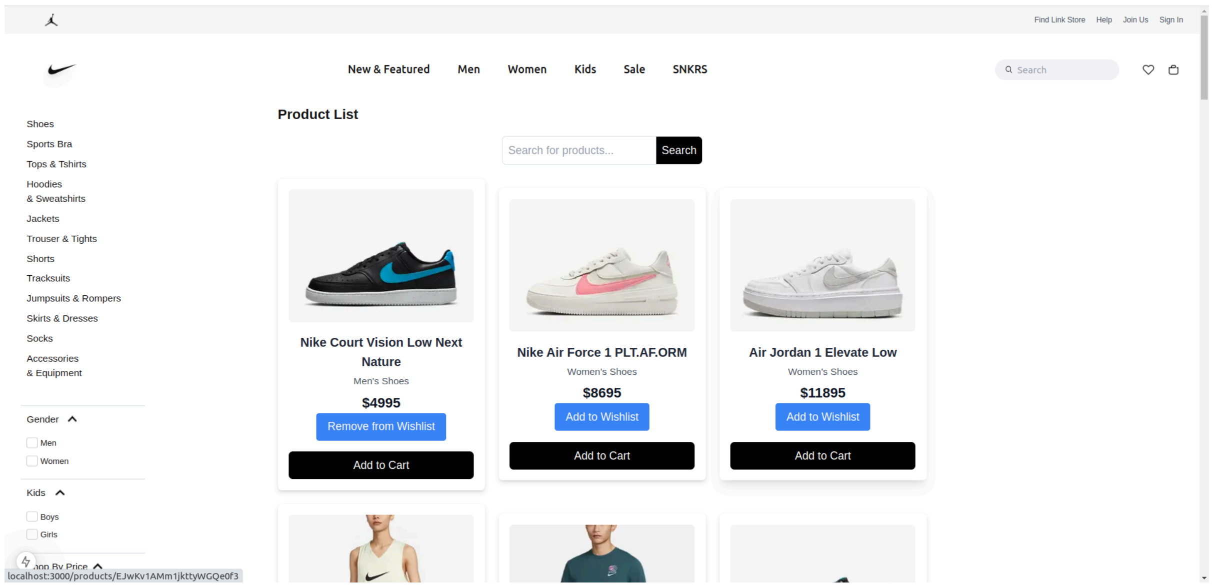
Day 4 - Dynamic Frontend Components - [Nike Website]

Functional Deliverables

Screenshots

1. Product Listing Page with Dynamic Data:

Showcasing a grid of products fetched dynamically from Sanity CMS, including functional filters, search bar, and pagination.



Product List

Nike Court Vision Low Next Nature
Men's Shoes
$4995

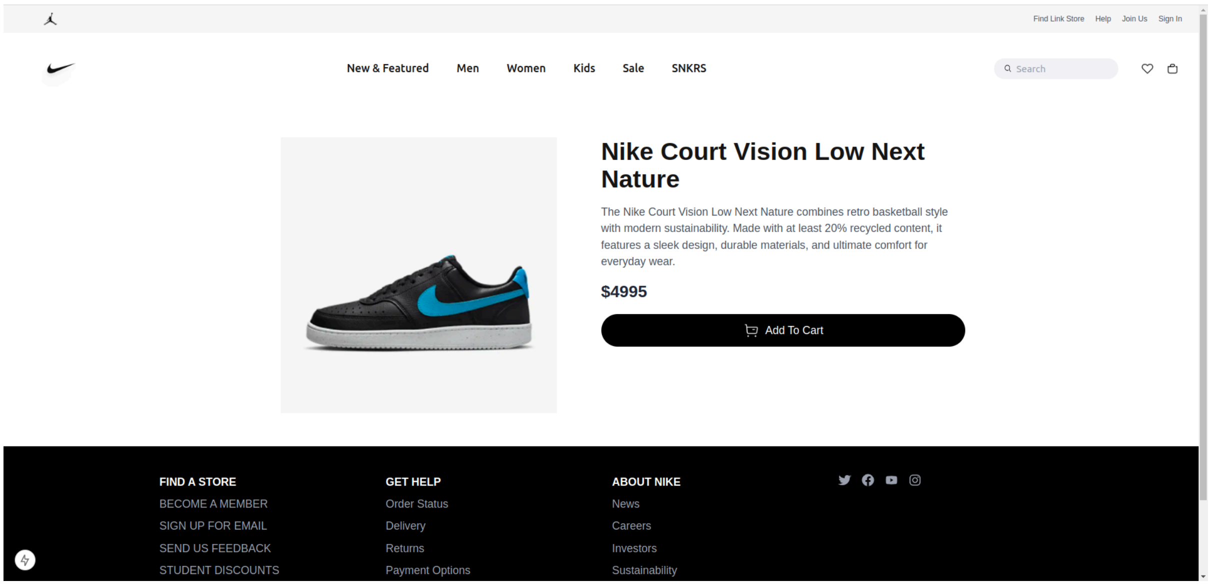
Nike Air Force 1 PLT.AF.ORM
Women's Shoes
$8695

Air Jordan 1 Elevate Low
Women's Shoes
$11895

localhost:3000/products/EJwKv1AMm1jkttyWGQe0f3

2.Individual Product Detail Pages:

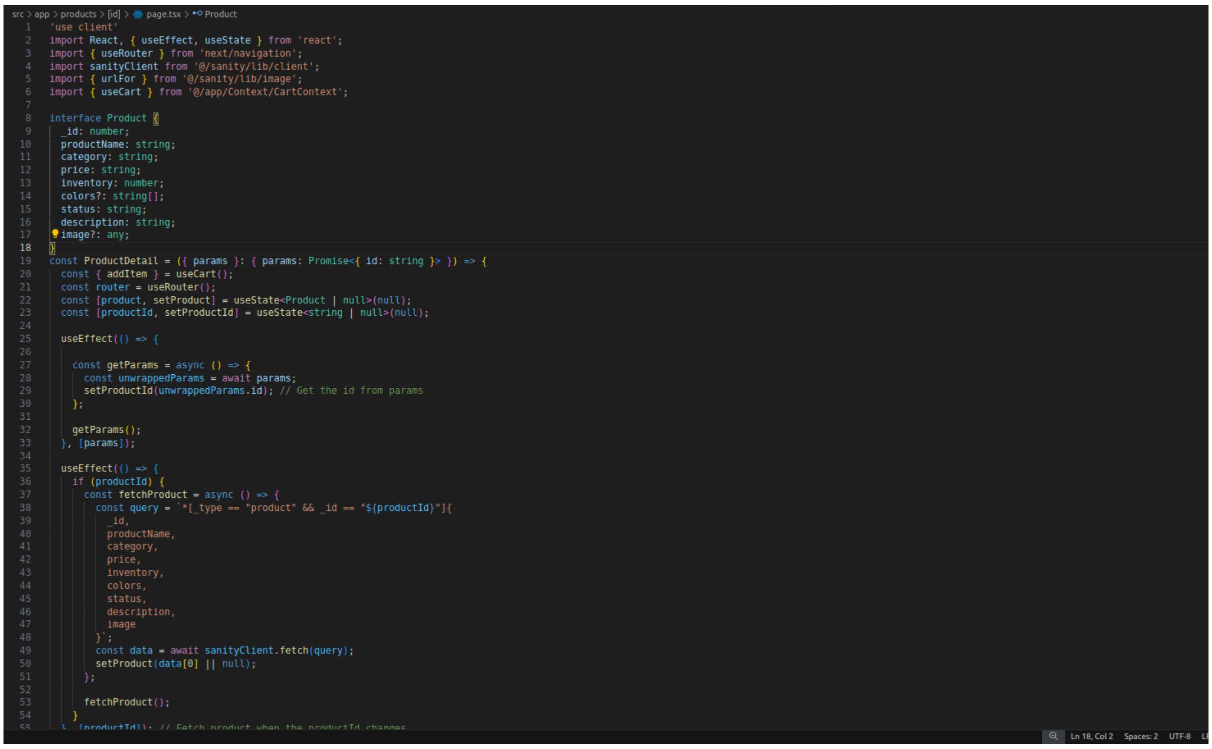
Implement accurate dynamic routing for each product, ensuring proper rendering of product details such as name, price, description, and image.



Nike Court Vision Low Next Nature

The Nike Court Vision Low Next Nature combines retro basketball style with modern sustainability. Made with at least 20% recycled content, it features a sleek design, durable materials, and ultimate comfort for everyday wear.

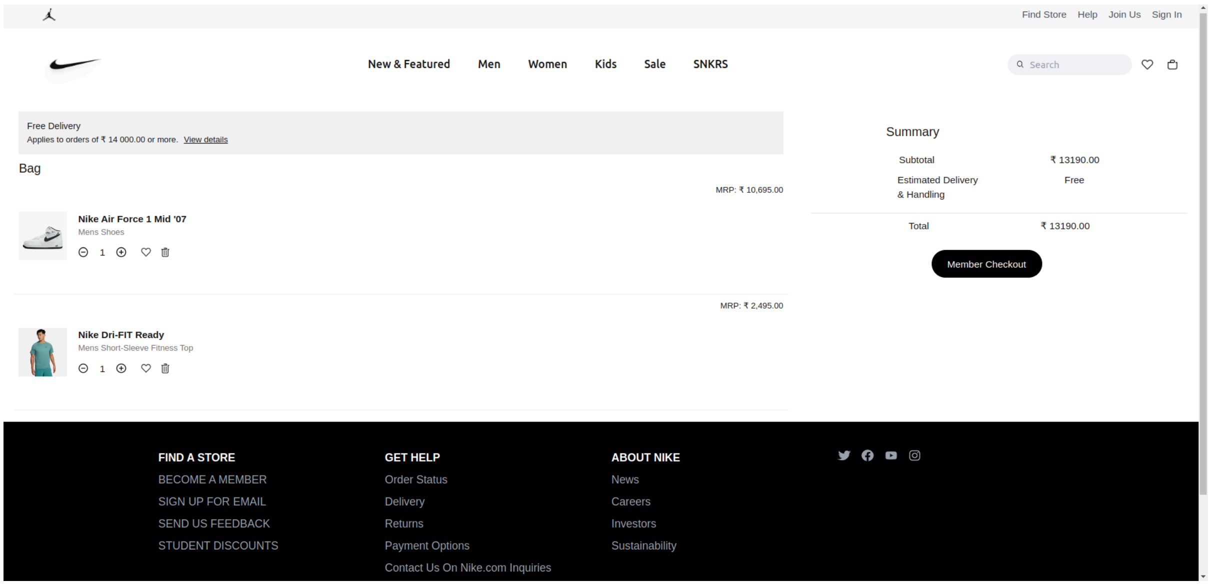
$4995



```tsx
'use client'
import React, { useEffect, useState } from 'react';
import { useRouter } from 'next/navigation';
import sanityClient from '@/sanity/lib/client';
import { urlFor } from '@/sanity/lib/image';
import { useCart } from '@/app/Context/CartContext';

interface Product {
  _id: number;
  productName: string;
  category: string;
  price: string;
  inventory: number;
  colors?: string[];
  status: string;
  description: string;
  image?: any;
}
const ProductDetail = ({ params }: { params: Promise<{ id: string }> }) => {
  const { addItem } = useCart();
  const router = useRouter();
  const [product, setProduct] = useState<Product | null>(null);
  const [productId, setProductId] = useState<string | null>(null);

  useEffect(() => {

    const getParams = async () => {
      const unwrappedParams = await params;
      setProductId(unwrappedParams.id); // Get the id from params
    };

    getParams();
  }, [params]);

  useEffect(() => {
    if (productId) {
      const fetchProduct = async () => {
        const query = `*[_type == "product" && _id == "${productId}"]{
          _id,
          productName,
          category,
          price,
          inventory,
          colors,
          status,
          description,
          image
        }`;
        const data = await sanityClient.fetch(query);
        setProduct(data[0] || null);
      };

      fetchProduct();
    }
  }, [productId]); // Fetch product when the productId changes
```

