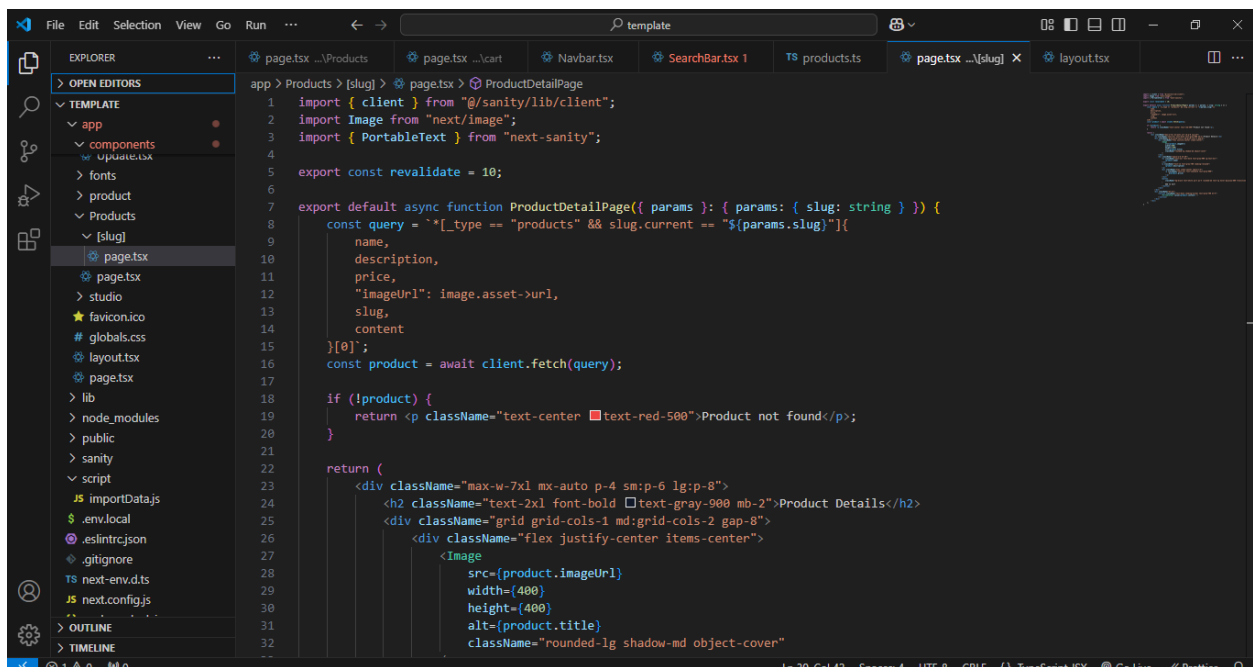


"Day 4 - Dynamic Frontend Components - [Shop.Co by MZ]"

To create dynamic routes for product pages, I set up a file structure in Next.js with `/pages/products/[slug].tsx` to handle product-specific routes. I utilized `getStaticProps` to fetch product data from Sanity based on the dynamic slug parameter. This data is then passed to the component to render the relevant product details. Finally, the product slug is used within links and the page content to dynamically display each product's information.



The screenshot shows a VS Code editor with the file explorer on the left and the code editor on the right. The file explorer shows the project structure, with the `pages/products/[slug].tsx` file selected. The code editor displays the implementation of the `ProductDetailPage` component. The code includes imports for `client`, `Image`, and `PortableText` from Next.js and Sanity. It defines a `revalidate` constant and a `ProductDetailPage` function that takes `params` as an argument. The function uses `getStaticProps` to fetch product data from Sanity based on the `slug` parameter. The data is then used to render the product details, including the product name, description, price, image, and content. The code also includes a fallback for when the product is not found.

```
1 import { client } from '@sanity/lib/client';
2 import Image from 'next/image';
3 import { PortableText } from 'next-sanity';
4
5 export const revalidate = 10;
6
7 export default async function ProductDetailPage({ params }: { params: { slug: string } }) {
8   const query = `*[ _type == "products" && slug.current == "${params.slug}" ]{
9     name,
10    description,
11    price,
12    "imageUrl": image.asset->url,
13    slug,
14    content
15  }[0]`;
16   const product = await client.fetch(query);
17
18   if (!product) {
19     return <p className="text-center text-red-500">Product not found</p>;
20   }
21
22   return (
23     <div className="max-w-7xl mx-auto p-4 sm:p-6 lg:p-8">
24       <h2 className="text-2xl font-bold text-gray-900 mb-2">Product Details</h2>
25       <div className="grid grid-cols-1 md:grid-cols-2 gap-8">
26         <div className="flex justify-center items-center">
27           <Image
28             src={product.imageUrl}
29             width={400}
30             height={400}
31             alt={product.title}
32             className="rounded-lg shadow-md object-cover"
33           />
34         <div>
35           <p>{product.name}</p>
36           <p>{product.description}</p>
37           <p>{product.price}</p>
38           <p>{product.content}</p>
39         </div>
40       </div>
41     </div>
42   );
43 }
```

