

Name: Nimra Ulfat

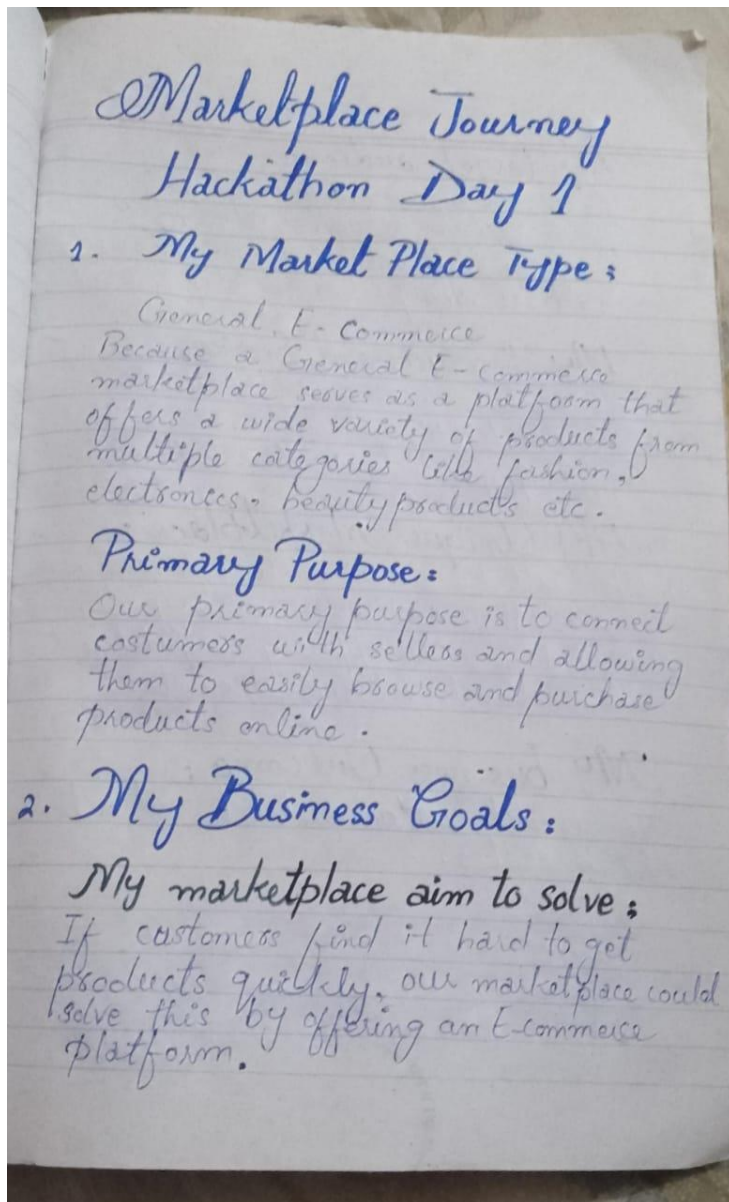
ID: 485961

Template: 1

Name: SHOP.CO

Hackathon 3 Marketplace Builder

DAY 1: LAYING THE FOUNDATION FOR MARKETPLACE JOURNEY



My target audience :
Both young and adults primarily
from urban areas. Working
professionals, student and home-
makers who prefer the shopping online.

My Products and Services :
Products: clothing and Fashion.
Our marketplace could offer a wide
range of products similar to Amazon.

My Unique Marketplace :
We will be ensuring that our customers
receive their purchases quickly also
we will focus on offering competitive
prices and exclusive discounts.

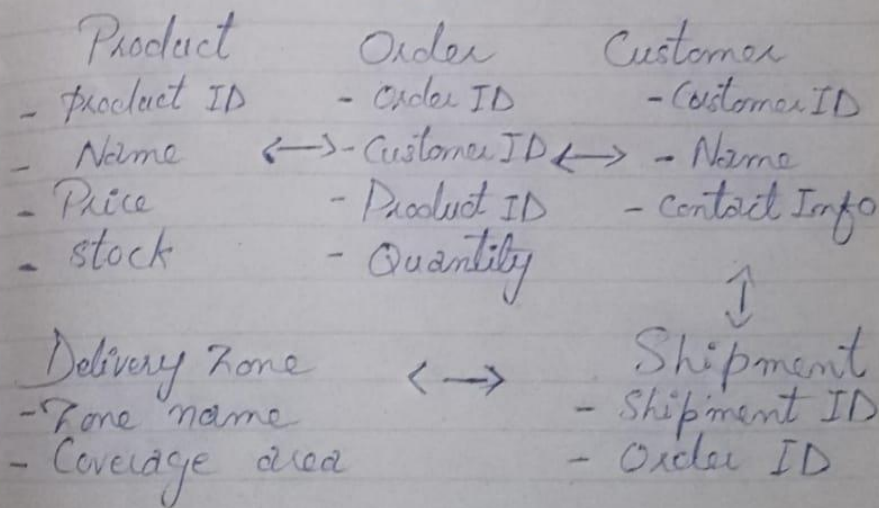
My business Outcome :
To build customers trust and achieve
high scales.

3. Data Schema:

Entities:

- * Products: items available for sale.
- * Orders: records of transactions between customers and our platform.
- * Customers: individuals or businesses purchasing
- * Delivery Zone: areas covered by service providers.
- * Shipment: Tracks the movement of items from warehouse to customer.
- * Payments: Tracks transactions and payment statuses.

Explain by flowchart:



Hackathon 3 Marketplace Builder

Day 2 - Planning The Technical Foundation

My Business Goals

Frontend

- Ensure users can easily explore products and access essential products information.
- Show detailed product information including images, description and price.
- Ensure the platform is fully responsive and mobile-friendly.
- Provide pages that users expect in an ecommerce experience.

Sanity CMS Backend

- Ensure that Sanity CMS acts as the backend for managing essential data like products, order and customers.
- Create structured schemas that align with the business needs of my platform.

Third Party APIs

- Integrate with Shipment service to track orders and provide real-time shipping updates.
- Integrate a payment gateway to handle secure transactions.

Designing System Architecture

User

Frontend (Next.js)

Sanity CMS

(Product Data API)