3.Add to Cart Functionality:

The "Add to Cart" functionality enables users to select and save products for purchase by clicking an "Add to Cart" button.



Find Store Help Join Us Sign In

New & Featured Men Women Kids Sale SNKRS

Search

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

Bag

MRP: ₹ 10,695.00

Nike Air Force 1 Mid '07
Mens Shoes
1

MRP: ₹ 2,495.00

Nike Dri-FIT Ready
Mens Short-Sleeve Fitness Top
1

Summary

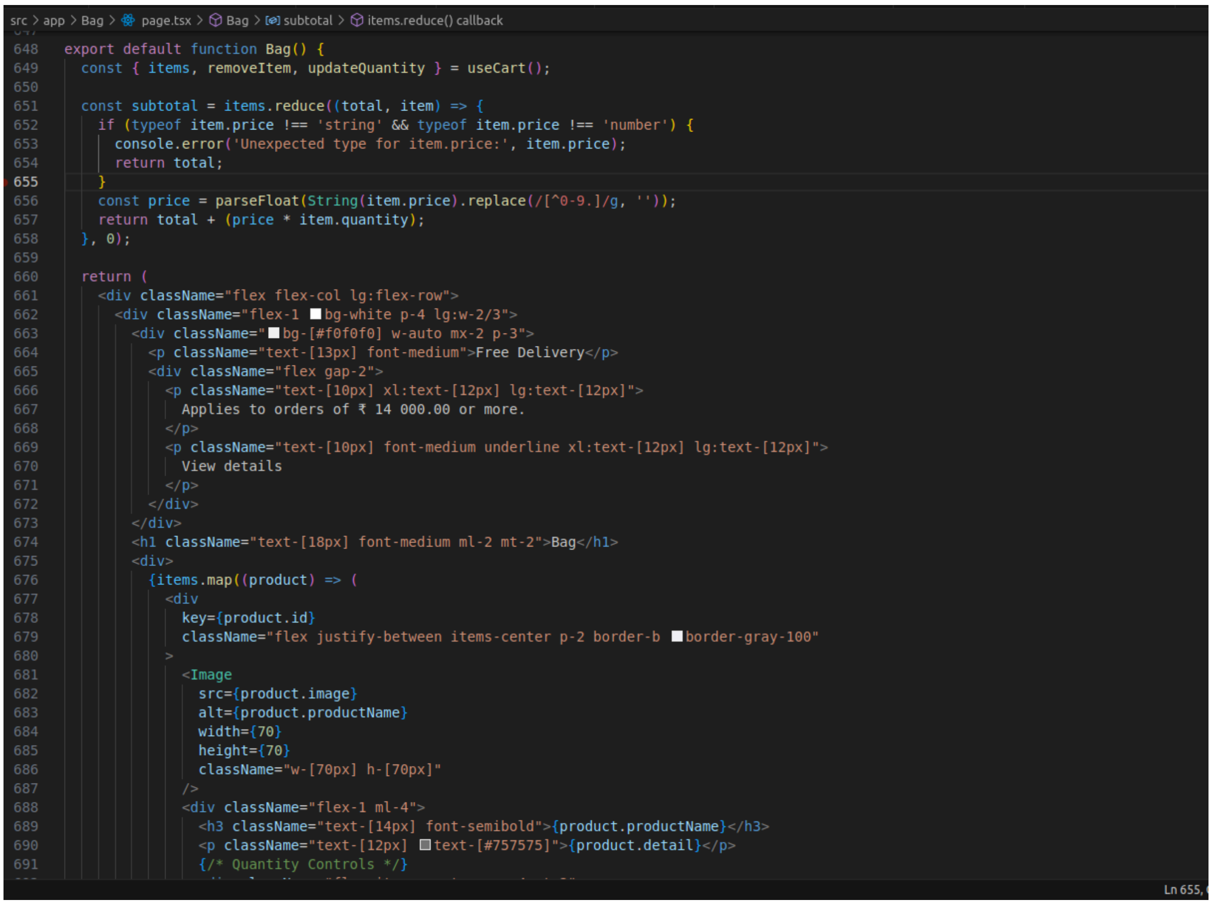
Subtotal	₹ 13190.00
Estimated Delivery & Handling	Free
Total	₹ 13190.00

Member Checkout

FIND A STORE
BECOME A MEMBER
SIGN UP FOR EMAIL
SEND US FEEDBACK
STUDENT DISCOUNTS

GET HELP
Order Status
Delivery
Returns
Payment Options
Contact Us On Nike.com Inquiries

