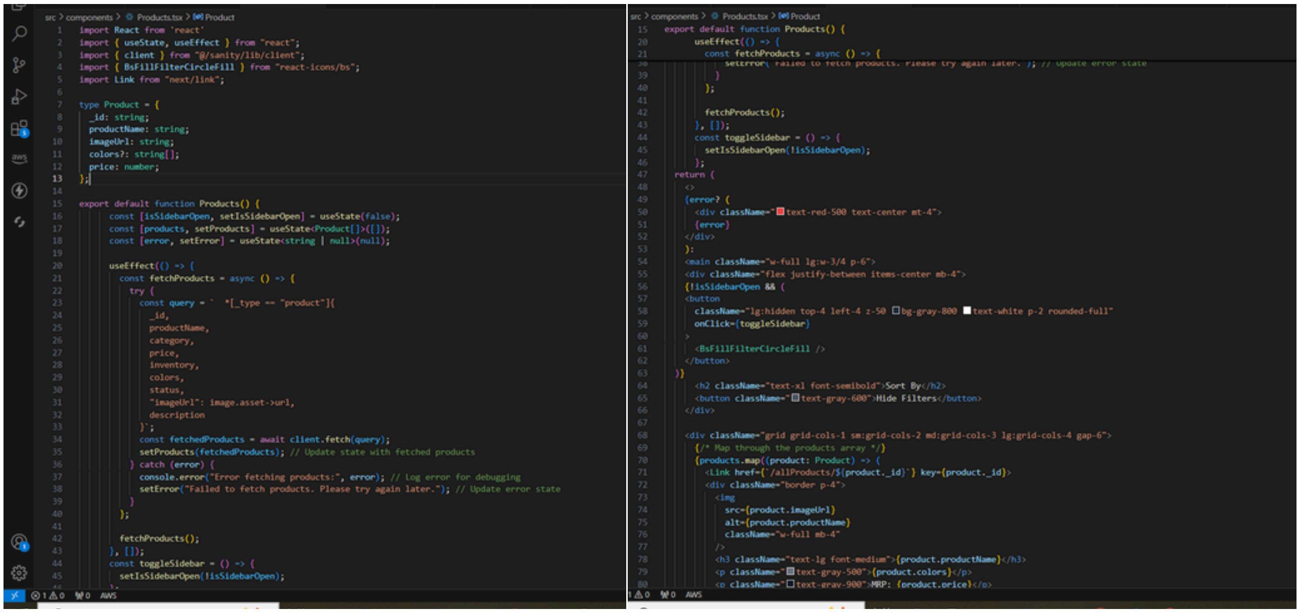
Day 04: Building Dynamic Frontend Components For Marketplace

Key Components to Build:

Product Listing Component:

Code Snippet:



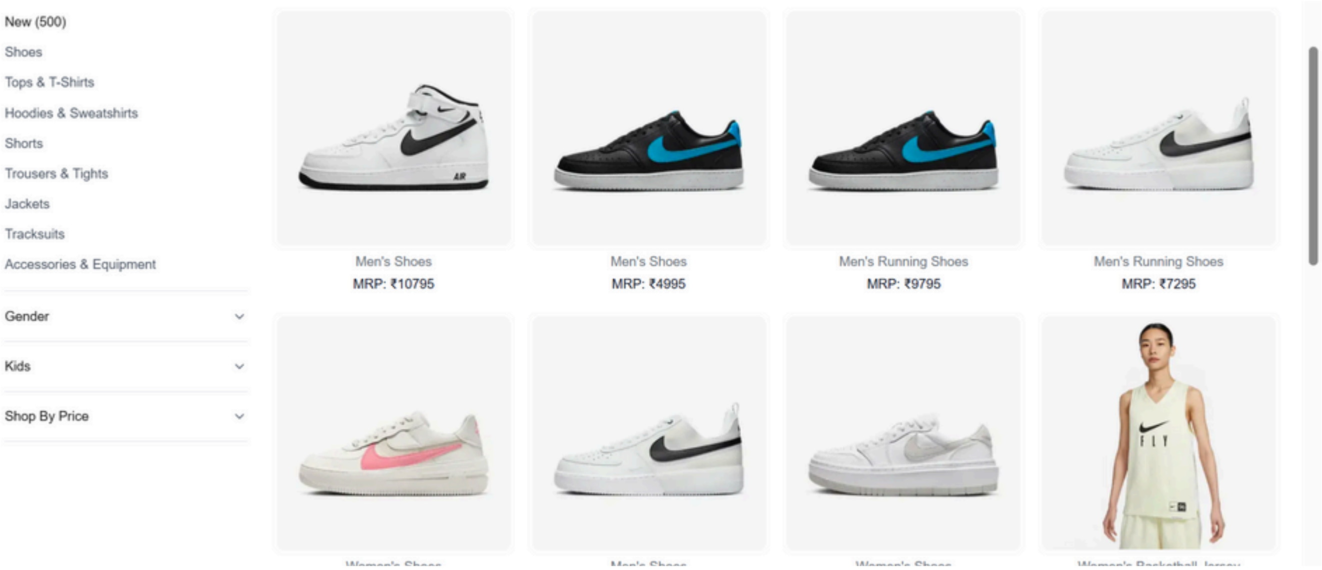
src > components > Products.tsx > Product

```tsx
import React from 'react'
import { useState, useEffect } from "react";
import { client } from "@/sanity/lib/client";
import { BsFillFilterCircleFill } from "react-icons/bs";
import Link from "next/link";

type Product = {
  _id: string;
  productName: string;
  imageUrl: string;
  colors?: string[];
  price: number;
};

export default function Products() {
      const [isSidebarOpen, setIsSidebarOpen] = useState(false);
      const [products, setProducts] = useState<Product[]>([]);
      const [error, setError] = useState<string | null>(null);

      useEffect(() => {
        const fetchProducts = async () => {
          try {
            const query = `  *[_type == "product"]{
              _id,
              productName,
              category,
              price,
              inventory,
              colors,
              status,
              "imageUrl": image.asset->url,
              description
            }`;
            const fetchedProducts = await client.fetch(query);
            setProducts(fetchedProducts); // Update state with fetched products
          } catch (error) {
            console.error("Error fetching products:", error); // Log error for debugging
            setError("Failed to fetch products. Please try again later."); // Update error state
          }
        };

        fetchProducts();
      }, []);
      const toggleSidebar = () => {
        setIsSidebarOpen(!isSidebarOpen);
```

src > components > Products.tsx > Product

```tsx
export default function Products() {
      useEffect(() => {
        const fetchProducts = async () => {
            setError("Failed to fetch products. Please try again later."); // Update error state
          }
        };

        fetchProducts();
      }, []);
      const toggleSidebar = () => {
        setIsSidebarOpen(!isSidebarOpen);
      };
    return (
      <>
      {error? (
        <div className="text-red-500 text-center mt-4">
        {error}
      </div>
      ):
      <main className="w-full lg:w-3/4 p-6">
      <div className="flex justify-between items-center mb-4">
      {!isSidebarOpen && (
      <button
        className="lg:hidden top-4 left-4 z-50 bg-gray-800 text-white p-2 rounded-full"
        onClick={toggleSidebar}
      >
        <BsFillFilterCircleFill />
      </button>
    )}
        <h2 className="text-xl font-semibold">Sort By</h2>
        <button className="text-gray-600">Hide Filters</button>
      </div>

      <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
        {/* Map through the products array */}
        {products.map((product: Product) => (
          <Link href={`/allProducts/${product._id}`} key={product._id}>
          <div className="border p-4">
            <img
              src={product.imageUrl}
              alt={product.productName}
              className="w-full mb-4"
            />
            <h3 className="text-lg font-medium">{product.productName}</h3>
            <p className="text-gray-500">{product.colors}</p>
            <p className="text-gray-900">MRP: {product.price}</p>
```

