage** and ensure the data updates correctly when the user changes quantities or adds/removes items.
- **Clean and Simple Code:** We followed a clean coding approach by using clear and descriptive names for pages and variables. This makes it easier to maintain and scale the app.
- **Responsive Design:** We used **Tailwind CSS** for styling, ensuring that the app is responsive on all screen sizes. This provides a smooth experience whether on desktop or mobile.