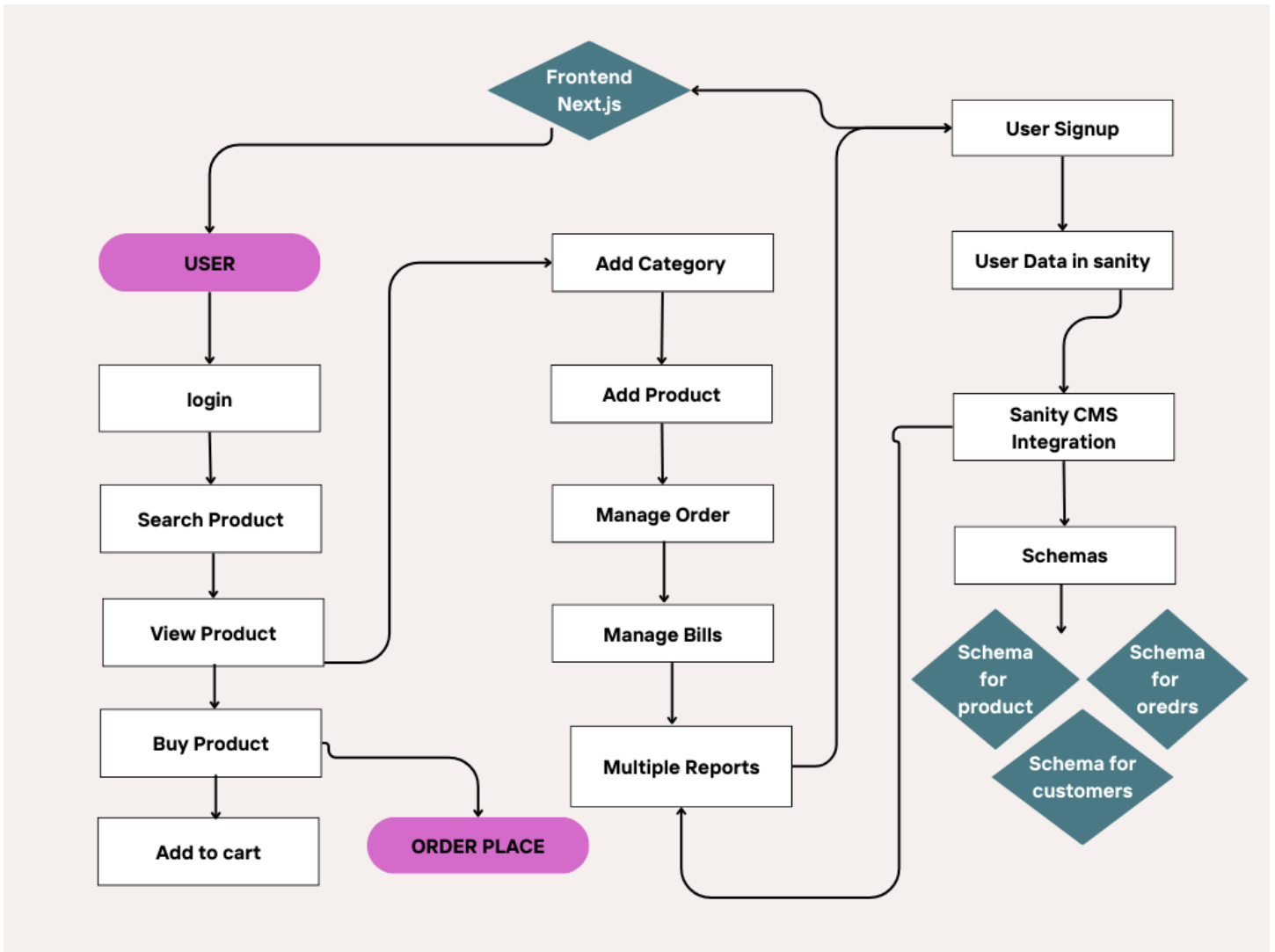
Third Party APIs

Payment Gateway

Shipment Tracking API

Planning API Requirements

End Point	Method	Description	Response
Product	GET	Fetch all available product details	[{"id": 1, "name": "product A", "price": 100, "stock": 50, "image": "url"}]
Orders	POST	Create a new order in sanity	[{"orderId": 1234, "status": "order placed"}]
Shipment	GET	Track order status via third party API	[{"shipmentId": "SHIP1234", "orderId": 1234, "status": "In Transit"}]
Payment	POST	Process payment details via third-part API	[{"paymentStatus": "success", "paymentTransactionId": "TXN1234"}]



Hackathon 3 Marketplace Builder

Day 3 - API Integration and Data Migration

Integrate APIs

This document provides detailed of the API Integration process, Schema adjustment and Data Migration into Sanity CMS.

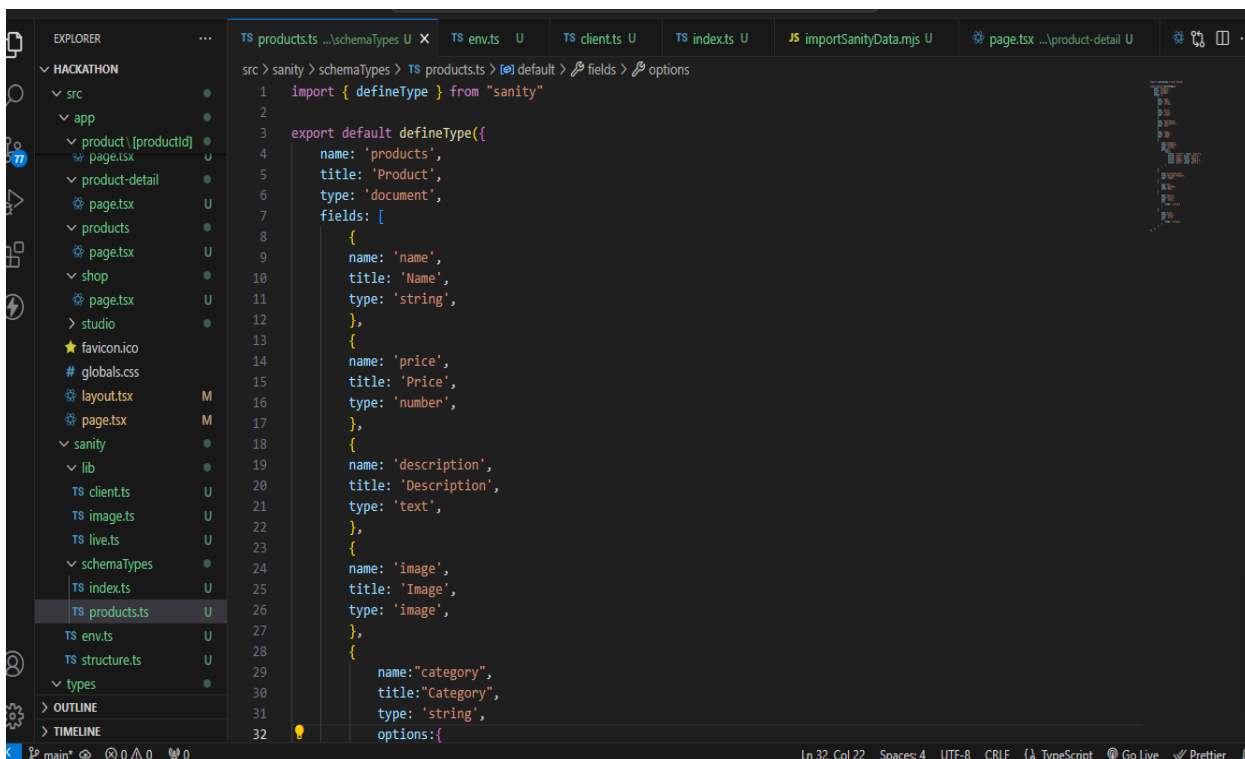
1. API Integration Process

The API Integration process involved fetching external products and data and populating it into Sanity CMS.

2. Validation and Adjust Schema

Adjust schema to match or map fields during migration.

- **name:** String field to store the product name.
- **price:** Number field to store the product price.
- **description:** Text field to store a detailed description.
- **image:** Image field to upload product images.

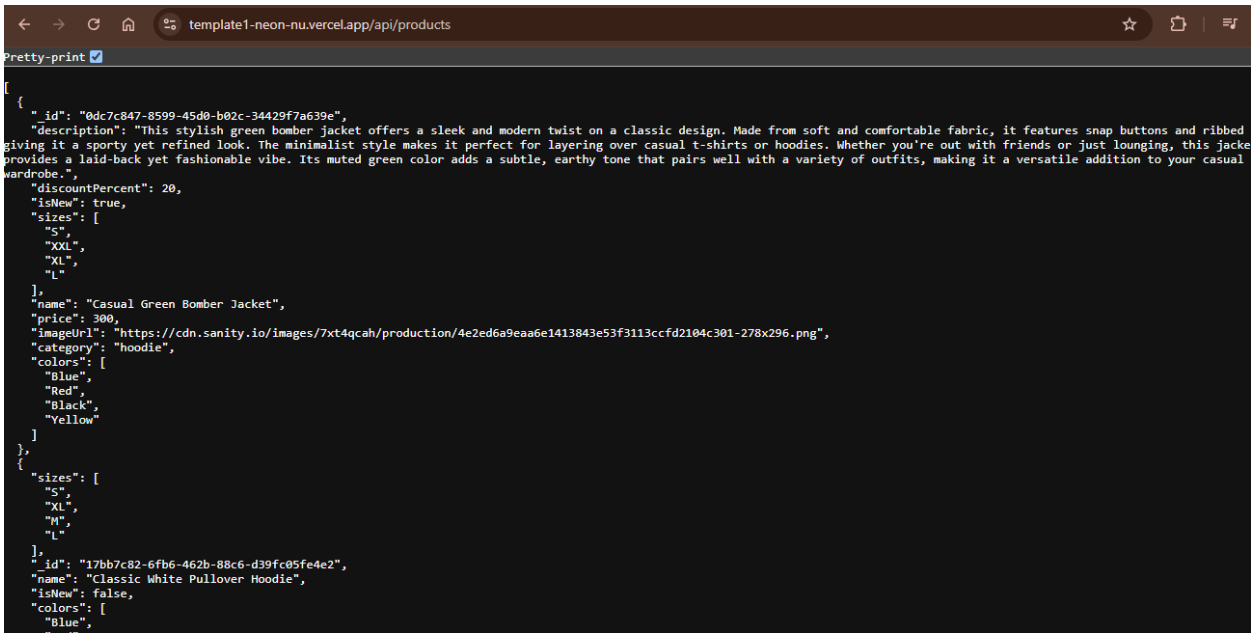


```
src > sanity > schemaTypes > TS products.ts > default > fields > options
1  import { defineType } from "sanity"
2
3  export default defineType({
4    name: 'products',
5    title: 'Product',
6    type: 'document',
7    fields: [
8      {
9        name: 'name',
10       title: 'Name',
11       type: 'string',
12     },
13     {
14       name: 'price',
15       title: 'Price',
16       type: 'number',
17     },
18     {
19       name: 'description',
20       title: 'Description',
21       type: 'text',
22     },
23     {
24       name: 'image',
25       title: 'Image',
26       type: 'image',
27     },
28     {
29       name: 'category',
30       title: 'Category',
31       type: 'string',
32     },
33   ],
34   options: {
```

