

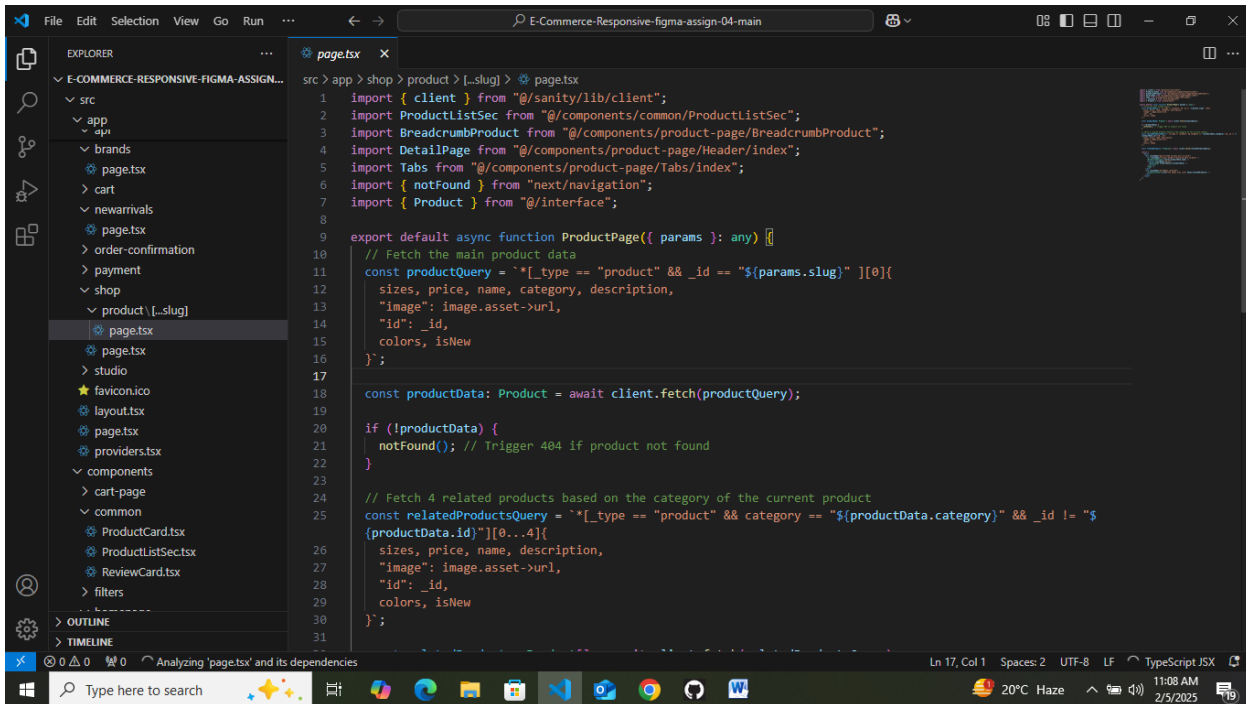
Dynamic Frontend Experiences with Sanity CMS