Output Snippet:



Men's Shoes
MRP: ₹10795

Men's Shoes
MRP: ₹4995

Men's Running Shoes
MRP: ₹9795

Men's Running Shoes
MRP: ₹7295

Women's Shoes

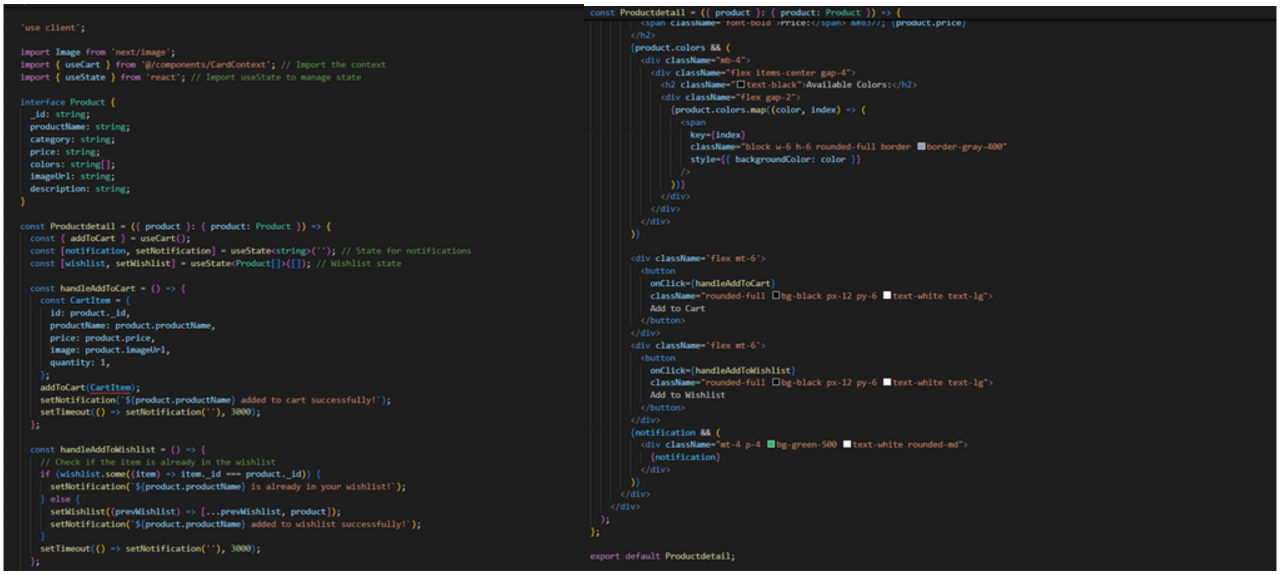
Men's Shoes

Women's Shoes

Women's Basketball Jersey

Product Detail Component:

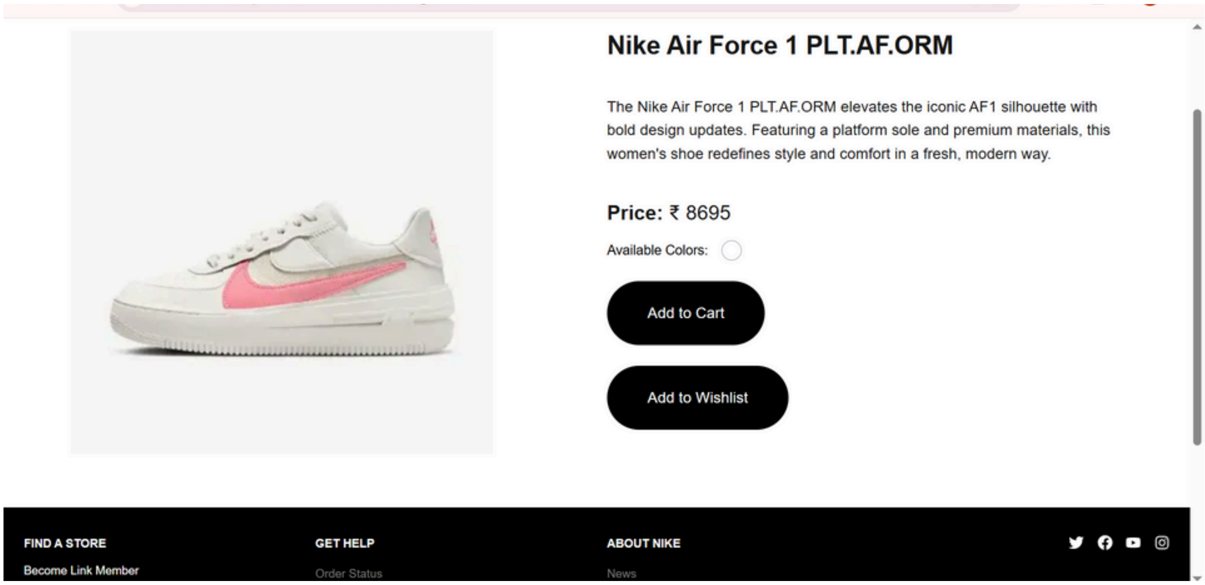
Code Snippet:



```tsx
'use client';

import Image from 'next/image';
import { useCart } from '@/components/CardContext'; // Import the context
import { useState } from 'react'; // Import useState to manage state

interface Product {
  _id: string;
  productName: string;
  category: string;
  price: string;
  colors: string[];
  imageUrl: string;
  description: string;
}

const Productdetail = ({ product }: { product: Product }) => {
  const { addToCart } = useCart();
  const [notification, setNotification] = useState<string>(''); // State for notifications
  const [wishlist, setWishlist] = useState<Product[]>([]); // Wishlist state

  const handleAddToCart = () => {
    const CartItem = {
      id: product._id,
      productName: product.productName,
      price: product.price,
      image: product.imageUrl,
      quantity: 1,
    };
    addToCart(CartItem);
    setNotification(`${product.productName} added to cart successfully!`);
    setTimeout(() => setNotification(''), 3000);
  };

  const handleAddToWishlist = () => {
    // Check if the item is already in the wishlist
    if (wishlist.some((item) => item._id === product._id)) {
      setNotification(`${product.productName} is already in your wishlist!`);
    } else {
      setWishlist((prevWishlist) => [...prevWishlist, product]);
      setNotification(`${product.productName} added to wishlist successfully!`);
    }
    setTimeout(() => setNotification(''), 3000);
  };
```

```tsx
const Productdetail = ({ product }: { product: Product }) => {
          <span className="font-bold">Price:</span> &#8377; {product.price}
        </h2>
        {product.colors && (
          <div className="mb-4">
            <div className="flex items-center gap-4">
              <h2 className="text-black">Available Colors:</h2>
              <div className="flex gap-2">
                {product.colors.map((color, index) => (
                  <span
                    key={index}
                    className="block w-6 h-6 rounded-full border border-gray-400"
                    style={{ backgroundColor: color }}
                  />
                ))}
              </div>
            </div>
          </div>
        )}

        <div className='flex mt-6'>
          <button
            onClick={handleAddToCart}
            className="rounded-full bg-black px-12 py-6 text-white text-lg">
            Add to Cart
          </button>
        </div>
        <div className='flex mt-6'>
          <button
            onClick={handleAddToWishlist}
            className="rounded-full bg-black px-12 py-6 text-white text-lg">
            Add to Wishlist
          </button>
        </div>
        {notification && (
          <div className="mt-4 p-4 bg-green-500 text-white rounded-md">
            {notification}
          </div>
        )}
      </div>
    </div>
  );
};

export default Productdetail;
```