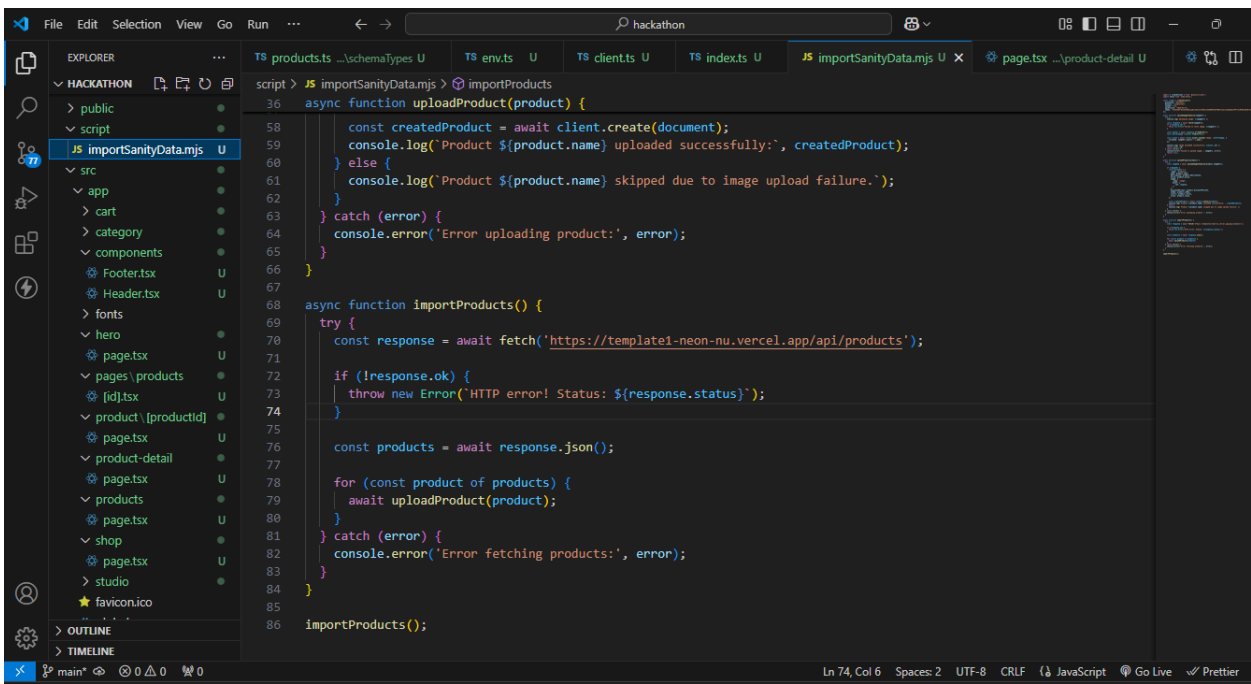
3. Data Migration

Using the Provided API

Use the given API according to my template 1 for fetching data.

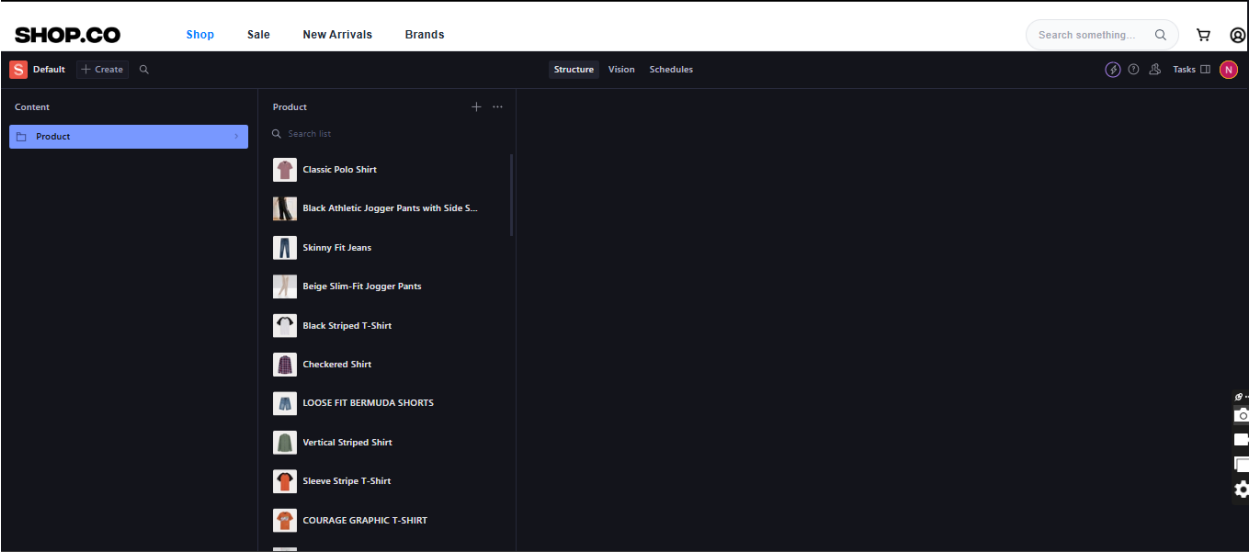
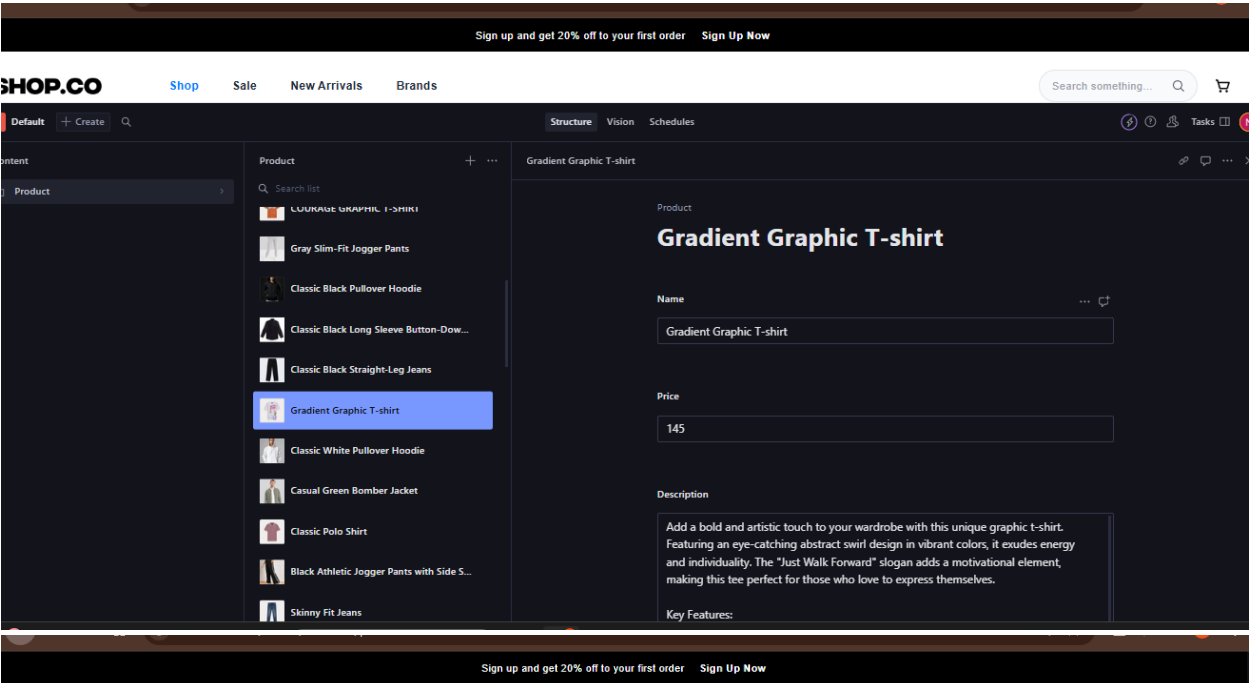