DAY 4

- I have displayed data through sanity and added changes into my website

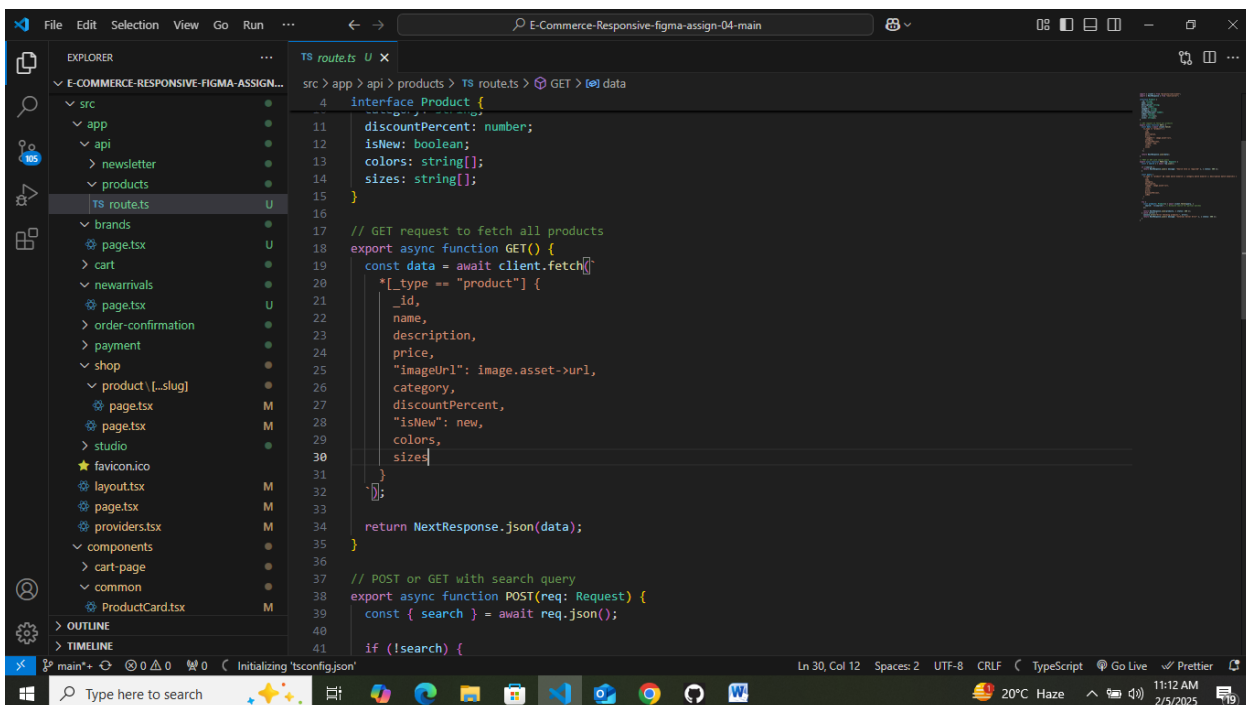
Code Snippets

Product Page



```
src > app > shop > product > [...slug] > page.tsx
1  import { client } from "@sanity/lib/client";
2  import ProductListSec from "@components/common/ProductListSec";
3  import BreadcrumbProduct from "@components/product-page/BreadcrumbProduct";
4  import DetailPage from "@components/product-page/Header/index";
5  import Tabs from "@components/product-page/Tabs/index";
6  import { notFound } from "next/navigation";
7  import { Product } from "@interface";
8
9  export default async function ProductPage({ params }: any) {}
10 // Fetch the main product data
11 const productQuery = `*[ _type == "product" && _id == "${params.slug}" ]{0}{
12   sizes, price, name, category, description,
13   "image": image.asset->url,
14   "id": _id,
15   colors, isNew
16 }`;
17
18 const productData: Product = await client.fetch(productQuery);
19
20 if (!productData) {
21   notFound(); // Trigger 404 if product not found
22 }
23
24 // Fetch 4 related products based on the category of the current product
25 const relatedProductsQuery = `*[ _type == "product" && category == "${productData.category}" && _id != "${productData.id}" ]{0...4}{
26   sizes, price, name, description,
27   "image": image.asset->url,
28   "id": _id,
29   colors, isNew
30 }`;
31
```