ABOUT NIKE
News
Careers
Investors
Sustainability



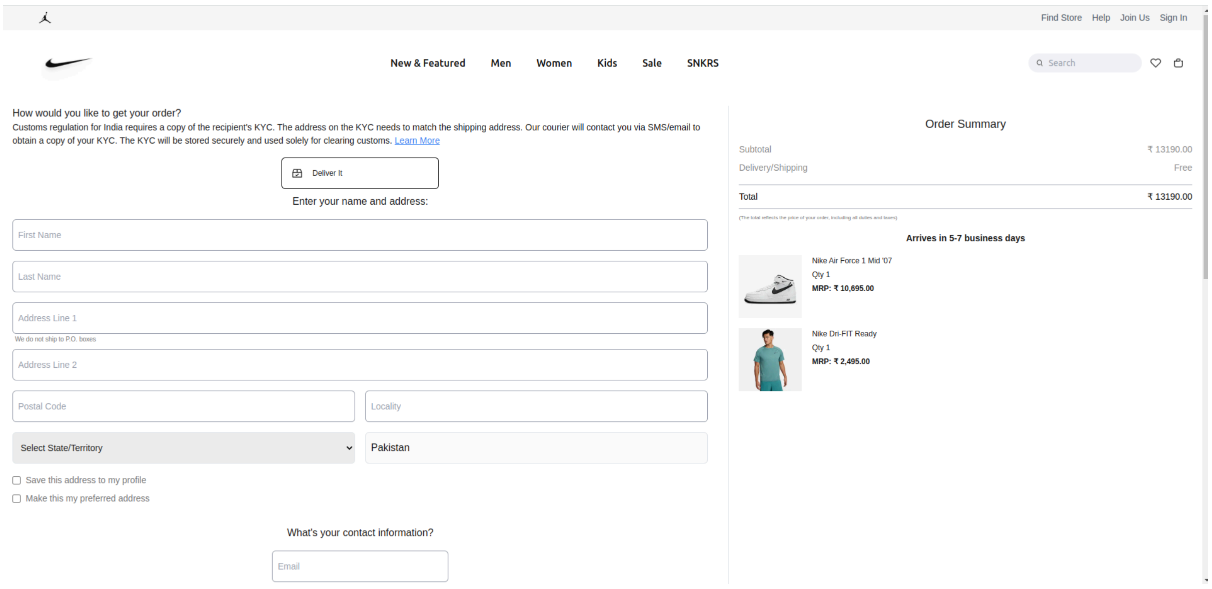
src > app > Bag > page.tsx > Bag > subtotal > items.reduce() callback

```tsx
export default function Bag() {
  const { items, removeItem, updateQuantity } = useCart();

  const subtotal = items.reduce((total, item) => {
    if (typeof item.price !== 'string' && typeof item.price !== 'number') {
      console.error('Unexpected type for item.price:', item.price);
      return total;
    }
    const price = parseFloat(String(item.price).replace(/[^0-9.]/g, ''));
    return total + (price * item.quantity);
  }, 0);

  return (
    <div className="flex flex-col lg:flex-row">
      <div className="flex-1 bg-white p-4 lg:w-2/3">
        <div className="bg-[#f0f0f0] w-auto mx-2 p-3">
          <p className="text-[13px] font-medium">Free Delivery</p>
          <div className="flex gap-2">
            <p className="text-[10px] xl:text-[12px] lg:text-[12px]">
              Applies to orders of ₹ 14 000.00 or more.
            </p>
            <p className="text-[10px] font-medium underline xl:text-[12px] lg:text-[12px]">
              View details
            </p>
          </div>
        </div>
        <h1 className="text-[18px] font-medium ml-2 mt-2">Bag</h1>
        <div>
          {items.map((product) => (
            <div
              key={product.id}
              className="flex justify-between items-center p-2 border-b border-gray-100"
            >
              <Image
                src={product.image}
                alt={product.productName}
                width={70}
                height={70}
                className="w-[70px] h-[70px]"
              />
              <div className="flex-1 ml-4">
                <h3 className="text-[14px] font-semibold">{product.productName}</h3>
                <p className="text-[12px] text-[#757575]">{product.detail}</p>
                {/* Quantity Controls */}
```

Ln 655,

4.Checkout Page:

Implemented a dedicated checkout page where users can review their selected items, adjust quantities, and proceed to payment.



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 clearing customs. Learn More

Deliver It

Enter your name and address:

First Name

Last Name

Address Line 1

We do not ship to P.O. boxes

Address Line 2

Postal Code | Locality

Select State/Territory | Pakistan

☐ Save this address to my profile

☐ Make this my preferred address

What's your contact information?