Output Snippet:



Nike Air Force 1 PLT.AF.ORM

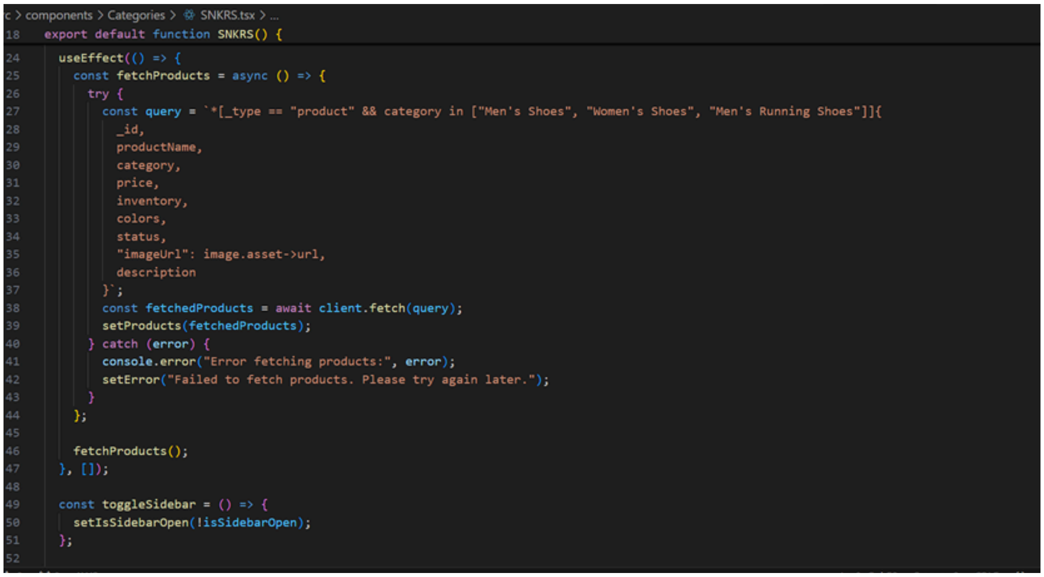
The Nike Air Force 1 PLT.AF.ORM elevates the iconic AF1 silhouette with bold design updates. Featuring a platform sole and premium materials, this women's shoe redefines style and comfort in a fresh, modern way.

Price: ₹ 8695

Available Colors: ○

Category Component:

Code Snippet:

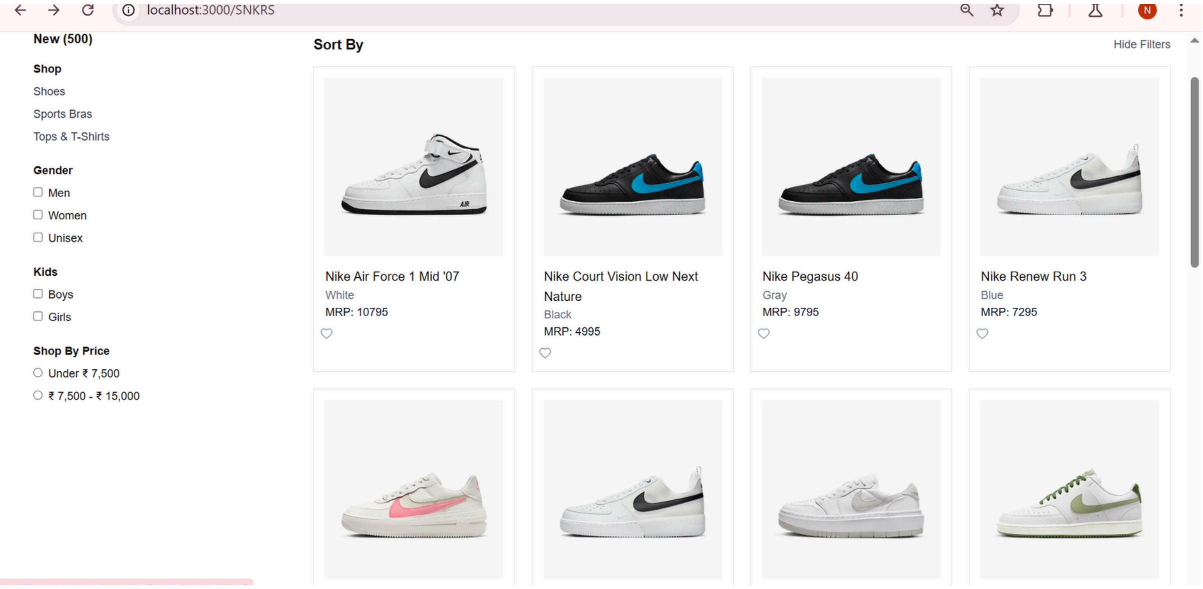


c > components > Categories > SNKRS.tsx > ...

```
export default function SNKRS() {
  useEffect(() => {
    const fetchProducts = async () => {
      try {
        const query = `*[_type == "product" && category in ["Men's Shoes", "Women's Shoes", "Men's Running Shoes"]]{
          _id,
          productName,
          category,
          price,
          inventory,
          colors,
          status,
          "imageUrl": image.asset->url,
          description
        }`;
        const fetchedProducts = await client.fetch(query);
        setProducts(fetchedProducts);
      } catch (error) {
        console.error("Error fetching products:", error);
        setError("Failed to fetch products. Please try again later.");
      }
    };

    fetchProducts();
  }, []);

  const toggleSidebar = () => {
    setIsSidebarOpen(!isSidebarOpen);
  };
```

Output Snippet:

localhost:3000/SNKRS



localhost:3000/SNKRS

New (500)

Shop

Shoes

Sports Bras

Tops & T-Shirts

Gender

□ Men

□ Women

□ Unisex

Kids

□ Boys

□ Girls

Shop By Price

○ Under ₹ 7,500

○ ₹ 7,500 - ₹ 15,000

Sort By

Hide Filters

Nike Air Force 1 Mid '07
White
MRP: 10795

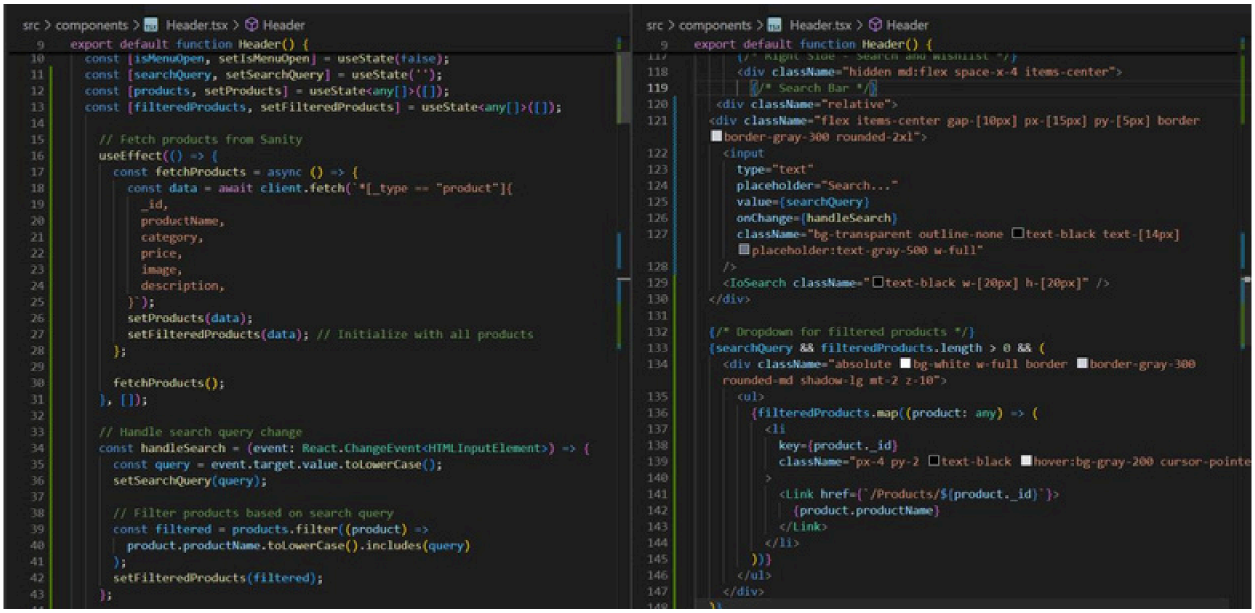
Nike Court Vision Low Next Nature
Black
MRP: 4995