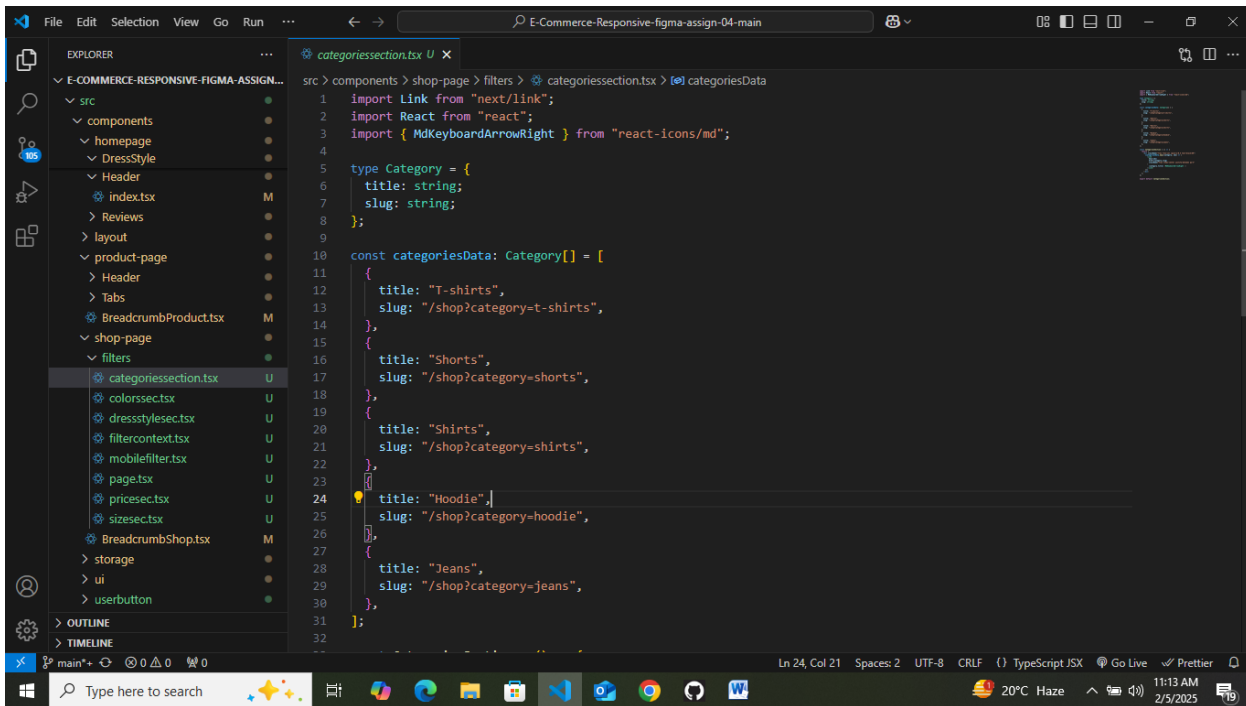
Data fetching



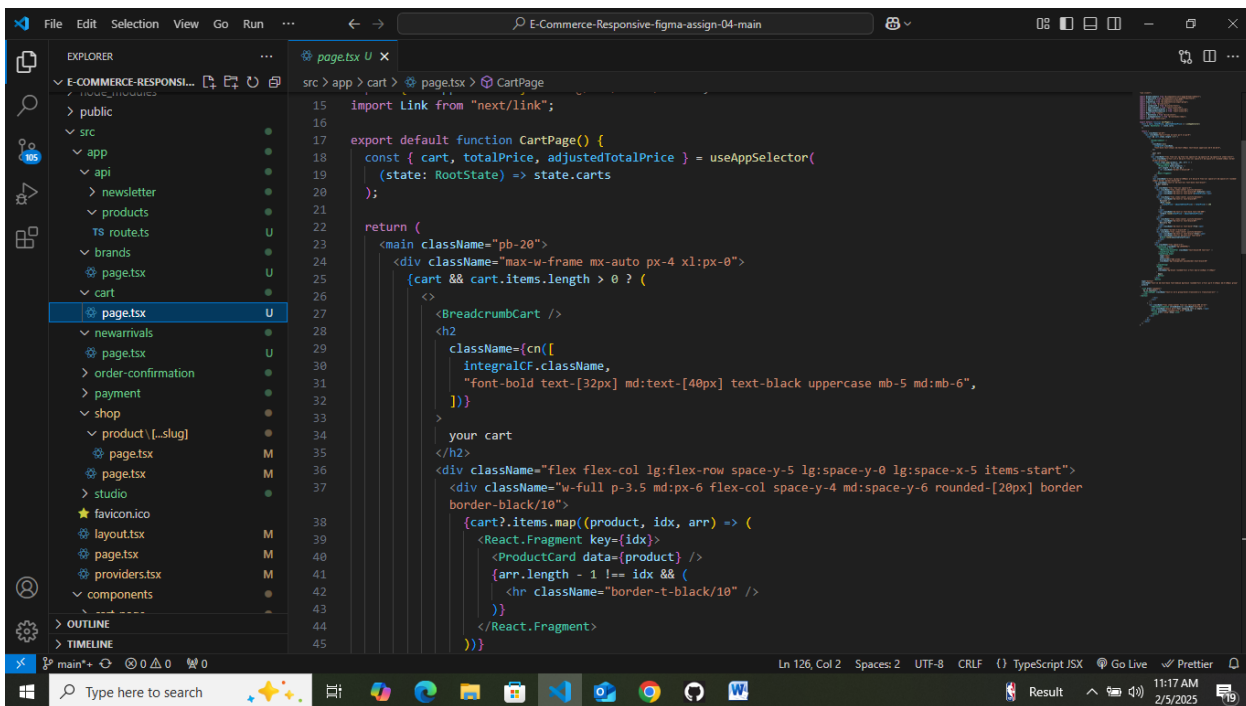
```
src > app > api > products > TS route.ts > GET > data
4  interface Product {
11   discountPercent: number;
12   isNew: boolean;
13   colors: string[];
14   sizes: string[];
15 }
16
17 // GET request to fetch all products
18 export async function GET() {
19   const data = await client.fetch(`
20     *[ _type == "product" ] {
21       _id,
22       name,
23       description,
24       price,
25       "imageUrl": image.asset->url,
26       category,
27       discountPercent,
28       "isNew": new,
29       colors,
30       sizes
31     }
32 `);
33
34   return NextResponse.json(data);
35 }
36
37 // POST or GET with search query
38 export async function POST(req: Request) {
39   const { search } = await req.json();
40
41   if (!search) {