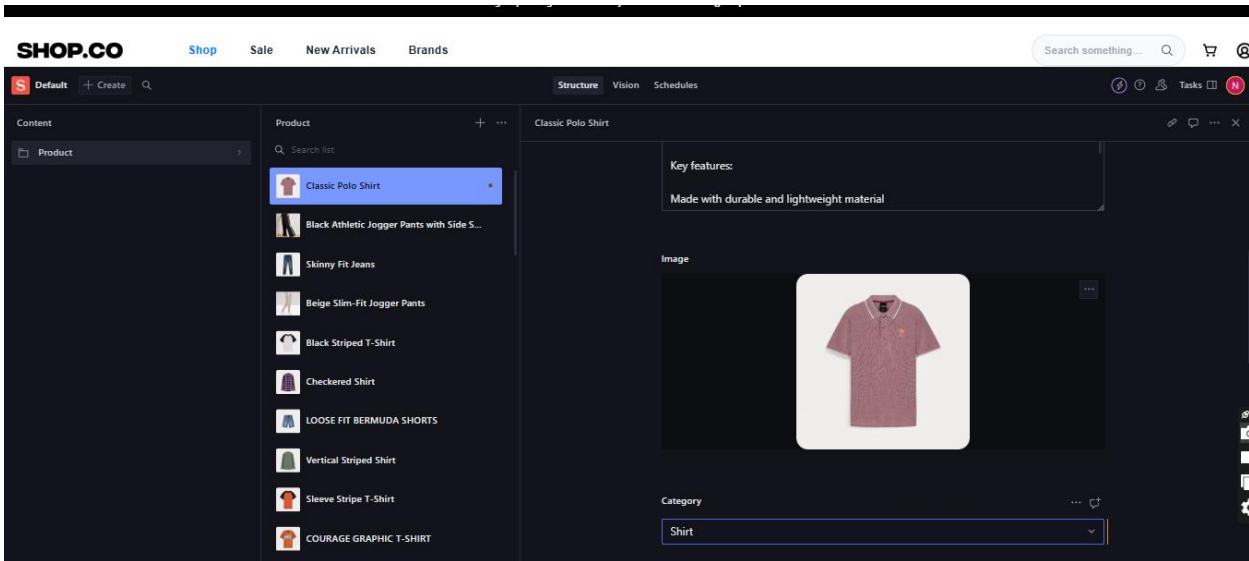
Migration Script



Manual Import Data

Export data from the API by using Sanity.





Hackathon 3 Marketplace Builder

Day 4 – Building Dynamic Frontend Components For Marketplace

Day 3 Recap

On Day 3, I focused on API integration and data migration to prepare my website for dynamic data handling.

Familiarized myself with the API documentation and the provided endpoints.

Tested the endpoints using Postman

Designed and implemented content models in Sanity CMS to store relevant data.

Migrated data from external sources APIs to Sanity CMS.

Implemented error handling and response parsing to ensure robustness in API integrations.

Day 4 - Building Dynamic Frontend Components for Marketplace

Building Dynamic Frontend Components to Display Data from Sanity CMS or APIs

- Developed dynamic frontend components that fetch and display data from Sanity CMS or external APIs.

The screenshot shows a VS Code editor with the file explorer on the left. The project structure includes a 'pages/products' directory. The main editor displays the code for 'page.tsx' in the 'products' directory. The code defines a 'ProductCards' component using React.FC and uses 'useState' for 'products' and 'cart'. It also includes an async 'fetchProducts' function that uses the 'sanity' client to fetch product data from a CMS. The code is as follows:

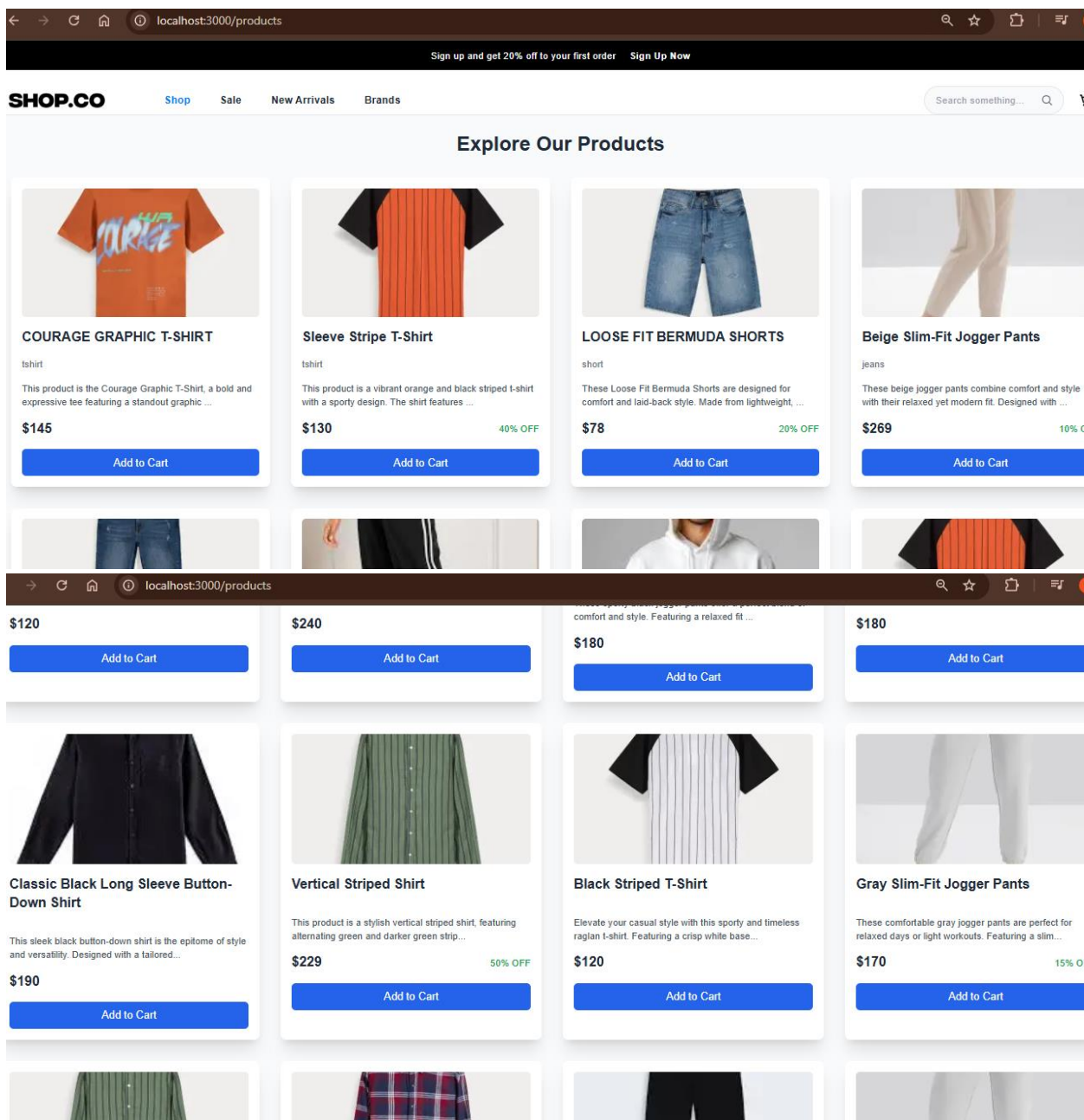
```
const ProductCards: React.FC = () => {  
  const [products, setProducts] = useState<Product[]>([]);  
  const [cart, setCart] = useState<Product[]>([]);  
  
  const fetchProducts = async () => {  
    try {  
      const query = `*_type == "products"{  
        _id,  
        name,  
        price,  
        description,  
        image {  
          asset -> {  
            url  
          }  
        },  
        category,  
        discountPercent,  
        new,  
        colors,  
        sizes  
      }`;  
      const data = await sanity.fetch(query);  
      const productsWithImageUrl = data.map((product: any) => ({  
        ...product,  
        imageUrl: product.image?.asset?.url || "",  
      }));  
      setProducts(productsWithImageUrl);  
    } catch (error) {  
      console.error("Error fetching Products:", error);  
    }  
  }  
}
```

- Used **React** for building components.
- **Sanity client** for retrieving data from Sanity CMS.
- Created components that fetch data dynamically

The screenshot shows a VS Code editor with the file explorer on the left. The main editor displays the code for 'importSanityData.mjs'. The code includes an async function 'uploadProduct' and an async function 'importProducts'. The 'importProducts' function uses 'fetch' to retrieve product data from a Vercel API and then calls 'uploadProduct' for each product. The code is as follows:

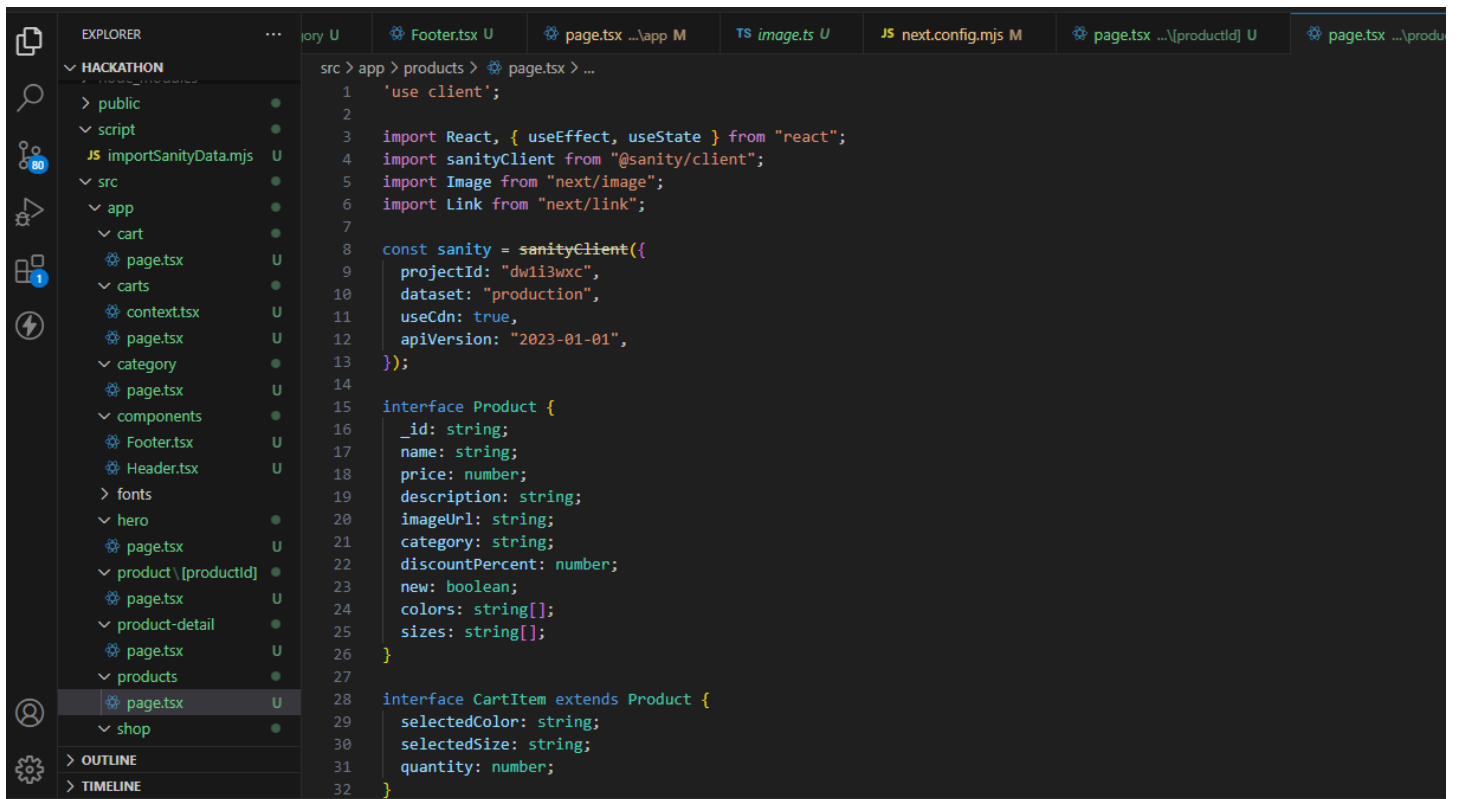
```
async function uploadProduct(product) {  
  // ...  
} else {  
  console.log(`Product ${product.name} skipped due to image upload failure.`);  
}  
} catch (error) {  
  console.error("Error uploading product:", error);  
}  
}  
  
async function importProducts() {  
  try {  
    const response = await fetch('https://template1-neon-nu.vercel.app/api/products');  
  
    if (!response.ok) {  
      throw new Error(`HTTP error! Status: ${response.status}`);  
    }  
  
    const products = await response.json();  
  
    for (const product of products) {  
      await uploadProduct(product);  
    }  
  } catch (error) {  
    console.error("Error fetching products:", error);  
  }  
}  
  
importProducts();
```

Frontend components to display data from Sanity CMS



Product Listing Component

Rendered product data dynamically.



```
src > app > products > page.tsx > ...
1  'use client';
2
3  import React, { useEffect, useState } from "react";
4  import sanityClient from "@sanity/client";
5  import Image from "next/image";
6  import Link from "next/link";
7
8  const sanity = sanityClient({
9    projectId: "dwii3wxc",
10   dataset: "production",
11   useCdn: true,
12   apiVersion: "2023-01-01",
13 });
14
15 interface Product {
16   _id: string;
17   name: string;
18   price: number;
19   description: string;
20   imageUrl: string;
21   category: string;
22   discountPercent: number;
23   new: boolean;
24   colors: string[];
25   sizes: string[];
26 }
27
28 interface CartItem extends Product {
29   selectedColor: string;
30   selectedSize: string;
31   quantity: number;
32 }
```

Product Detail Component

Created individual product detail pages using dynamic routing in Next.js

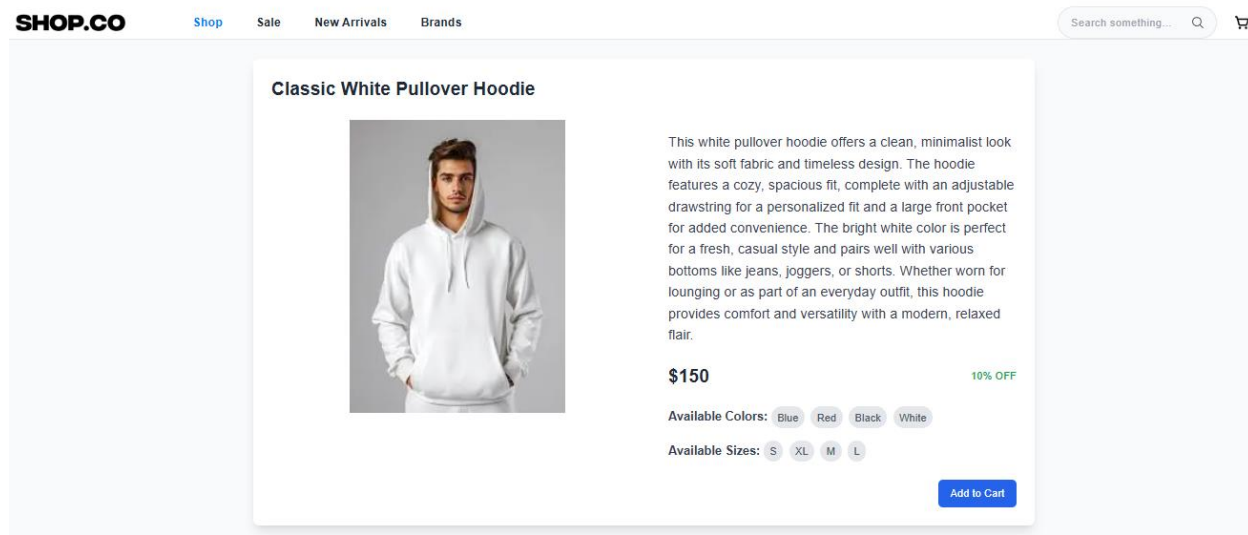
```
const ProductDetailPage = () => {
  const { productId } = useParams();
  const [product, setProduct] = useState<Product | null>(null);
  const [loading, setLoading] = useState(true);

  const fetchProduct = async (id: string) => {
    try {
      const query = `[_type == "products" && _id == "${id}"]{
        _id,
        name,
        price,
        description,
        image {
          asset -> {
            url
          }
        },
        category,
        discountPercent,
        new,
        colors,
        sizes
      }`;

      const data = await sanity.fetch(query);
      if (data.length > 0) {
        const productWithImageUrl = {
          ...data[0],
          imageUrl: data[0].image?.asset?.url || '',
        };
      }
    }
  };
}
```

Category Component

Displayed categories dynamically fetched from the data source.



Black Striped T-Shirt



Elevate your casual style with this sporty and timeless raglan t-shirt. Featuring a crisp white base adorned with vertical pinstripes, it blends a retro vibe with modern appeal. The contrasting black raglan sleeves add a bold, athletic touch, making it a versatile piece for any wardrobe. Key Features: Soft and breathable fabric for all-day comfort Classic pinstripe design for a clean, stylish look Contrasting raglan sleeves for a sporty edge Regular fit with a crew neckline Perfect for casual outings or active days, pair this tee with your favorite jeans or joggers for effortless

\$120

Available Colors: Blue White Black Red

Available Sizes: M L S XXL

Add to Cart



Casual Green Bomber Jacket

This stylish green bomber jacket offers a sleek and modern twist on a classic design. Made from soft...

\$300

20% OFF

Add to Cart



Classic White Pullover

This white pullover hoodie offers a timeless design with its soft fabric and timeless des...

\$150

Add to Cart



Casual Green Bomber Jacket



This stylish green bomber jacket offers a sleek and modern twist on a classic design. Made from soft and comfortable fabric, it features snap buttons and ribbed cuffs, giving it a sporty yet refined look. The minimalist style makes it perfect for layering over casual t-shirts or hoodies. Whether you're out with friends or just lounging, this jacket provides a laid-back yet fashionable vibe. Its muted green color adds a subtle, earthy tone that pairs well with a variety of outfits, making it a versatile addition to your casual wardrobe.