Nike Pegasus 40
Gray
MRP: 9795

Nike Renew Run 3
Blue
MRP: 7295

Search Bar:

Code Snippet:



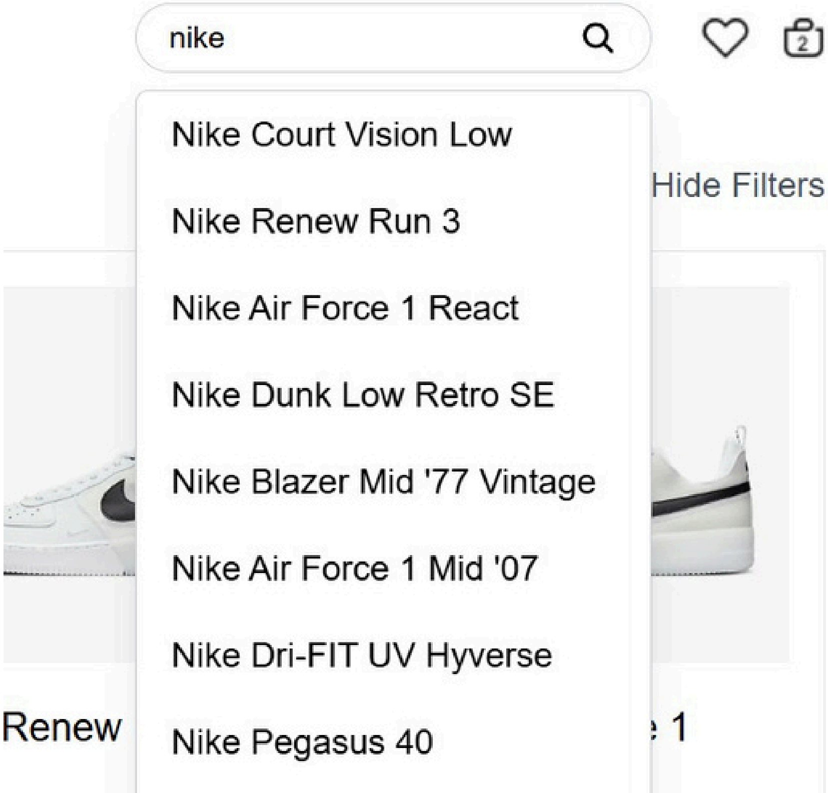
src > components > Header.tsx > Header

```
export default function Header() {
  const [isMenuOpen, setIsMenuOpen] = useState(false);
  const [searchQuery, setSearchQuery] = useState('');
  const [products, setProducts] = useState<any[]>([]);
  const [filteredProducts, setFilteredProducts] = useState<any[]>([]);

  // Fetch products from Sanity
  useEffect(() => {
    const fetchProducts = async () => {
      const data = await client.fetch(`*[_type == "product"]{
        _id,
        productName,
        category,
        price,
        image,
        description,
      }`);
      setProducts(data);
      setFilteredProducts(data); // Initialize with all products
    };

    fetchProducts();
  }, []);

  // Handle search query change
  const handleSearch = (event: React.ChangeEvent<HTMLInputElement>) => {
    const query = event.target.value.toLowerCase();
    setSearchQuery(query);

    // Filter products based on search query
    const filtered = products.filter((product) =>
      product.productName.toLowerCase().includes(query)
    );
    setFilteredProducts(filtered);
  };
```

src > components > Header.tsx > Header

```
export default function Header() {
        {/* Right Side - Search and Wishlist */}
        <div className="hidden md:flex space-x-4 items-center">
          {/* Search Bar */}
      <div className="relative">
     <div className="flex items-center gap-[10px] px-[15px] py-[5px] border
     □border-gray-300 rounded-2xl">
       <input
         type="text"
         placeholder="Search..."
         value={searchQuery}
         onChange={handleSearch}
         className="bg-transparent outline-none ■text-black text-[14px]
         ▨placeholder:text-gray-500 w-full"
       />
       <IoSearch className="■text-black w-[20px] h-[20px]" />
     </div>

     {/* Dropdown for filtered products */}
     {searchQuery && filteredProducts.length > 0 && (
       <div className="absolute □bg-white w-full border □border-gray-300
       rounded-md shadow-lg mt-2 z-10">
         <ul>
           {filteredProducts.map((product: any) => (
             <li
               key={product._id}
               className="px-4 py-2 ■text-black □hover:bg-gray-200 cursor-pointe
             >
               <Link href={`/Products/${product._id}`}>
                 {product.productName}
               </Link>
             </li>
           ))}
         </ul>
       </div>
     )}
```

Output Snippet:



Nike Court Vision Low

Nike Renew Run 3

Nike Air Force 1 React

Nike Dunk Low Retro SE

Nike Blazer Mid '77 Vintage

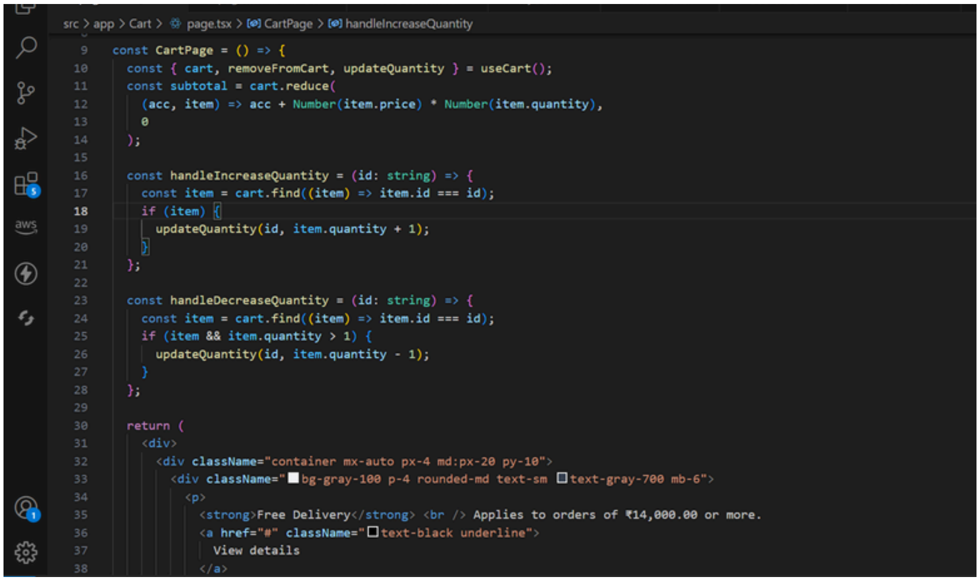
Nike Air Force 1 Mid '07

Nike Dri-FIT UV Hyverse

Nike Pegasus 40

Cart Component:

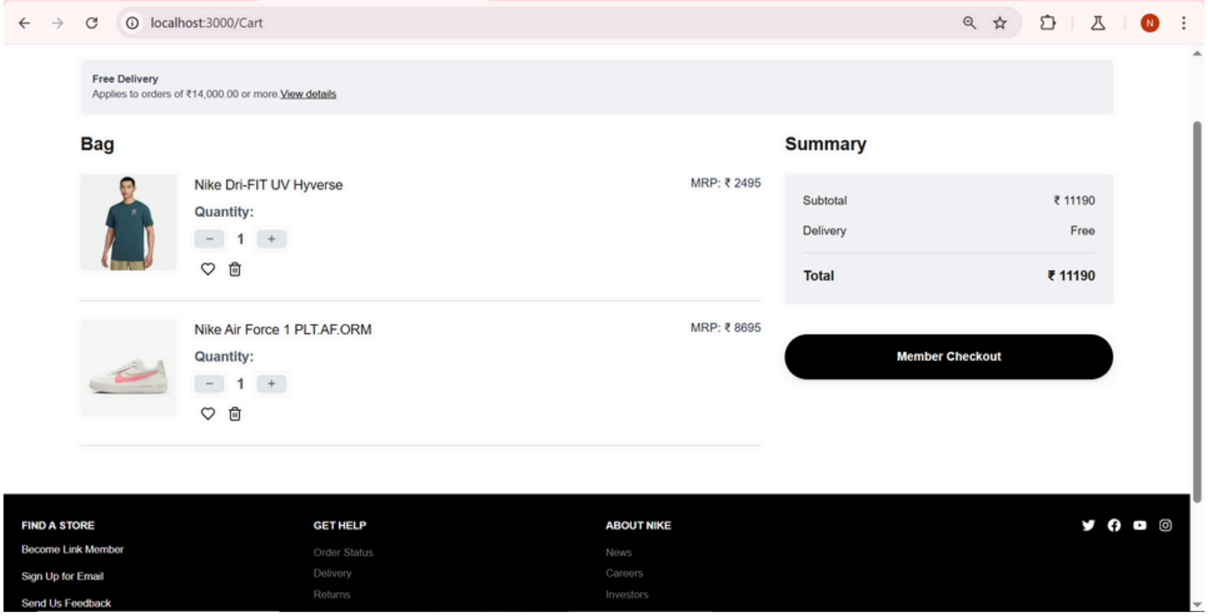
Code Snippet:



src > app > Cart > page.tsx > CartPage > handleIncreaseQuantity

```
const CartPage = () => {
  const { cart, removeFromCart, updateQuantity } = useCart();
  const subtotal = cart.reduce(
    (acc, item) => acc + Number(item.price) * Number(item.quantity),
    0
  );

  const handleIncreaseQuantity = (id: string) => {
    const item = cart.find((item) => item.id === id);
    if (item) {
      updateQuantity(id, item.quantity + 1);
    }
  };

  const handleDecreaseQuantity = (id: string) => {
    const item = cart.find((item) => item.id === id);
    if (item && item.quantity > 1) {
      updateQuantity(id, item.quantity - 1);
    }
  };

  return (
    <div>
      <div className="container mx-auto px-4 md:px-20 py-10">
        <div className="bg-gray-100 p-4 rounded-md text-sm text-gray-700 mb-6">
          <p>
            <strong>Free Delivery</strong> <br /> Applies to orders of ₹14,000.00 or more.
            <a href="#" className="text-black underline">
              View details
            </a>
```

Output Snippet:



localhost:3000/Cart

Free Delivery
Applies to orders of ₹14,000.00 or more View details

Bag

Nike Dri-FIT UV Hyverse — MRP: ₹ 2495
Quantity: 1

Nike Air Force 1 PLT.AF.ORM — MRP: ₹ 8695
Quantity: 1

Summary

Subtotal ₹ 11190
Delivery Free
Total ₹ 11190

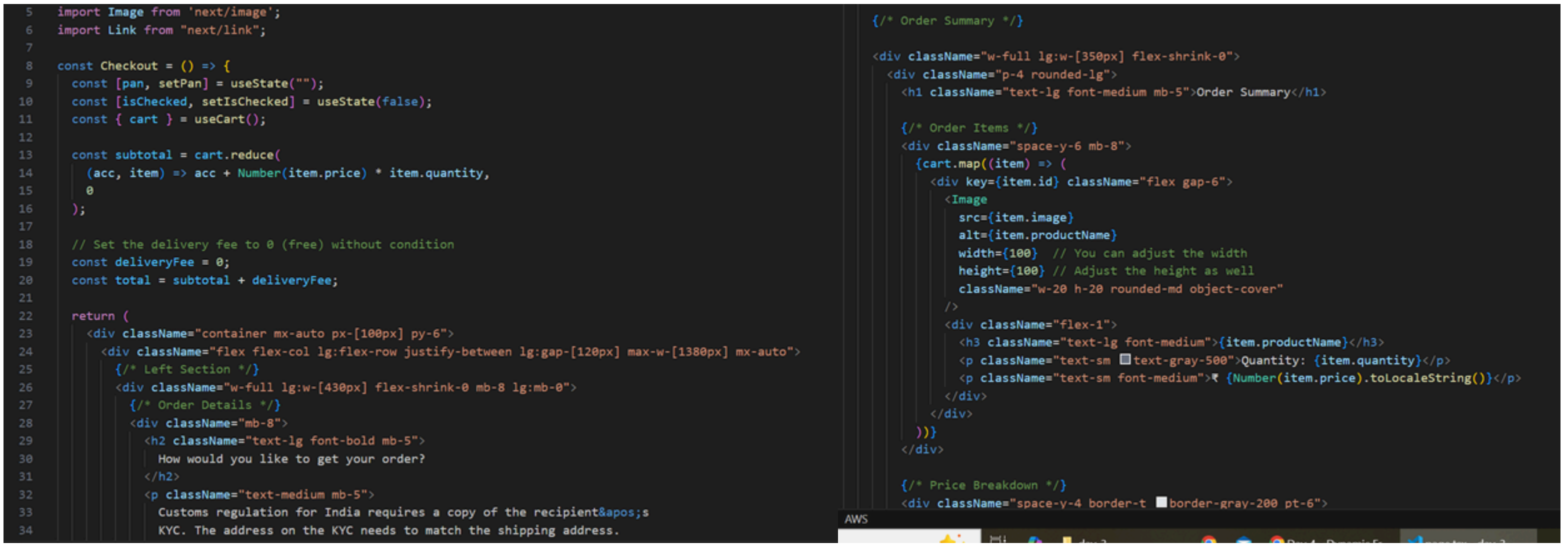
FIND A STORE
Become Link Member
Sign Up for Email
Send Us Feedback

GET HELP
Order Status
Delivery
Returns

ABOUT NIKE
News
Careers
Investors

Checkout Flow Component:

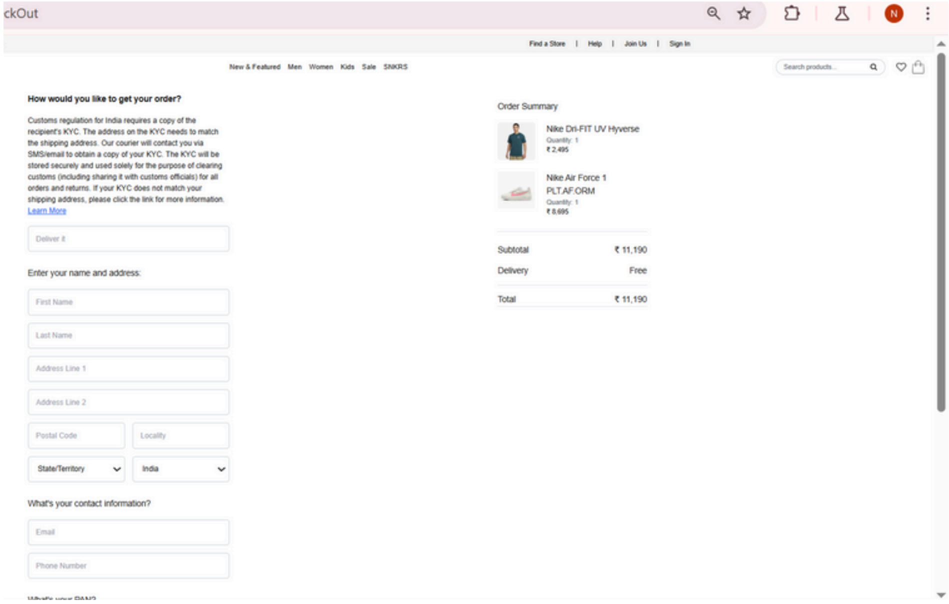
Code Snippet:



```
import Image from 'next/image';
import Link from "next/link";

const Checkout = () => {
  const [pan, setPan] = useState("");
  const [isChecked, setIsChecked] = useState(false);
  const { cart } = useCart();

  const subtotal = cart.reduce(
    (acc, item) => acc + Number(item.price) * item.quantity,
    0
  );

  // Set the delivery fee to 0 (free) without condition
  const deliveryFee = 0;
  const total = subtotal + deliveryFee;

  return (
    <div className="container mx-auto px-[100px] py-6">
      <div className="flex flex-col lg:flex-row justify-between lg:gap-[120px] max-w-[1380px] mx-auto">
        {/* Left Section */}
        <div className="w-full lg:w-[430px] flex-shrink-0 mb-8 lg:mb-0">
          {/* Order Details */}
          <div className="mb-8">
            <h2 className="text-lg font-bold mb-5">
              How would you like to get your order?
            </h2>
            <p className="text-medium mb-5">
              Customs regulation for India requires a copy of the recipient&apos;s
              KYC. The address on the KYC needs to match the shipping address.
```

```
        {/* Order Summary */}

        <div className="w-full lg:w-[350px] flex-shrink-0">
          <div className="p-4 rounded-lg">
            <h1 className="text-lg font-medium mb-5">Order Summary</h1>

            {/* Order Items */}
            <div className="space-y-6 mb-8">
              {cart.map((item) => (
                <div key={item.id} className="flex gap-6">
                  <Image
                    src={item.image}
                    alt={item.productName}
                    width={100}  // You can adjust the width
                    height={100} // Adjust the height as well
                    className="w-20 h-20 rounded-md object-cover"
                  />
                  <div className="flex-1">
                    <h3 className="text-lg font-medium">{item.productName}</h3>
                    <p className="text-sm text-gray-500">Quantity: {item.quantity}</p>
                    <p className="text-sm font-medium">₹ {Number(item.price).toLocaleString()}</p>
                  </div>
                </div>
              ))}
            </div>

            {/* Price Breakdown */}
            <div className="space-y-4 border-t border-gray-200 pt-6">
```

Output Snippet:



ckOut
Find a Store | Help | Join Us | Sign In
New & Featured Men Women Kids Sale SNKRS
Search products...

How would you like to get your order?

Customs regulation for India requires 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

Deliver it

Enter your name and address:

First Name

Last Name

Address Line 1

Address Line 2

Postal Code
Locality

State/Territory
India

What's your contact information?

Email

Phone Number

What's your PAN?

Order Summary

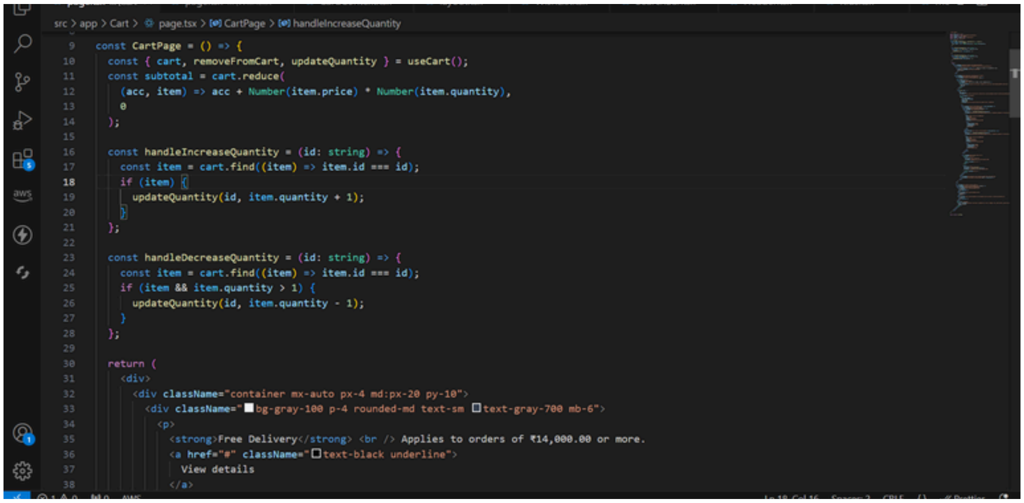
Nike Dri-FIT UV Hyverse
Quantity: 1
₹ 2,495

Nike Air Force 1 PLT.AF.ORM
Quantity: 1
₹ 8,695

Subtotal	₹ 11,190
Delivery	Free
Total	₹ 11,190

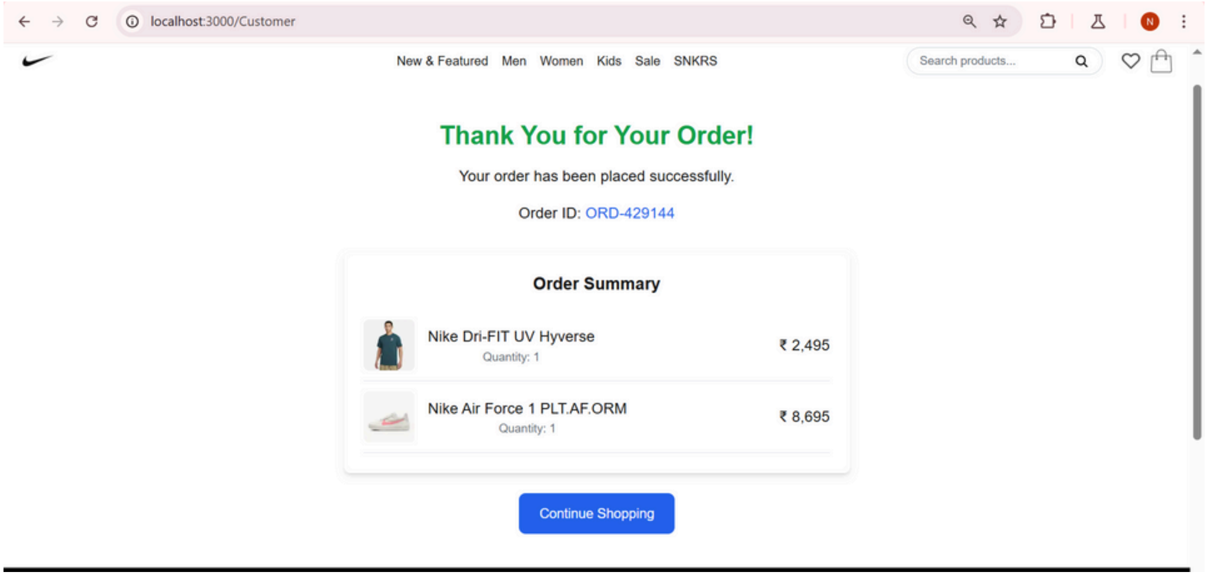
Customer Thank you Card

Code Snippet:


src > app > Cart > page.tsx > CartPage > handleIncreaseQuantity

```
const CartPage = () => {
  const { cart, removeFromCart, updateQuantity } = useCart();
  const subtotal = cart.reduce(
    (acc, item) => acc + Number(item.price) * Number(item.quantity),
    0
  );

  const handleIncreaseQuantity = (id: string) => {
    const item = cart.find((item) => item.id === id);
    if (item) {
      updateQuantity(id, item.quantity + 1);
    }
  };

  const handleDecreaseQuantity = (id: string) => {
    const item = cart.find((item) => item.id === id);
    if (item && item.quantity > 1) {
      updateQuantity(id, item.quantity - 1);
    }
  };

  return (
    <div>
      <div className="container mx-auto px-4 md:px-20 py-10">
        <div className="bg-gray-100 p-4 rounded-md text-sm text-gray-700 mb-6">
          <p>
            <strong>Free Delivery</strong> <br /> Applies to orders of ₹14,000.00 or more.
            <a href="#" className="text-black underline">
              View details
            </a>
```

Output Snippet:



Thank You for Your Order!

Your order has been placed successfully.

Order ID: ORD-429144

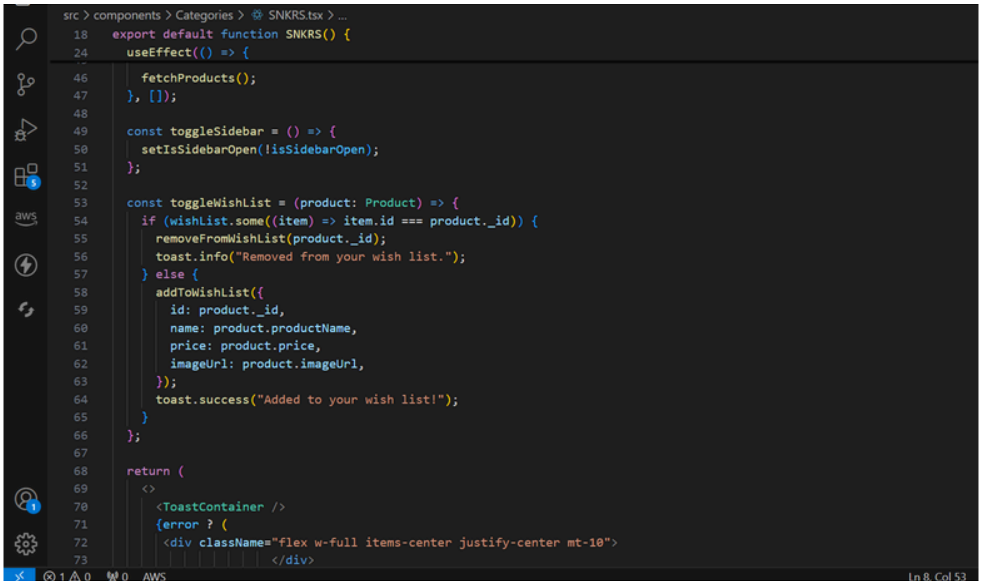
Order Summary

Nike Dri-FIT UV Hyverse Quantity: 1	₹ 2,495
Nike Air Force 1 PLT.AF.ORM Quantity: 1	₹ 8,695

Continue Shopping

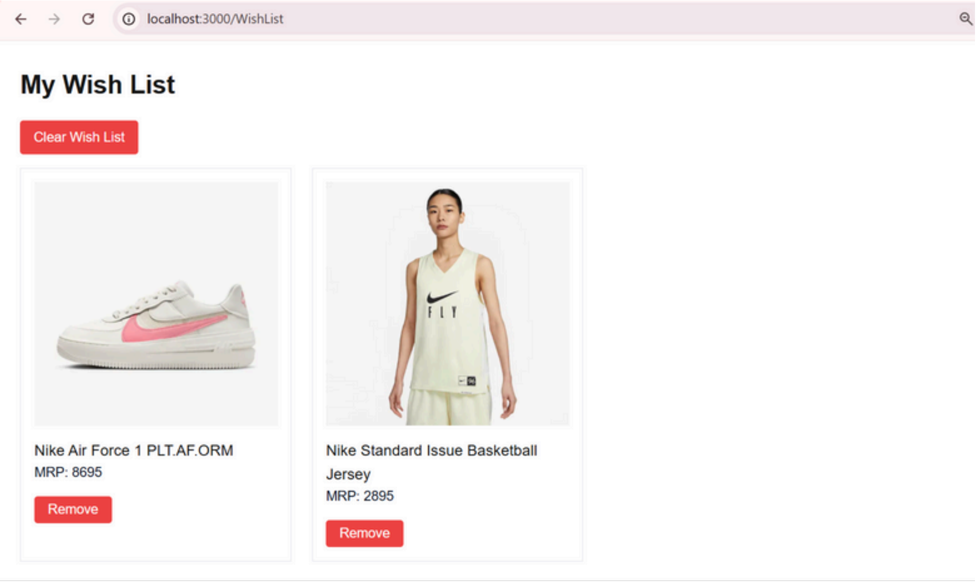
wishlist

Code Snippet:



```
export default function SNKRS() {
  useEffect(() => {
    fetchProducts();
  }, []);

  const toggleSidebar = () => {
    setIsSidebarOpen(!isSidebarOpen);
  };

  const toggleWishList = (product: Product) => {
    if (wishList.some((item) => item.id === product._id)) {
      removeFromWishList(product._id);
      toast.info("Removed from your wish list.");
    } else {
      addToWishList({
        id: product._id,
        name: product.productName,
        price: product.price,
        imageUrl: product.imageUrl,
      });
      toast.success("Added to your wish list!");
    }
  };

  return (
    <>
      <ToastContainer />
      {error ? (
       <div className="flex w-full items-center justify-center mt-10">
                  </div>
```

Output Snippet:



My Wish List

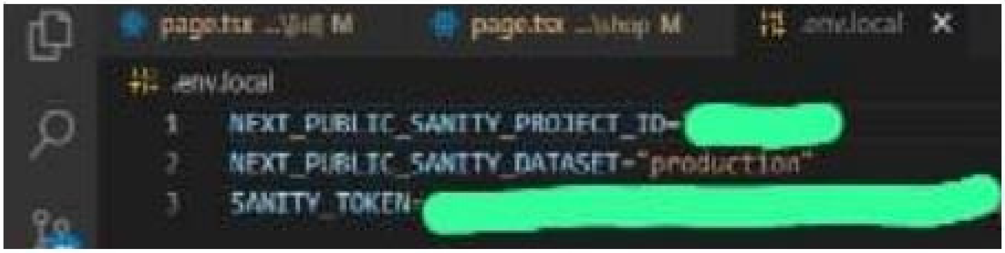
Nike Air Force 1 PLT.AF.ORM
MRP: 8695

Nike Standard Issue Basketball Jersey
MRP: 2895

API Integration:

1. Sanity API Fetching Logic: Since you're using Sanity CMS, you need to ensure data fetching is properly managed.

- Sanity Client Setup (if not already set up): In your lib folder, configure the Sanity client:



```
NEXT_PUBLIC_SANITY_PROJECT_ID=[redacted]
NEXT_PUBLIC_SANITY_DATASET="production"
SANITY_TOKEN=[redacted]
```

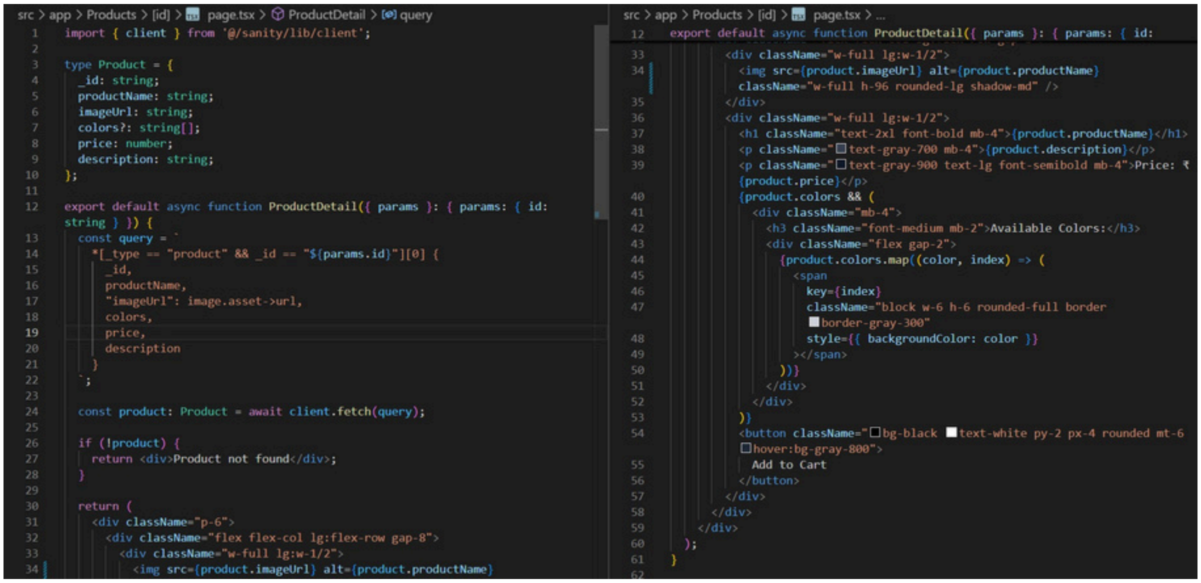
Dynamic Routing:

Dynamic routing is used to handle routes like /Products/:id for individual product pages.

1. Create Dynamic Route File:

In the pages directory, create a [id] file inside the shop folder:

pages/Products/[id]



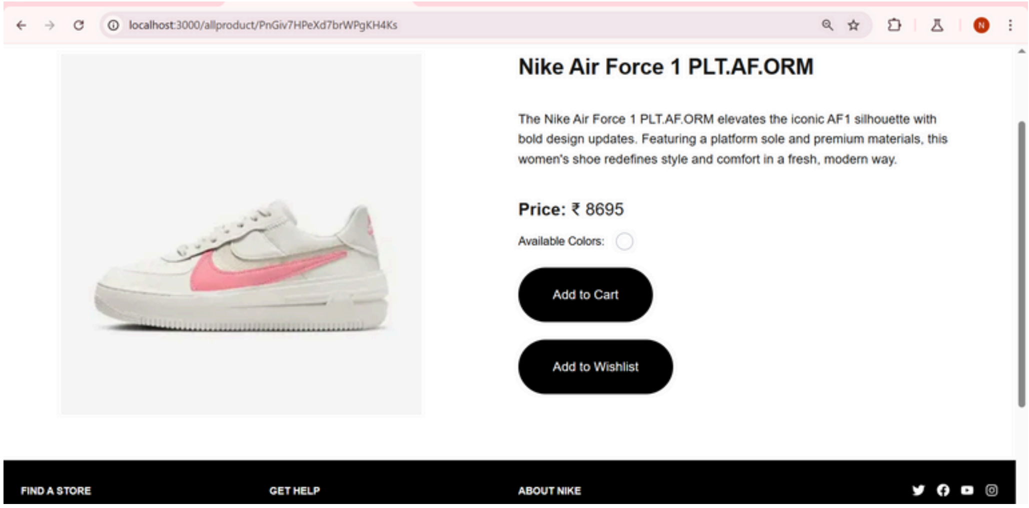
src > app > Products > [id] > page.tsx > ProductDetail > query

```tsx
import { client } from '@/sanity/lib/client';

type Product = {
  _id: string;
  productName: string;
  imageUrl: string;
  colors?: string[];
  price: number;
  description: string;
};

export default async function ProductDetail({ params }: { params: { id:
string } }) {
  const query = `
    *[_type == "product" && _id == "${params.id}"][0] {
      _id,
      productName,
      "imageUrl": image.asset->url,
      colors,
      price,
      description
    }
  `;

  const product: Product = await client.fetch(query);

  if (!product) {
    return <div>Product not found</div>;
  }

  return (
    <div className="p-6">
      <div className="flex flex-col lg:flex-row gap-8">
        <div className="w-full lg:w-1/2">
          <img src={product.imageUrl} alt={product.productName}
```

src > app > Products > [id] > page.tsx > ...

```tsx
export default async function ProductDetail({ params }: { params: { id:
        <div className="w-full lg:w-1/2">
          <img src={product.imageUrl} alt={product.productName}
          className="w-full h-96 rounded-lg shadow-md" />
        </div>
        <div className="w-full lg:w-1/2">
          <h1 className="text-2xl font-bold mb-4">{product.productName}</h1>
          <p className="text-gray-700 mb-4">{product.description}</p>
          <p className="text-gray-900 text-lg font-semibold mb-4">Price: ₹
          {product.price}</p>
          {product.colors && (
            <div className="mb-4">
              <h3 className="font-medium mb-2">Available Colors:</h3>
              <div className="flex gap-2">
                {product.colors.map((color, index) => (
                  <span
                    key={index}
                    className="block w-6 h-6 rounded-full border
                    border-gray-300"
                    style={{ backgroundColor: color }}
                  ></span>
                ))}
              </div>
            </div>
          )}
          <button className="bg-black text-white py-2 px-4 rounded mt-6
          hover:bg-gray-800">
            Add to Cart
          </button>
        </div>
      </div>
    </div>
  );
}
```

2. Get Product Data Dynamically:



localhost:3000/allproduct/PnGiv7HPeXd7brWPgKH4Ks

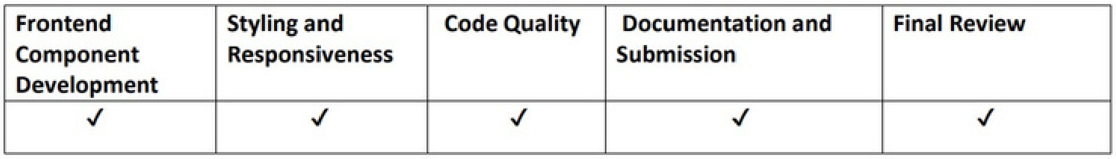
Nike Air Force 1 PLT.AF.ORM

The Nike Air Force 1 PLT.AF.ORM elevates the iconic AF1 silhouette with bold design updates. Featuring a platform sole and premium materials, this women's shoe redefines style and comfort in a fresh, modern way.

Price: ₹ 8695

Available Colors: ○

Final CheckList



Frontend Component Development	Styling and Responsiveness	Code Quality	Documentation and Submission	Final Review
✓	✓	✓	✓	✓