```

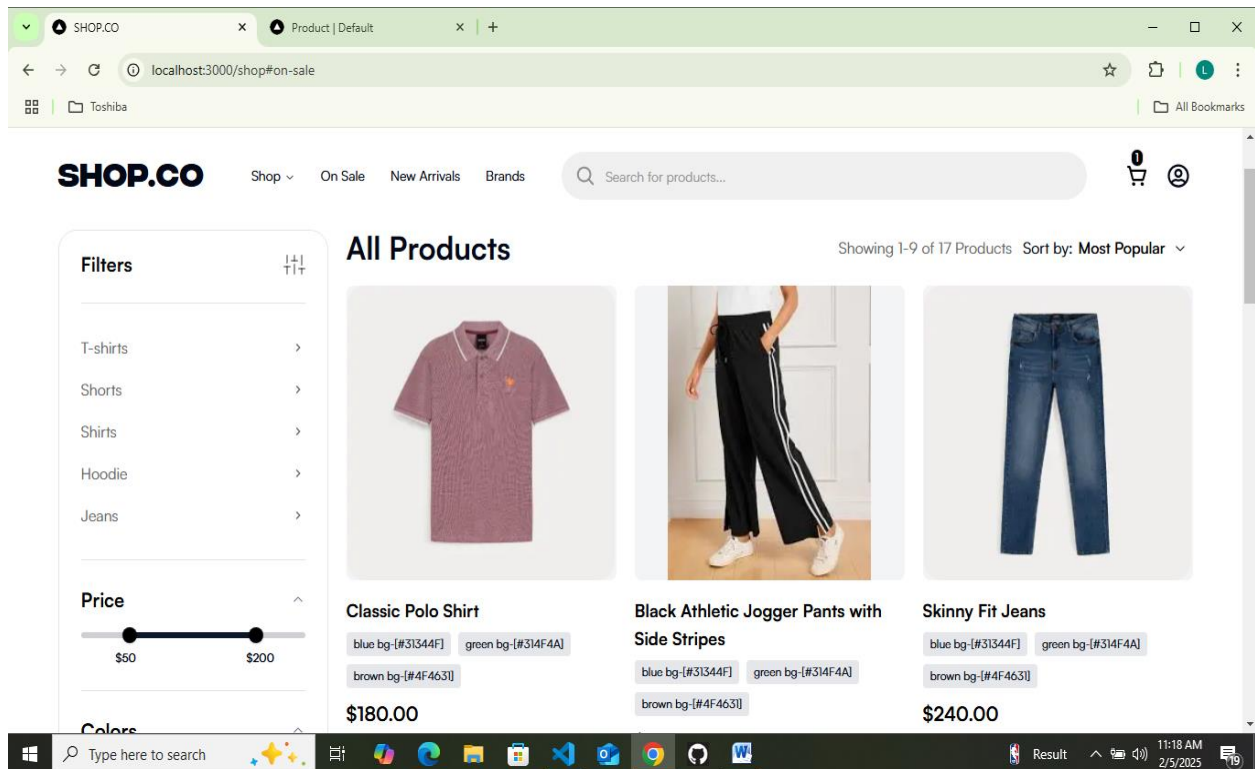
Filters



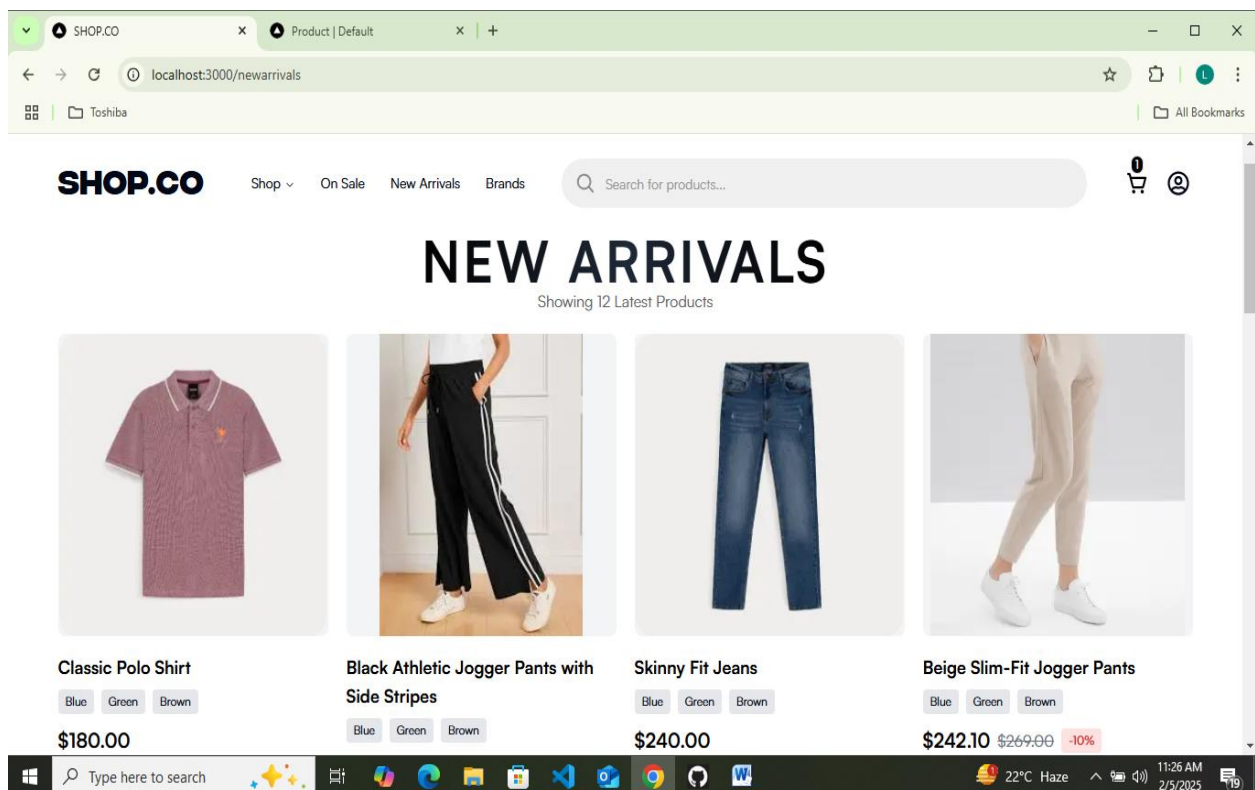