Here are some examples:

localhost:3000/Products/classic-black-long-sleeve-button-down-shirt

Sign up and get 20% off to your first order. [Sign Up Now](#)


SHOP.CO

Shop ▾ On Sale New Arrivals Brands

Search for products...

🛒 @

Product Details



Classic Black Long Sleeve Button-Down Shirt

This sleek black button-down shirt is the epitome of style and versatility. Designed with a tailored fit, it provides a sharp, refined look that's perfect for both professional and casual settings. The long sleeves and crisp collar offer a classic silhouette, while the lightweight fabric ensures comfort throughout the day. Whether worn at the office, for a dinner out, or dressed down with jeans, this black shirt adds a polished touch to any outfit. A must-have wardrobe staple for those who appreciate clean lines and timeless style.

\$190

Add to Cart

localhost:3000/Products/black-athletic-jogger-pants-with-side-stripes

Sign up and get 20% off to your first order. [Sign Up Now](#)


SHOP.CO

Shop ▾ On Sale New Arrivals Brands

Search for products...

🛒 @

Product Details

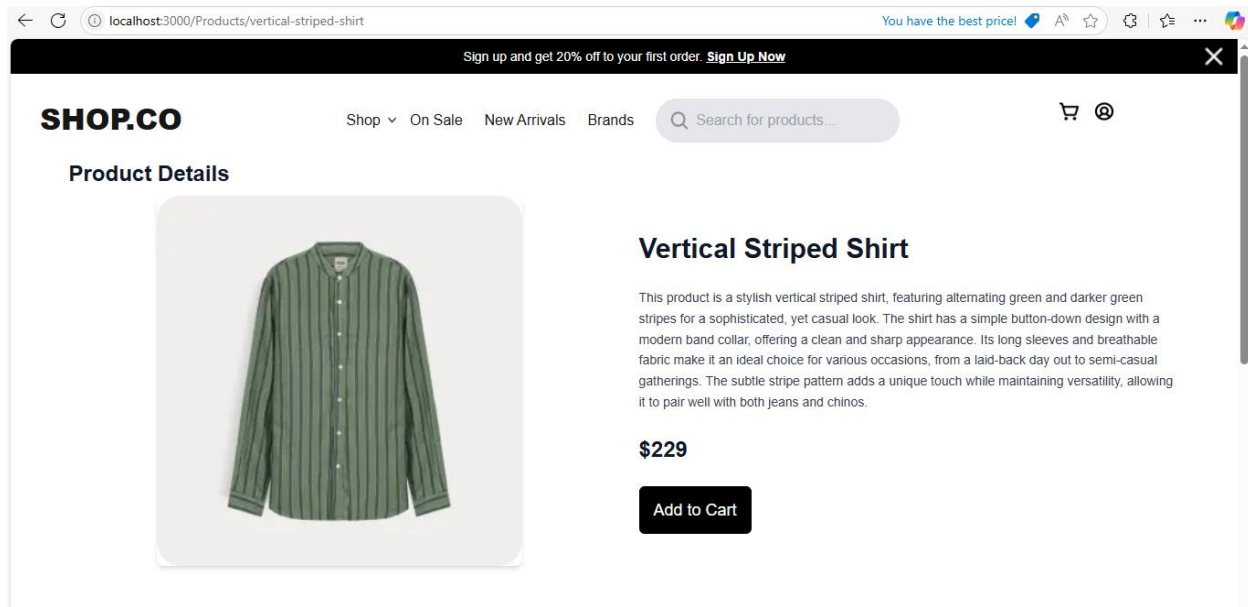


Black Athletic Jogger Pants with Side Stripes

These sporty black jogger pants offer a perfect blend of comfort and style. Featuring a relaxed fit with an elastic waistband and adjustable drawstring, they are ideal for both casual wear and light athletic activities. The distinctive white side stripes add a stylish, modern touch that enhances their sporty appeal. With deep pockets for convenience, these pants are versatile enough to pair with a variety of tops, from t-shirts to hoodies. Comfortable and easy to wear, these jogger pants are a great choice for a laid-back, active look.