\$300 20% OFF

Available Colors: Blue Red Black Yellow

Available Sizes: S XXL XL L

Add to Cart

Search Bar:

Find Clothes That Matches Your Style

Browse through our diverse ranges of meticulously crafted garments, designed to bring out your individuality and cater to your sense of style.

Shop Now



Cart Component

localhost:3000 says

LOOSE FIT BERMUDA SHORTS has been added to your cart!

OK



Sleeve Stripe T-Shirt

tshirt

This product is a vibrant orange and black striped t-shirt with a sporty design. The shirt features ...

Red

S

\$130

40% OFF

Add to Cart



LOOSE FIT BERMUDA SHORTS

short

These Loose Fit Bermuda Shorts are designed for comfort and laid-back style. Made from lightweight, ...

Red

L


\$78

20% OFF

Add to Cart

Display added items, quantity, and total price.

Your Cart



Gradient Graphic T-shirt

Size: Large


Color: White

-

2

+

\$240



CHECKERED SHIRT

Size: Medium


Color: Red

-

1

+

\$70



SKINNY FIT JEANS

Size: Large

Color: Blue

-

1

+

\$20

Order Summary

Subtotal

\$330

Shipping

\$2.00

Tax

\$4.00

Total

\$336

Checkout

Continue Shopping

Filter Panel

SHOP.CO

[Shop](#) [Sale](#) [New Arrivals](#) [Brands](#)

Filters

T-Shirt Names

Gradient Graphic T-shirt

Polo with Tipping Details

Black Striped T-shirt

Skinny Fit Jeans

Checkered Shirt

Sleeve Striped T-shirt

Price Range

\$0


\$500

Color

Size

Select Size

Dress Style





Gradient Graphic T-shirt

★★★★★ 4.5/5

\$145

In Stock







Polo with Tipping Details

★★★★ 3.5/5

\$180

Out of Stock






Black Striped T-shirt

★★★★★ 4.5/5

\$120

In Stock



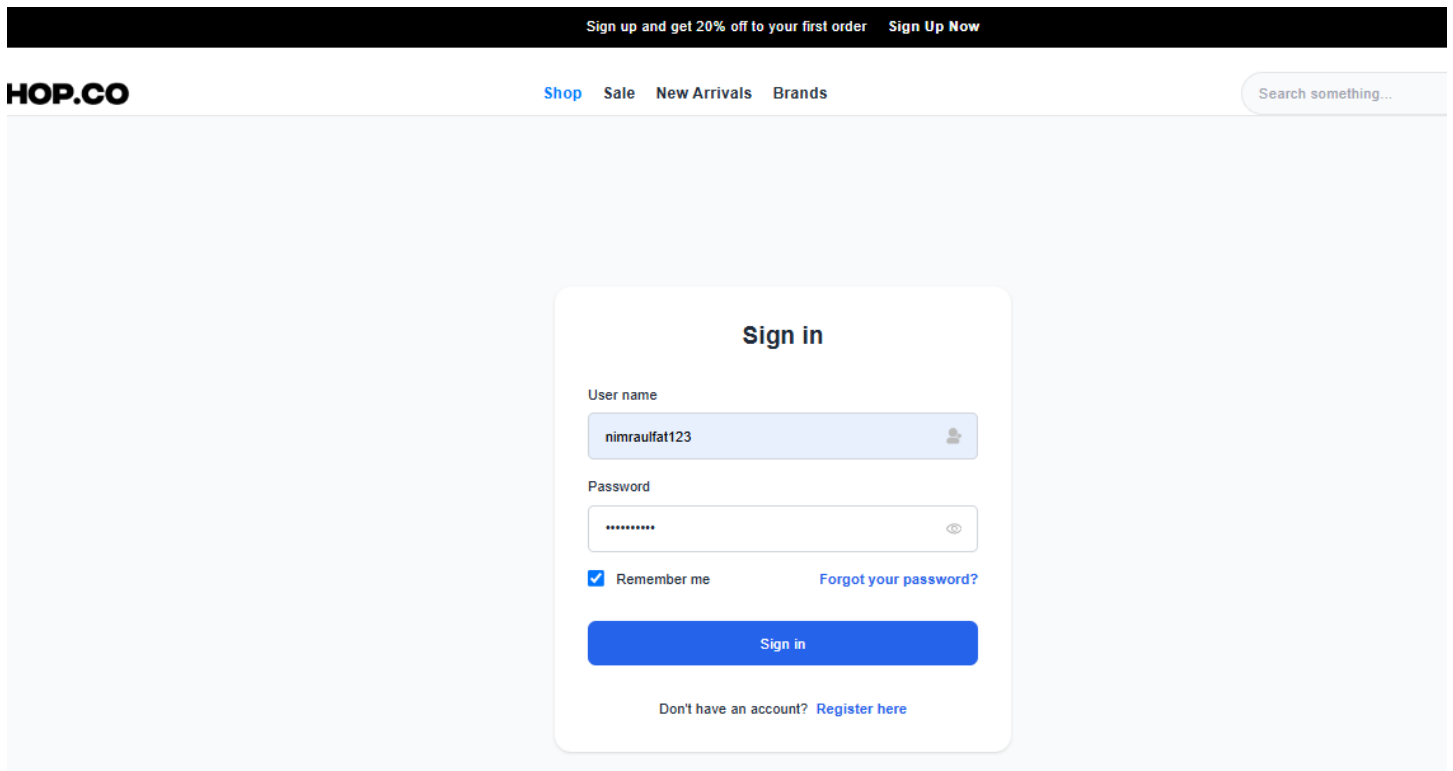
Skinny Fit Jeans

★★★★★ 4.5/5

\$240

In Stock

Sign In Page



Hackathon 3 Marketplace Builder

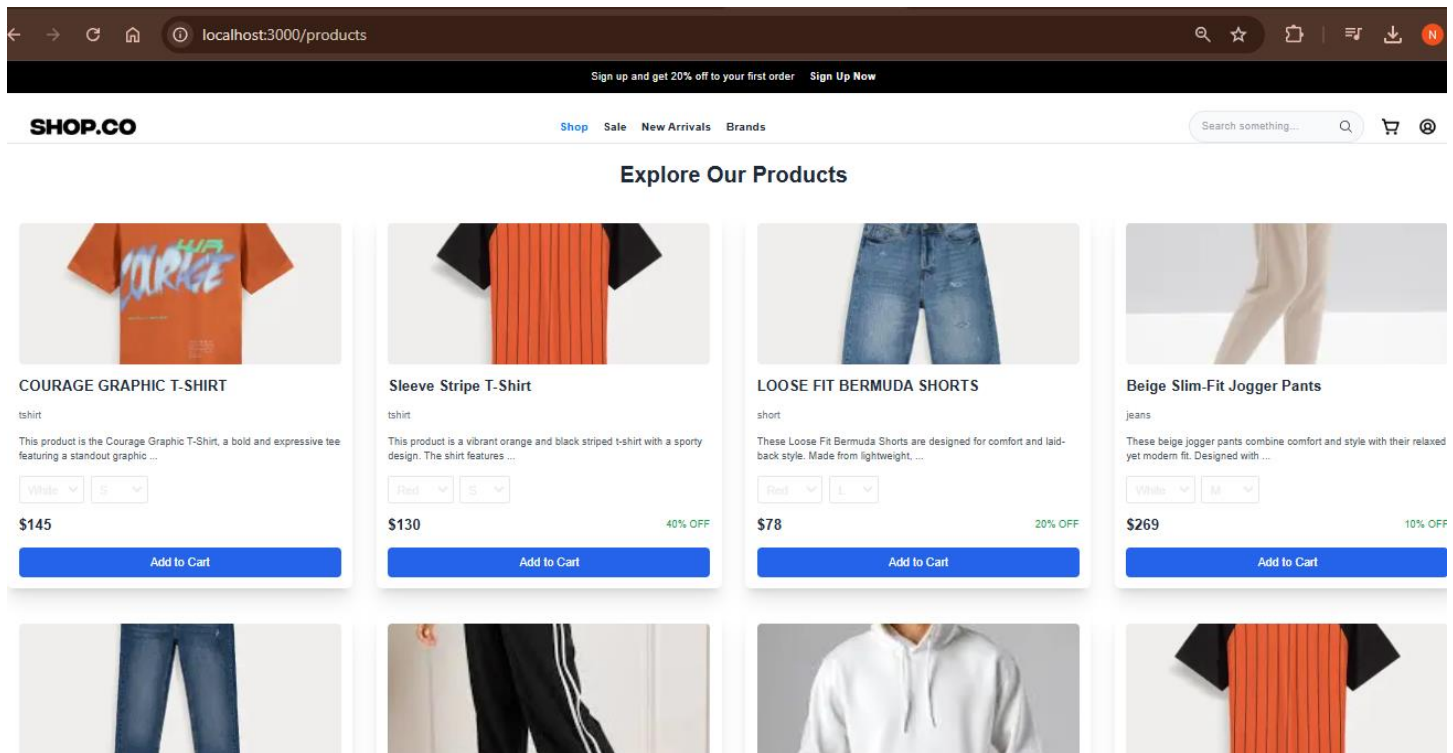
Day 5 - Testing, Error Handling, and Backend Integration Refinement

Functional Testing

All marketplace features have been thoroughly tested and validated to ensure they are functioning as intended.