Cart



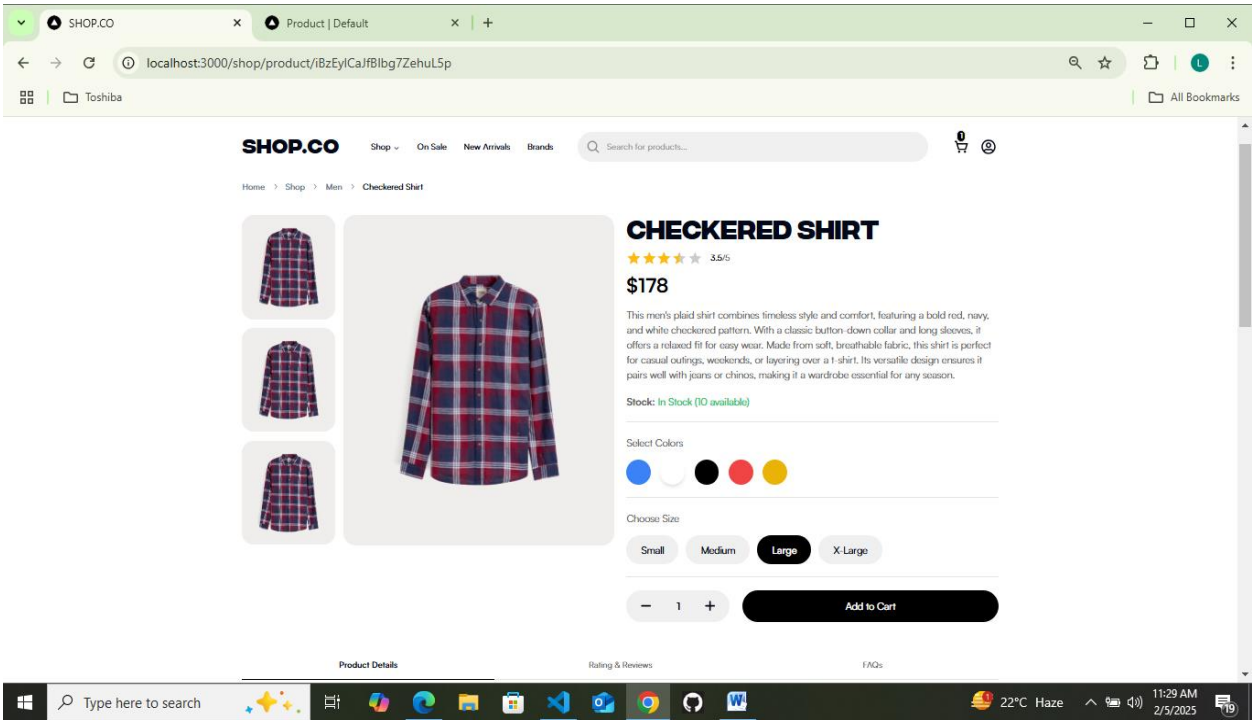
Frontend Filters



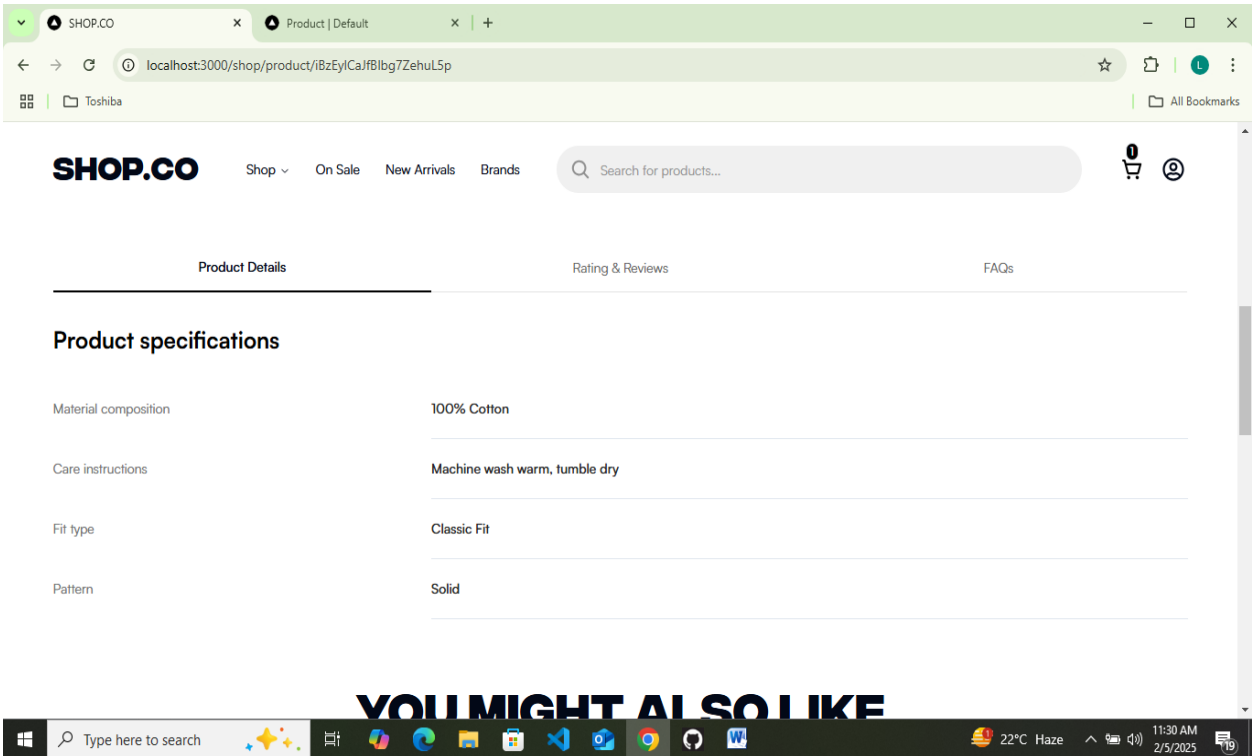