Email

Order Summary

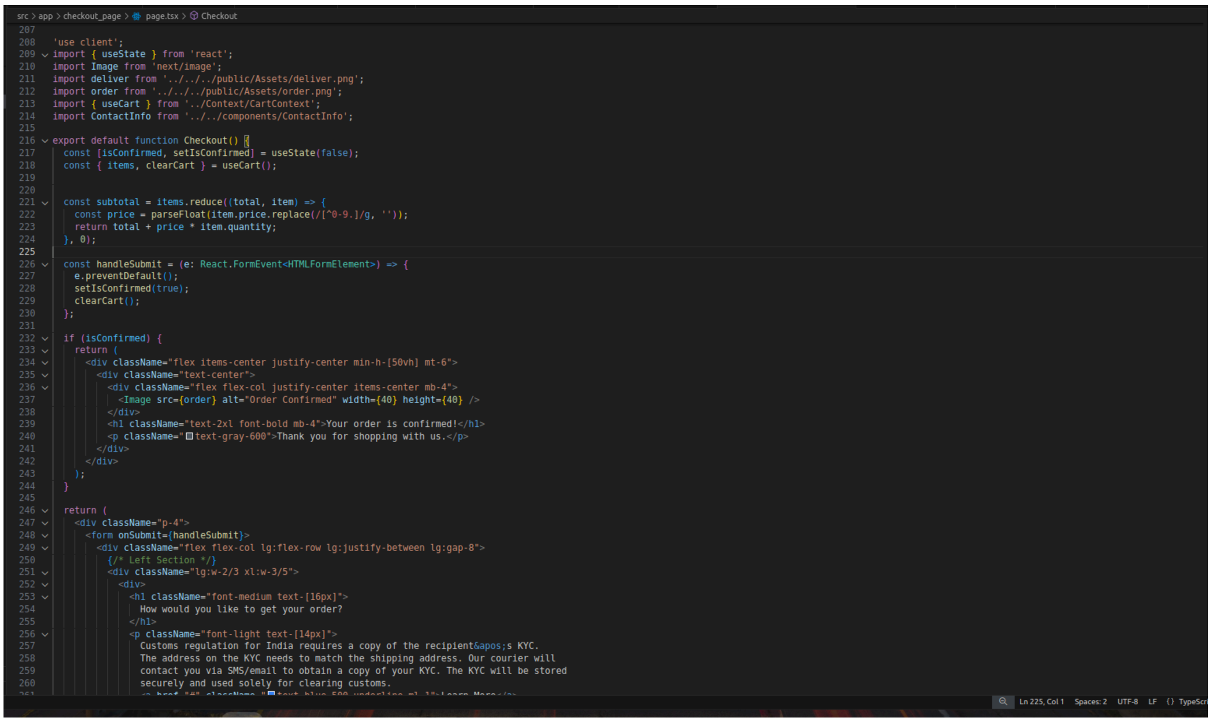
Subtotal	₹ 13190.00
Delivery/Shipping	Free
Total	₹ 13190.00

(The total reflects the price of your order, including all duties and taxes)

Arrives in 5-7 business days

Nike Air Force 1 Mid '07
Qty 1
MRP: ₹ 10,695.00

Nike Dri-FIT Ready
Qty 1
MRP: ₹ 2,495.00



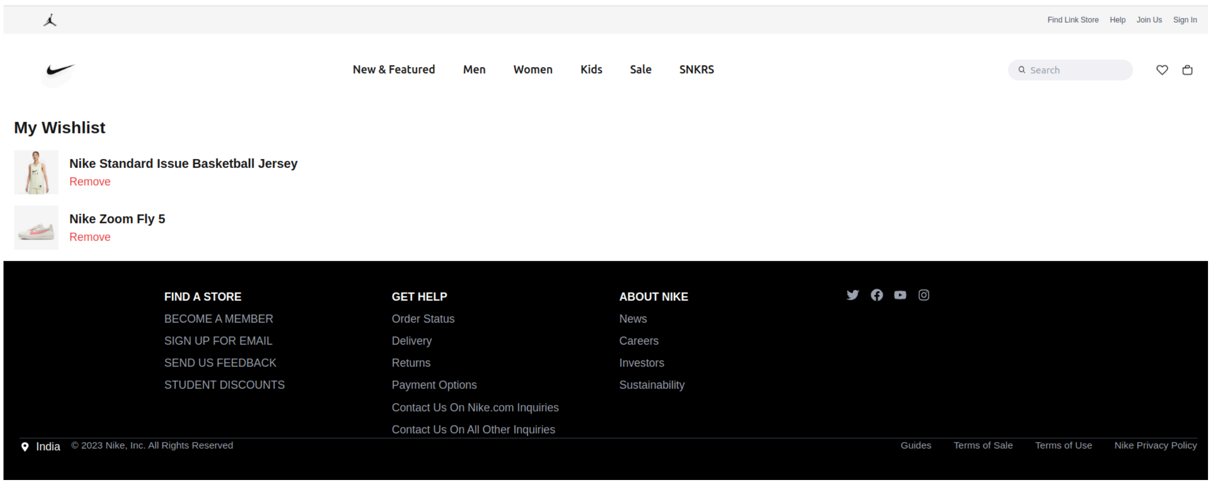
src > app > checkout_page > page.tsx > Checkout

```tsx

'use client';
import { useState } from 'react';
import Image from 'next/image';
import deliver from '../../../public/Assets/deliver.png';
import order from '../../../public/Assets/order.png';
import { useCart } from '../Context/CartContext';
import ContactInfo from '../../components/ContactInfo';

export default function Checkout() {
  const [isConfirmed, setIsConfirmed] = useState(false);
  const { items, clearCart } = useCart();


  const subtotal = items.reduce((total, item) => {
    const price = parseFloat(item.price.replace(/[^0-9.]/g, ''));
    return total + price * item.quantity;
  }, 0);

  const handleSubmit = (e: React.FormEvent<HTMLFormElement>) => {
    e.preventDefault();
    setIsConfirmed(true);
    clearCart();
  };

  if (isConfirmed) {
    return (
      <div className="flex items-center justify-center min-h-[50vh] mt-6">
        <div className="text-center">
          <div className="flex flex-col justify-center items-center mb-4">
            <Image src={order} alt="Order Confirmed" width={40} height={40} />
          </div>
          <h1 className="text-2xl font-bold mb-4">Your order is confirmed!</h1>
          <p className="text-gray-600">Thank you for shopping with us.</p>
        </div>
      </div>
    );
  }

  return (
    <div className="p-4">
      <form onSubmit={handleSubmit}>
        <div className="flex flex-col lg:flex-row lg:justify-between lg:gap-8">
          {/* Left Section */}
          <div className="lg:w-2/3 xl:w-3/5">
            <div>
              <h1 className="font-medium text-[16px]">
                How would you like to get your order?
              </h1>
              <p className="font-light text-[14px]">
                Customs regulation for India requires a copy of the recipient&apos;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 clearing customs.
                <a href="#" className="text-blue-500 underline ml-1">Learn More</a>
```

Ln 225, Col 1   Spaces: 2   UTF-8   LF   {} TypeScri

5.Wishlist Feature:

Added a wishlist functionality for users to save their favorite products for later.



