

Day 3 API Integration

[illegible]

```
export const productSchema = {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'productName',
      title: 'Product Name',
      type: 'string',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'inventory',
      title: 'Inventory',
      type: 'number',
    },
    {
      name: 'colors',
      title: 'Colors',
      type: 'array',
      of: [{ type: 'string' }],
    },
    {
      name: 'status',
      title: 'Status',
      type: 'string',
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image', // Using Sanity's image type for image field
      options: {
        hotspot: true,
      },
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
  ],
}
```



```
app > components > shoes > page.tsx > [0] SHOES > [0] useEffect() callback > [0] fetchProducts > [0] fetchedProducts
4 import { urlFor } from "@sanity/lib/image";
5 import { allProducts, four } from "@sanity/lib/queries";
6 import { Product } from "@types/products";
7 import Image from "next/image";
8 import React, { useEffect, useState } from "react";
9
10 Tabnine | Edit | Explain
11 const SHOES = () => {
12   const [products, setProducts] = useState<Product[]>([]);
13
14   Pieces: Comment | Pieces: Explain
15   useEffect(() => {
16     Pieces: Comment | Pieces: Explain
17     async function fetchProducts() {
18       const fetchedProducts: Product[] = await client.fetch(allProducts);
19       setProducts(fetchedProducts);
20     }
21     fetchProducts();
22   }, []);
23
24   return (
25     <div className="max-w-6xl mx-auto px-4 py-8">
26       <h1 className="text-2xl font-bold mb-6 text-center">Our Latest Shoes</h1>
27       <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
28         Pieces: Comment | Pieces: Explain
29         {products.map((product) => (
30           <div
31             key={product._id}
32             className="border rounded-lg shadow-md p-4 hover:shadow-lg transition duration-200"
33           >
```

```
34 <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
35   Pieces: Comment | Pieces: Explain
36   {products.map((product) => (
37     <div
38       key={product._id}
39       className="border rounded-lg shadow-md p-4 hover:shadow-lg transition duration-200"
40     >
41       <div>
42         <img
43           src={urlFor(product.image).url()}
44           alt={product.productName}
45           width={200}
46           height={200}
47           className="w-full h-48 object-cover rounded-md"
48         />
49       </div>
50       <div>
51         <h2 className="text-lg font-semibold mt-4">{product.productName}</h2>
52         <p className="text-gray-500 mt-2">
53           {product.price ? `${product.price}` : "Price not available"}
54         </p>
55       </div>
56     </div>
57   )
58 )
59 </div>
```