New Arrivals



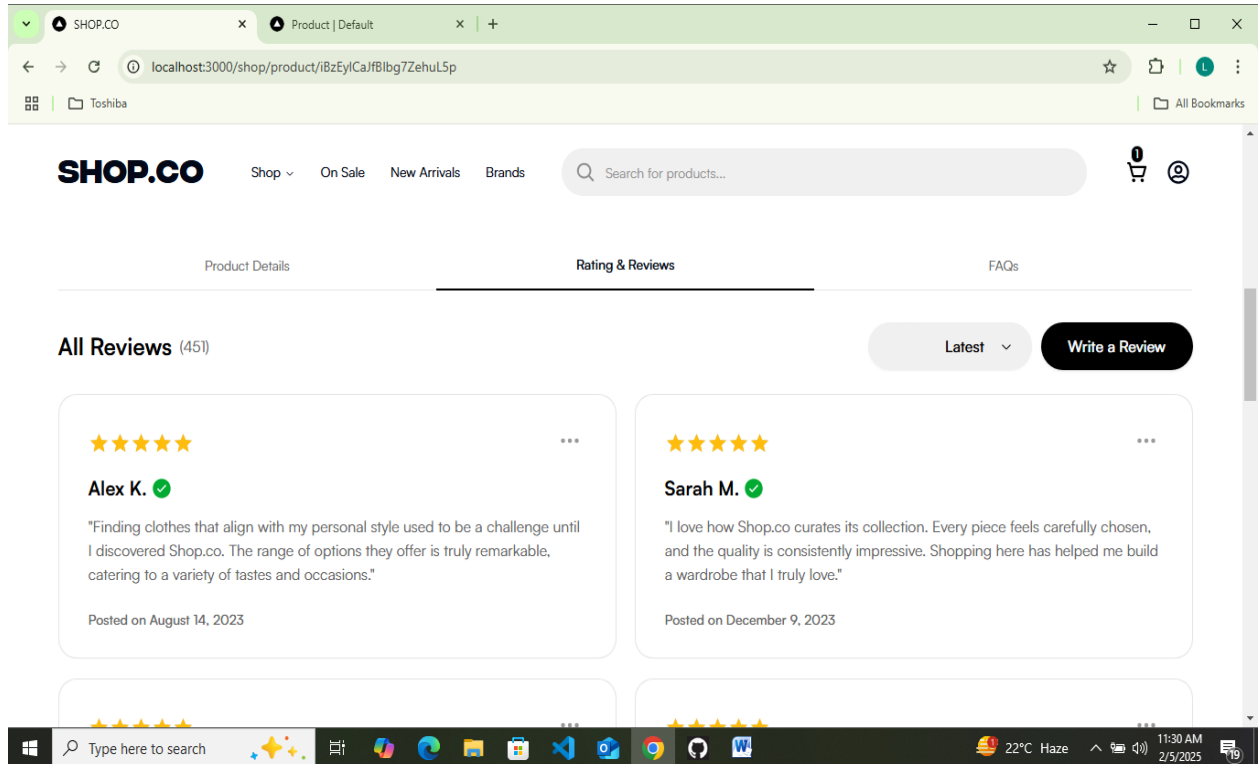
Product Detail