My Wishlist

Nike Standard Issue Basketball Jersey
Remove

Nike Zoom Fly 5
Remove

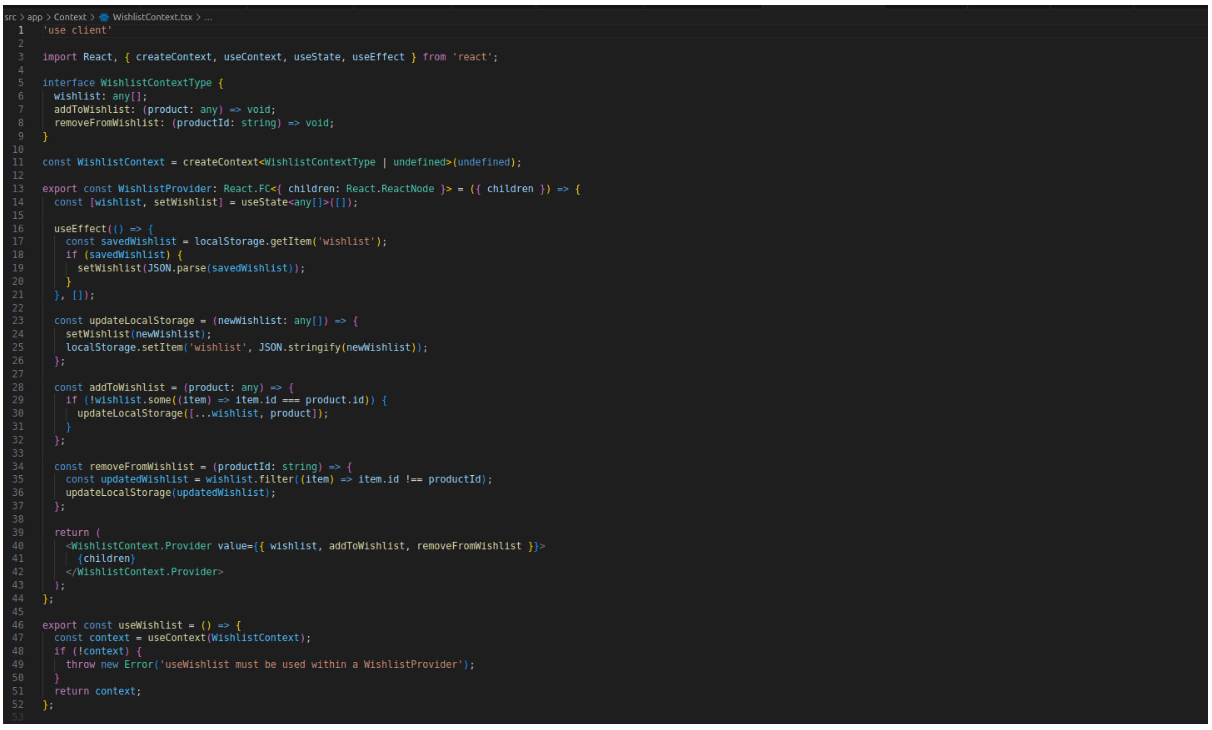
FIND A STORE
BECOME A MEMBER
SIGN UP FOR EMAIL
SEND US FEEDBACK
STUDENT DISCOUNTS

GET HELP
Order Status
Delivery
Returns
Payment Options
Contact Us On Nike.com Inquiries
Contact Us On All Other Inquiries

ABOUT NIKE
News
Careers
Investors
Sustainability

India © 2023 Nike, Inc. All Rights Reserved

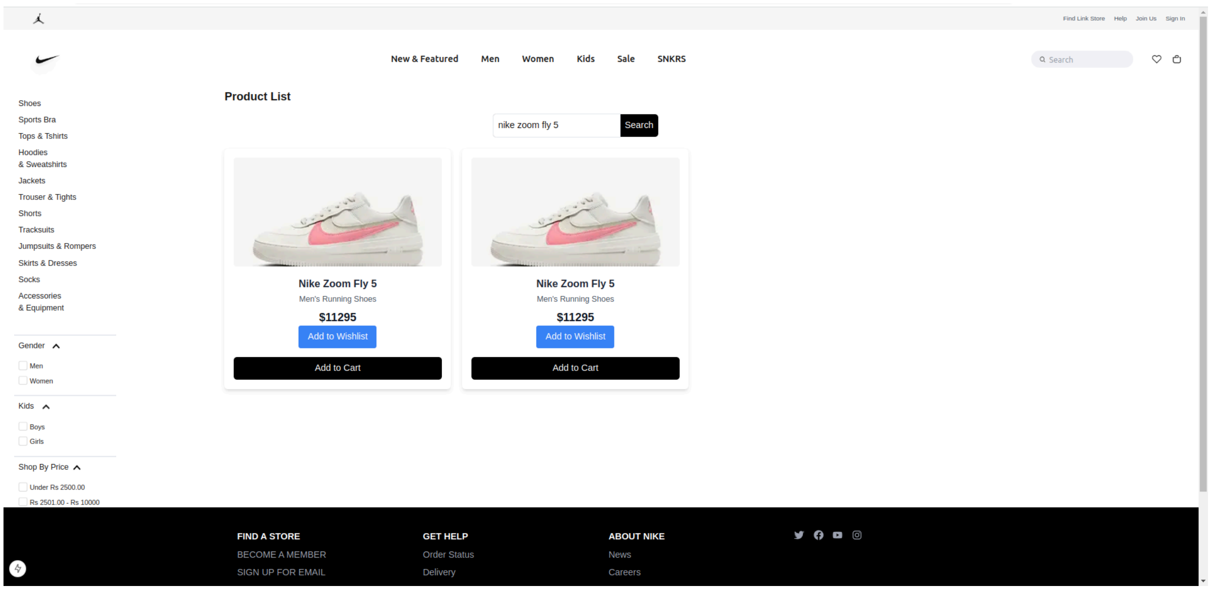
Guides Terms of Sale Terms of Use Nike Privacy Policy