Product listing

Products are displayed correctly



Cart operations

Add, update, and remove items from the cart working perfectly

localhost:3000 says

LOOSE FIT BERMUDA SHORTS has been added to your cart!

OK



Sleeve Stripe T-Shirt

tshirt

This product is a vibrant orange and black striped t-shirt with a sporty design. The shirt features ...

Red S

\$130

40% OFF

Add to Cart



LOOSE FIT BERMUDA SHORTS

short

These Loose Fit Bermuda Shorts are designed for comfort and laid-back style. Made from lightweight, ...

Red L

\$78

20% OFF

Add to Cart

Dynamic routing

Verified individual product detail pages load correctly

Classic White Pullover Hoodie



This white pullover hoodie offers a clean, minimalist look with its soft fabric and timeless design. The hoodie features a cozy, spacious fit, complete with an adjustable drawstring for a personalized fit and a large front pocket for added convenience. The bright white color is perfect for a fresh, casual style and pairs well with various bottoms like jeans, joggers, or shorts. Whether worn for lounging or as part of an everyday outfit, this hoodie provides comfort and versatility with a modern, relaxed flair.

\$150

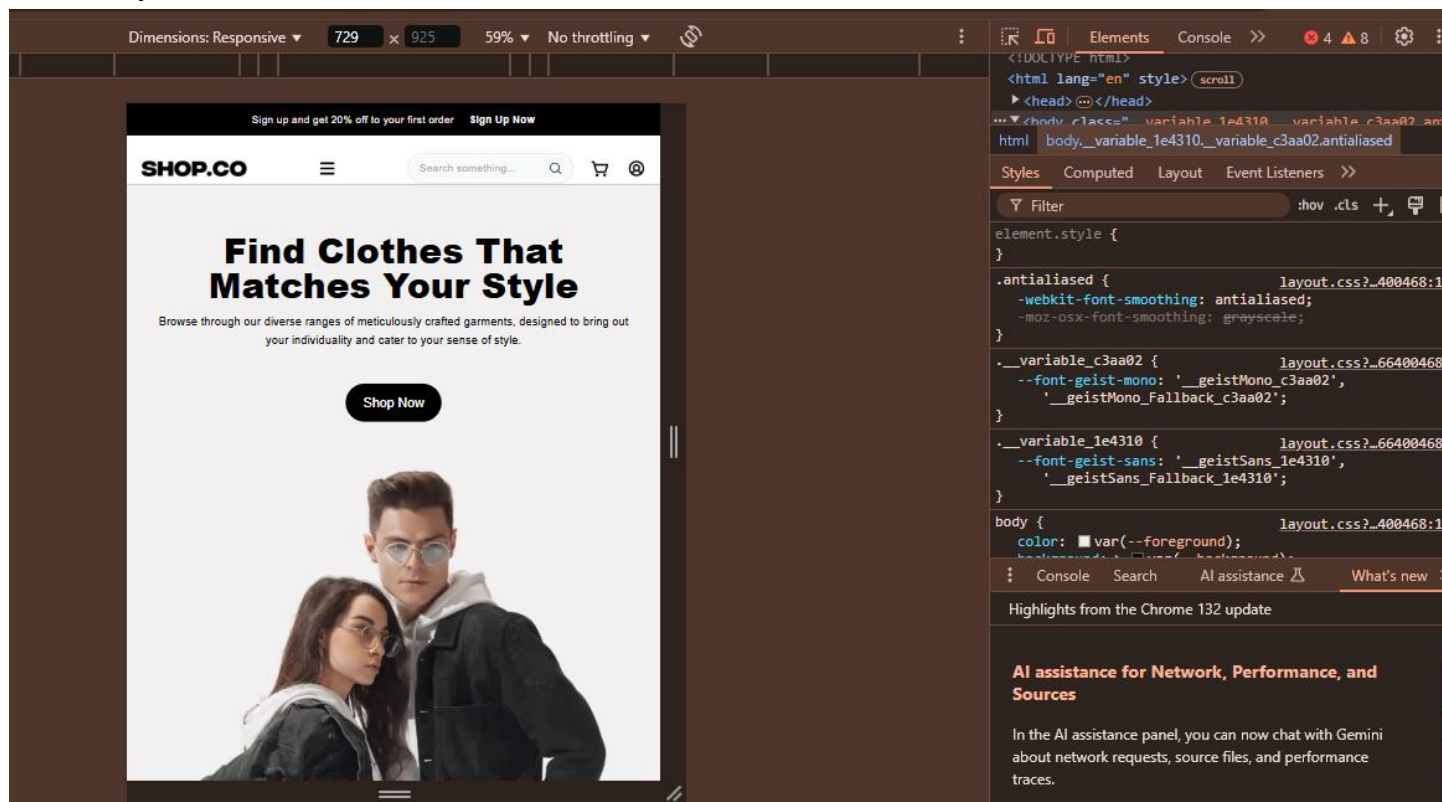
10% OFF

Available Colors: Blue Red Black WhiteAvailable Sizes: S XL M L

Add to Cart

Responsive

Responsive design testing has been completed to ensure the marketplace is fully optimized and functions seamlessly across all devices and screen sizes



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

SHOP.CO



Search something...



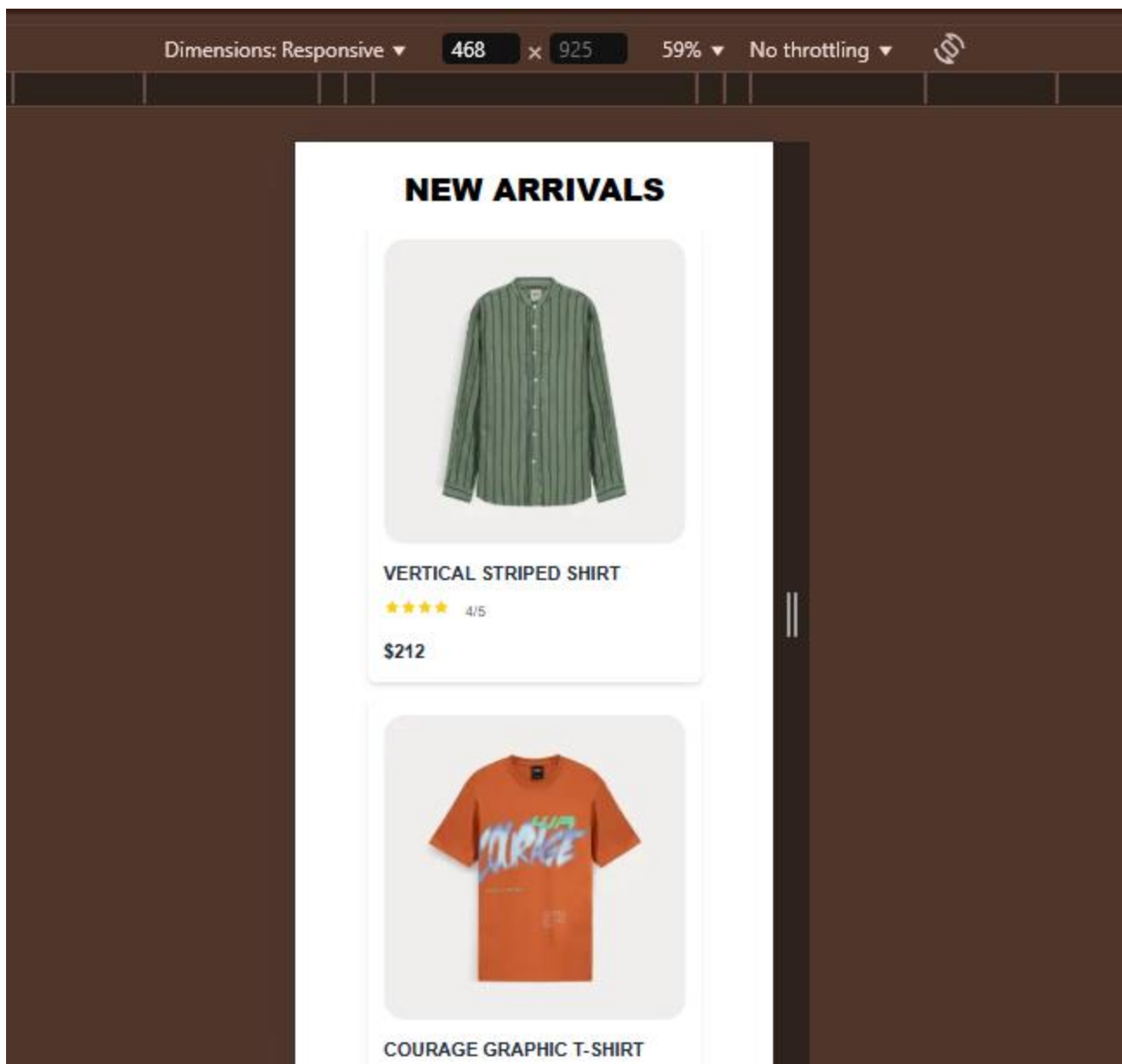
Find Clothes That Matches Your Style

Browse through our diverse ranges of meticulously crafted garments, designed to bring out your individuality and cater to your sense of style.