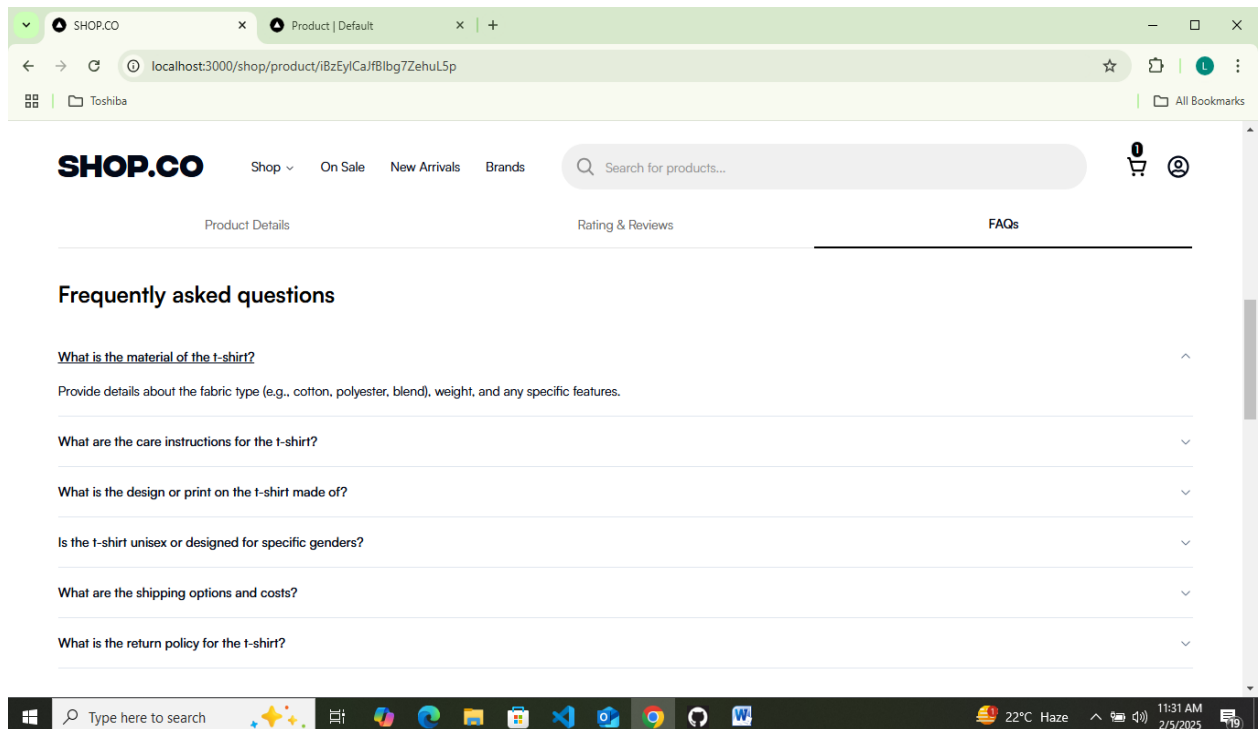
Product Specifications



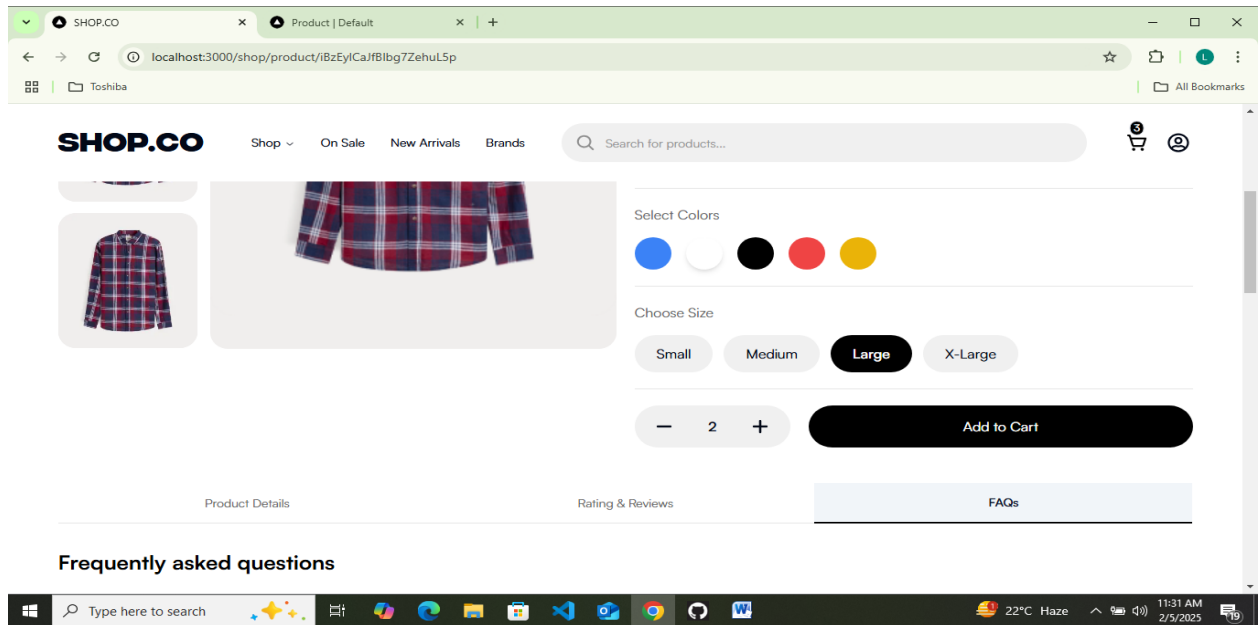