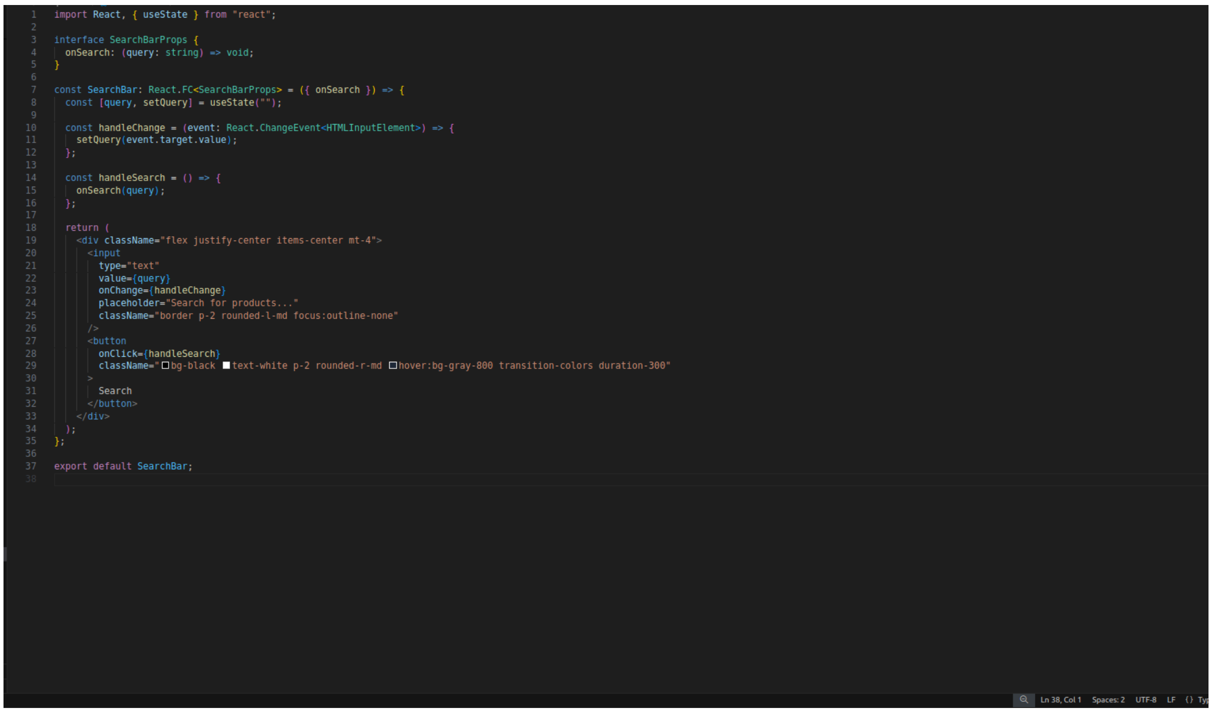

src > app > Context > WishlistContext.tsx > ...

```tsx
'use client'

import React, { createContext, useContext, useState, useEffect } from 'react';

interface WishlistContextType {
  wishlist: any[];
  addToWishlist: (product: any) => void;
  removeFromWishlist: (productId: string) => void;
}

const WishlistContext = createContext<WishlistContextType | undefined>(undefined);

export const WishlistProvider: React.FC<{ children: React.ReactNode }> = ({ children }) => {
  const [wishlist, setWishlist] = useState<any[]>([]);

  useEffect(() => {
    const savedWishlist = localStorage.getItem('wishlist');
    if (savedWishlist) {
      setWishlist(JSON.parse(savedWishlist));
    }
  }, []);

  const updateLocalStorage = (newWishlist: any[]) => {
    setWishlist(newWishlist);
    localStorage.setItem('wishlist', JSON.stringify(newWishlist));
  };

  const addToWishlist = (product: any) => {
    if (!wishlist.some((item) => item.id === product.id)) {
      updateLocalStorage([...wishlist, product]);
    }
  };

  const removeFromWishlist = (productId: string) => {
    const updatedWishlist = wishlist.filter((item) => item.id !== productId);
    updateLocalStorage(updatedWishlist);
  };

  return (
    <WishlistContext.Provider value={{ wishlist, addToWishlist, removeFromWishlist }}>
      {children}
    </WishlistContext.Provider>
  );
};

export const useWishlist = () => {
  const context = useContext(WishlistContext);
  if (!context) {
    throw new Error('useWishlist must be used within a WishlistProvider');
  }
  return context;
};
```

