

## Day 04: Building Dynamic Frontend Components For Marketplace

Product Listing Component:

Code

```

You, 3 seconds ago | 1 author (You)
'use client'

import React, { useEffect, useState } from "react"
import { Product } from "../../types/products"
import { client } from "@sanity/lib/client"
import { allProducts } from "@sanity/lib/queries"
import Link from "next/link"
import Image from "next/image"
import { urlFor } from "@sanity/lib/image"

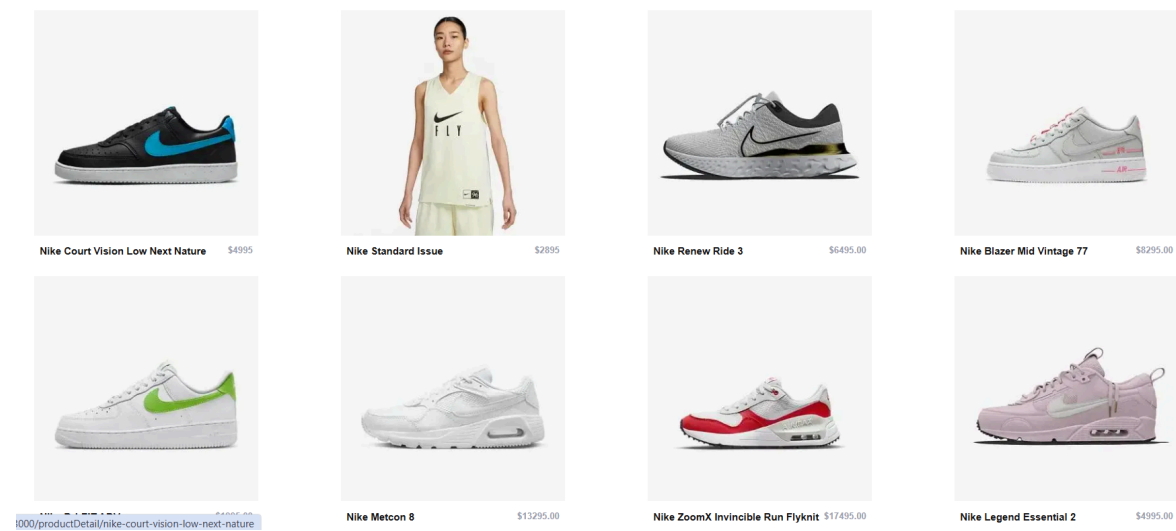
export default function FetchProducts(){

  const [products , setproducts] =useState<Product[]> ([])

  useEffect(() => {
    async function fetching(){
      const fetchingProducts : Product[] = await client.fetch(allProducts)
      setproducts(fetchingProducts)
    }
    fetching()
  }, [])

  return <div className="grid grid-cols-4 gap-[30px] pl-[120px] pt-[20px]">
    {
      products.map((products) => (
        <Link href={` /productDetail/${products.slug.current}`}>
          <div>
            {products.image && (
              <Image src={urlFor(products.image).url()} alt="" width={400} height={400}/>
            )}
            <div className="flex justify-between w-[400px] font-bold text-[18px] px-[10px] pt-[15px]">
              <p>{products.productName}</p>
              <p className="text-gray-400 text-[16px] font-semibold">${products.price}</p>
            </div>
          </div>
        </Link>
      ))
    }
  </div>
}
```

## OUTPUT



## PRODUCT DETAIL PAGE

### Code

```
You, 18 hours ago | 1 author (You)
interface ProductsPageProps {
  params : Promise<{slug : string}>
}

async function getProduct(slug : string): Promise<Product > {
  return client.fetch (
    gql`*[_type == "product" && slug.current == $slug][0] {
      _id,
      productName,
      category,
      _type,
      image,
      price,
      description,
    }`, {slug}
  )
}

export default async function ProductPage({params}: ProductsPageProps) {
  const {slug} = await params;
  const product = await getProduct(slug)

  return <div className="w-full flex h-[560px] px-[300px] mt-[90px] font-sans">
    {product?.image && (
      <Image src={urlFor(product.image).url()} alt="" width={350} height={350} className="w-[500px] h-[500px]" />
    )}

    <div className="w-[600px] ml-[70px] mt-[45px]">
      <p className="text-[26px] font-bold">{product?.productName}</p>
      <p className="text-[19px] text-gray-400 mt-[5px]">{product?.category}</p>
      <p className="text-[22px] font-bold mt-[5px]">${product?.price}</p>
      <p className="text-[18px] text-gray-500 mt-[5px]">{product?.description}</p>

      <AddToCartButton product={product} />
    </div>
  </div>
}
```



### Nike Metcon 8

Men's Shoes

**\$13295.00**

The Nike Metcon 8 is designed for high-intensity training, offering a stable base and a durable, grippy outsole. With a comfortable, breathable upper and enhanced support, these shoes are perfect for lifting, running, and various training activities.