\$180

Add to Cart



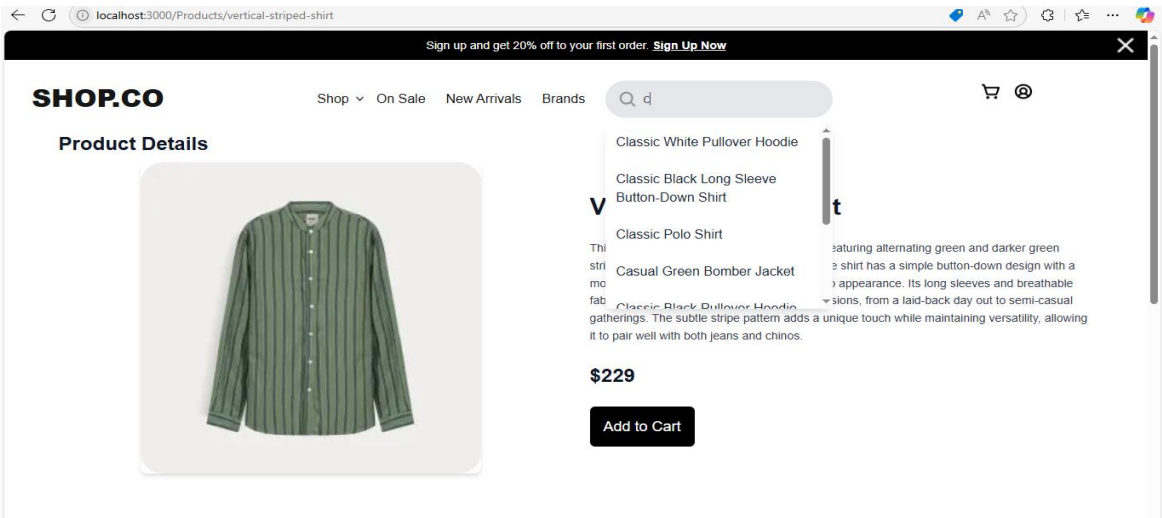
To set up a search bar with suggestions appearing after the first character, I used an input field where the value is bound to the `searchTerm` state. The `useEffect` hook listens for changes in the `searchTerm` and triggers a query to fetch products based on the input. The query is executed as soon as the user starts typing (with no character limit). Suggestions are displayed in a dropdown under the input field, showing relevant product names. I used Sanity's query language to fetch product data and filter it dynamically based on the input.

```

1  "use client";
2  import React, { useState, useEffect } from "react";
3  import Image from "next/image";
4  import { createClient } from "next-sanity";
5  import Link from "next/link";
6
7  // Sanity client setup
8  const client = createClient({
9    projectId: "i2nafnty",
10   dataset: "production",
11   apiVersion: "v2025-01-18",
12   useCdn: true,
13 });
14
15 // Define the structure of a product
16 interface Product {
17   name: string;
18   slug: { current: string };
19 }
20
21 const SearchBar: React.FC = () => {
22   const [searchTerm, setSearchTerm] = useState<string>("");
23   const [suggestions, setSuggestions] = useState<Product[]>([]);
24
25   // Fetch products from Sanity
26   const fetchProducts = async (query: string) => {
27     try {
28       // Sanity query string with parameterized search
29       const data = await client.fetch<
30         `[[_type == "products" && name match $query]](name, slug)`;
31     } catch (error) {
32       console.error("Error fetching products:", error);
33     }
34   };

```

1.



2.