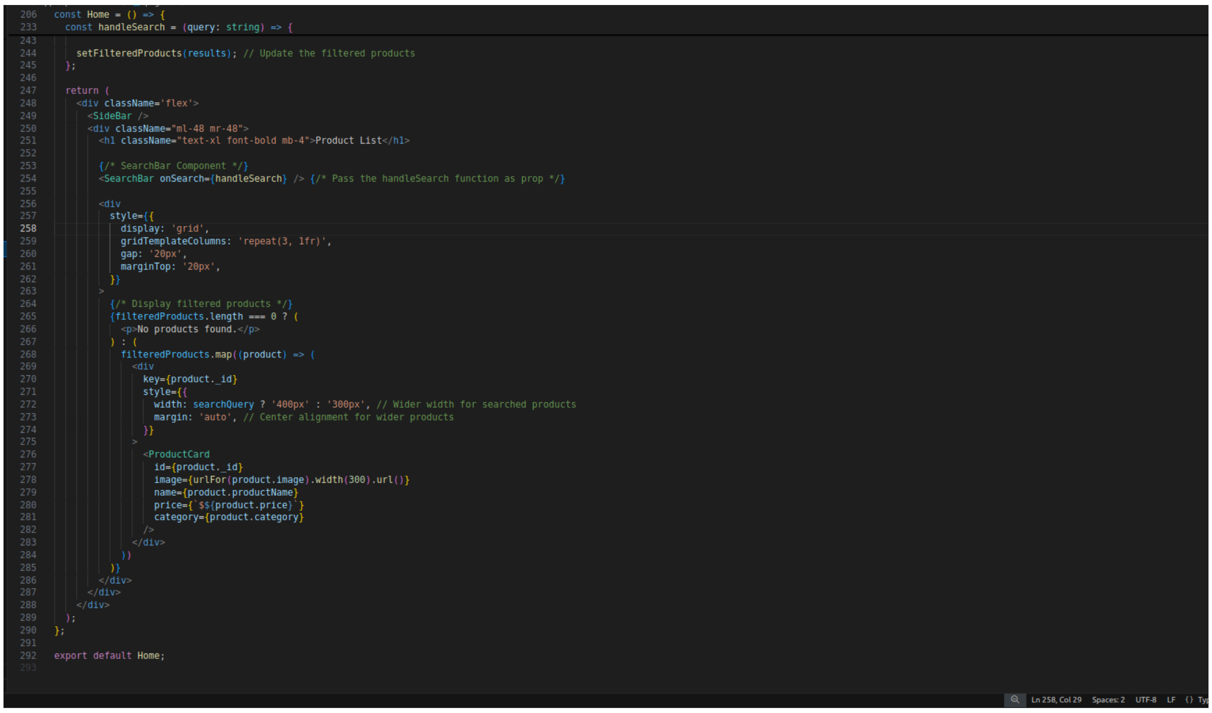
6. Search Bar:

Implement a fully functional category filtering mechanism and a real-time search bar to filter products by name or category.





```tsx
import React, { useState } from "react";

interface SearchBarProps {
  onSearch: (query: string) => void;
}

const SearchBar: React.FC<SearchBarProps> = ({ onSearch }) => {
  const [query, setQuery] = useState("");

  const handleChange = (event: React.ChangeEvent<HTMLInputElement>) => {
    setQuery(event.target.value);
  };

  const handleSearch = () => {
    onSearch(query);
  };

  return (
    <div className="flex justify-center items-center mt-4">
      <input
        type="text"
        value={query}
        onChange={handleChange}
        placeholder="Search for products..."
        className="border p-2 rounded-l-md focus:outline-none"
      />
      <button
        onClick={handleSearch}
        className="bg-black text-white p-2 rounded-r-md hover:bg-gray-800 transition-colors duration-300"
      >
        Search
      </button>
    </div>
  );
};

export default SearchBar;
```



```tsx
const Home = () => {
  const handleSearch = (query: string) => {

    setFilteredProducts(results); // Update the filtered products
  };

  return (
    <div className='flex'>
      <SideBar />
      <div className="ml-48 mr-48">
        <h1 className="text-xl font-bold mb-4">Product List</h1>

        {/* SearchBar Component */}
        <SearchBar onSearch={handleSearch} /> {/* Pass the handleSearch function as prop */}

        <div
          style={{
            display: 'grid',
            gridTemplateColumns: 'repeat(3, 1fr)',
            gap: '20px',
            marginTop: '20px',
          }}
        >
          {/* Display filtered products */}
          {filteredProducts.length === 0 ? (
            <p>No products found.</p>
          ) : (
            filteredProducts.map((product) => (
              <div
                key={product._id}
                style={{
                  width: searchQuery ? '400px' : '300px', // Wider width for searched products
                  margin: 'auto', // Center alignment for wider products
                }}
              >
                <ProductCard
                  id={product._id}
                  image={urlFor(product.image).width(300).url()}
                  name={product.productName}
                  price={`$${product.price}`}
                  category={product.category}
                />
              </div>
            ))
          )}
        </div>
      </div>
    </div>
  );
};

export default Home;
```

Ln 258, Col 29   Spaces: 2   UTF-8   LF   {} Typ

Steps Taken:

1.Fetched dynamic product data from Sanity CMS.

2. Built components such as `ProductCard`, `SearchBar`, and `Sidebar` to display and filter products.

3.Integrated dynamic routing for individual product pages.

4.Implemented **Add to Cart functionality** to manage shopping behavior.

5.Created a **Checkout Page** for users to review selected items and proceed with payments.

6.Added a **Wishlist Feature** for users to save their favorite products for later.

7.Styled the components for responsiveness and better user experience.

Challenges Faced:

1.Dynamic Routing Errors:

Resolved by debugging the `next/router` configuration.

2.Empty Image Source:

Fixed by adding fallback placeholders for images.

3.Search and Filter Logic:

Optimized by converting all text to lowercase for case-insensitive search.

Self-Validation Checklist

Frontend Component Development:

✔ ✘

Styling and Responsiveness:

✔ ✘

Code Quality:

✔ ✘

Documentation and Submission:

✔ ✘

Presented By:

Rimsha Mukhtar

Slot: Saturday 2 to 5