Reviews



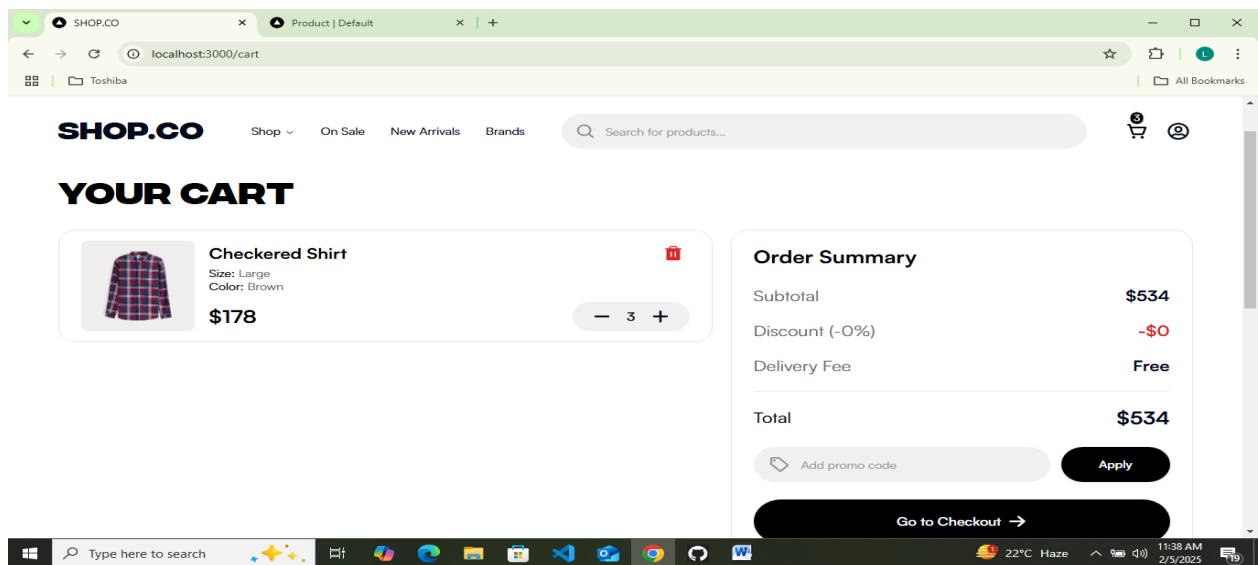
Faqs



Add To Cart



Cart



Checkout