Add To Bag

## Category format

Code :

```
import { client } from "@sanity/lib/client"

export default async function WomensItems(){
  const query = `
*[_type == "product" && category == "Women's Shoes"]{
  id,
  productName,
  price,
  inventory,
  category,
  colors,
  status,
  description,
  "productSlug":slug.current,
  "productImage":image.asset->url
}
`

  const Data = await client.fetch(query)
  return <div className="grid grid-cols-3">
    {
      Data.map((Data:any) => [
        <div>
          <img src={Data.productImage} alt="" className="lg:w-[600px] lg:h-[600px] ex:w-[300px] ex:h-[300px]" />
          <p>{Data.productName}</p>
          <p>{Data.category}</p>
          <p>{Data.price}</p>
        </div>
      ])
    }
  </div>
}
```



localhost:3000/Female

Output

## Search Bar

## Code

```
import { client } from "@sanity/lib/client";
import { useEffect, useState } from "react";

export default function SearchProducts() {
  const [products, setProducts] = useState([]);
  const [category, setCategory] = useState("all");

  const fetchProducts = async (selectedCategory: string) => {
    let query = `
      *[_type == "product" & ${
        selectedCategory !== "all" ? `& category == "${selectedCategory}"` : ""
      }] {
        productName,
        price,
        category,
        status,
        "productImage": image.asset->url,
        "productSlug": slug.current
      }
    `;
    const fetchedProducts = await client.fetch(query);
    setProducts(fetchedProducts);
  };

  useEffect(() => {
    fetchProducts(category);
  }, [category]);

  return (
    <div className="flex justify-center mt-10">
      <select
        value={category}
        onChange={(e) => setCategory(e.target.value)}
        className="w-[250px] hidden md:flex lg:h-[40px] ex:h-[35px] ex:ml-[60px] ex:mt-[8px] rounded-full bg-[#F5F5F5] border border-black text-black lg:ml-[180px] px-[10px] hover:bg-black"
      >
        <option value="all">All</option>
        <option value="Men's Shoes">Men Shoes</option>
        <option value="Women's Shoes">Women Shoes</option>
        <option value="Accessories">Accessories</option>
        { /* Add more categories here as needed */ }
      </select>
    </div>
  );
}
```

## Cart Component

## Output

### Shopping Cart

**Nike ZoomX Invincible Run Flyknit**  
\$17495.00

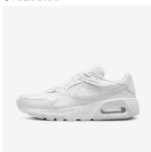


Men's Shoes

- 1 +

Remove

**Nike Metcon 8**  
\$13295.00



Men's Shoes

- 1 +

Remove

Total: \$30790

Checkout

Code :

```
const handleRemove = (id: string) => {
  }).then((result) => {
  });
});

const handleQuantityChange = (id: string, quantity: number) => {
  updateQuantity(id, quantity);
  setCartItem(getItems());
};

const handleIncrement = (id: string) => {
  const product = cartItem.find((item) => item._id === id);
  if (product) {
    handleQuantityChange(id, product.inventory + 1);
  }
};

const handleDecrement = (id: string) => {
  const product = cartItem.find((item) => item._id === id);
  if (product && product.inventory > 1) {
    handleQuantityChange(id, product.inventory - 1);
  }
};

const router = useRouter();
const handleProceed = () => {
  Swal.fire({
    title: 'Proceed to Checkout',
    text: 'Please review your cart before checkout.',
    icon: 'question',
    showCancelButton: true,
    confirmButtonColor: '#3085d6',
    cancelButtonColor: '#d33',
    confirmButtonText: 'Yes, Proceed!',
  }).then((result) => {
    if (result.isConfirmed) {
      Swal.fire('Success', 'Your order has been successfully processed!', 'success');
      setCartItem([]);
      router.push("/checkout")
    }
  });
};
```