[Shop Now](#)



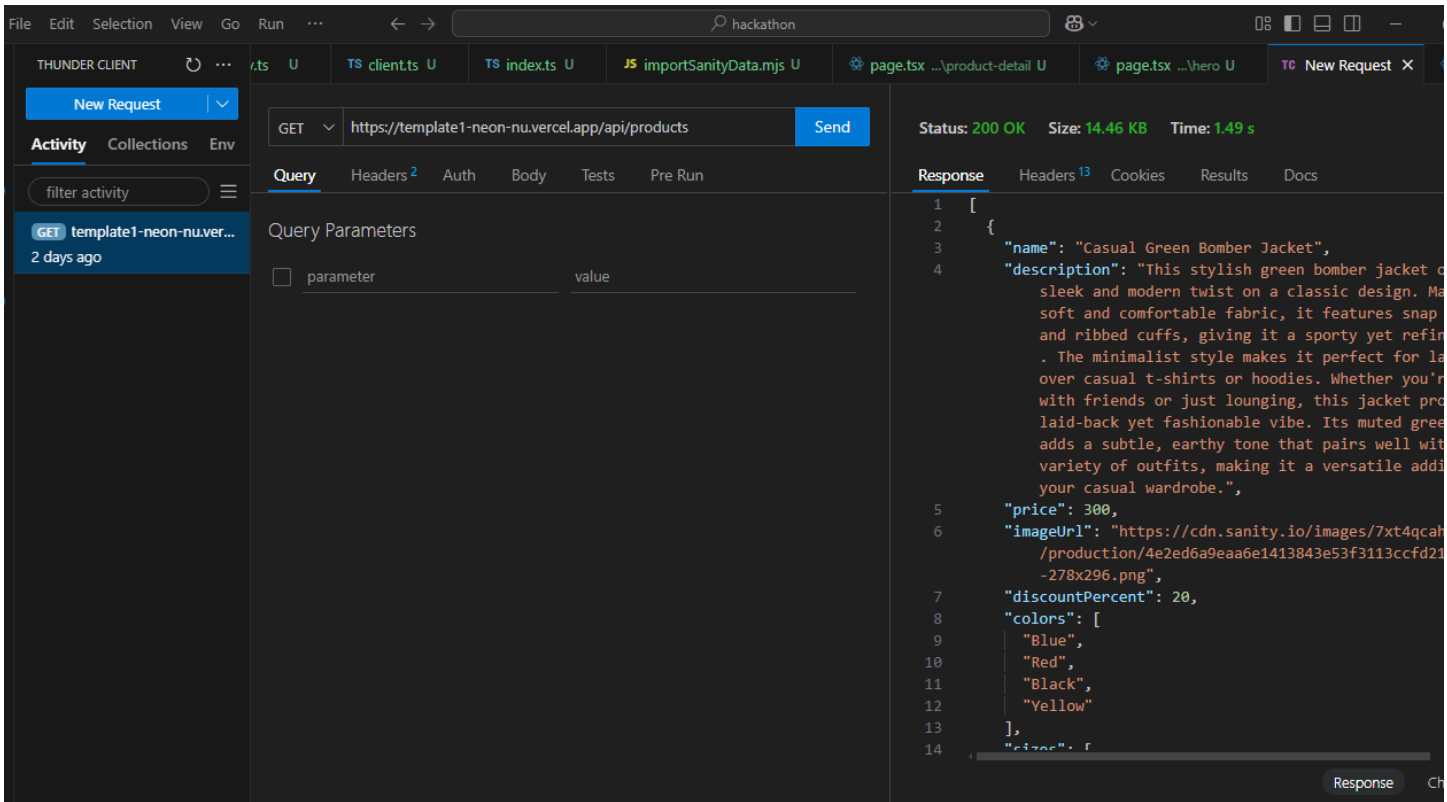
VERSACE ZARA



Testing Tools

Postman

API responses have been tested using Postman to ensure they meet the expected behavior and performance standards



Performance Optimization

Testing Report

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks	
2	TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	high	-	No issues found
3	TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	-	Handled gracefully
4	TC003	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	High	-	No issue found
5	TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	Medium	-	Test successful

67

78

96

91



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0–49 ■ 50–89 ● 90–100

METRICS

[Expand view](#)

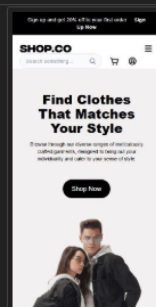
● First Contentful Paint
1.1 s

▲ Total Blocking Time
3,040 ms

● Speed Index

■ Largest Contentful Paint
2.7 s

● Cumulative Layout Shift
0.037





PASSED AUDITS (22)



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so [manual testing](#) is also encouraged.



NOT APPLICABLE (33)

Show



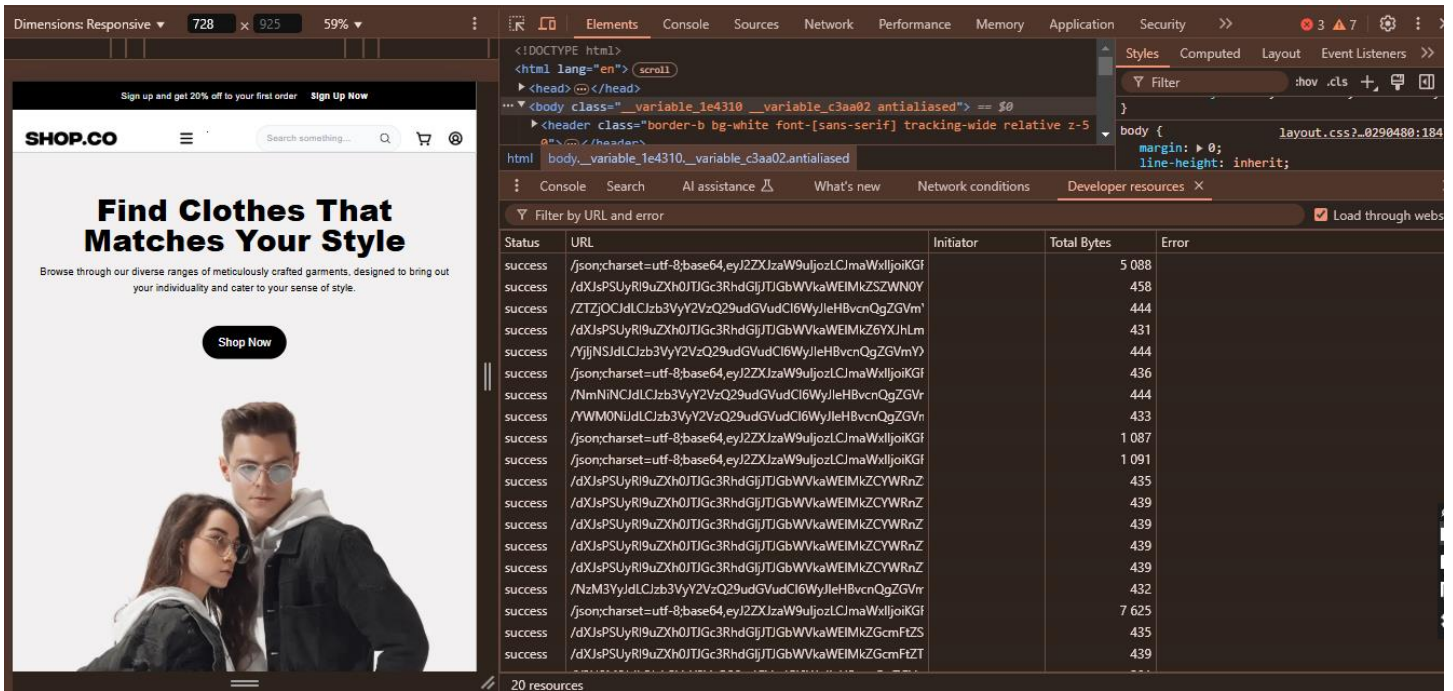
Best Practices



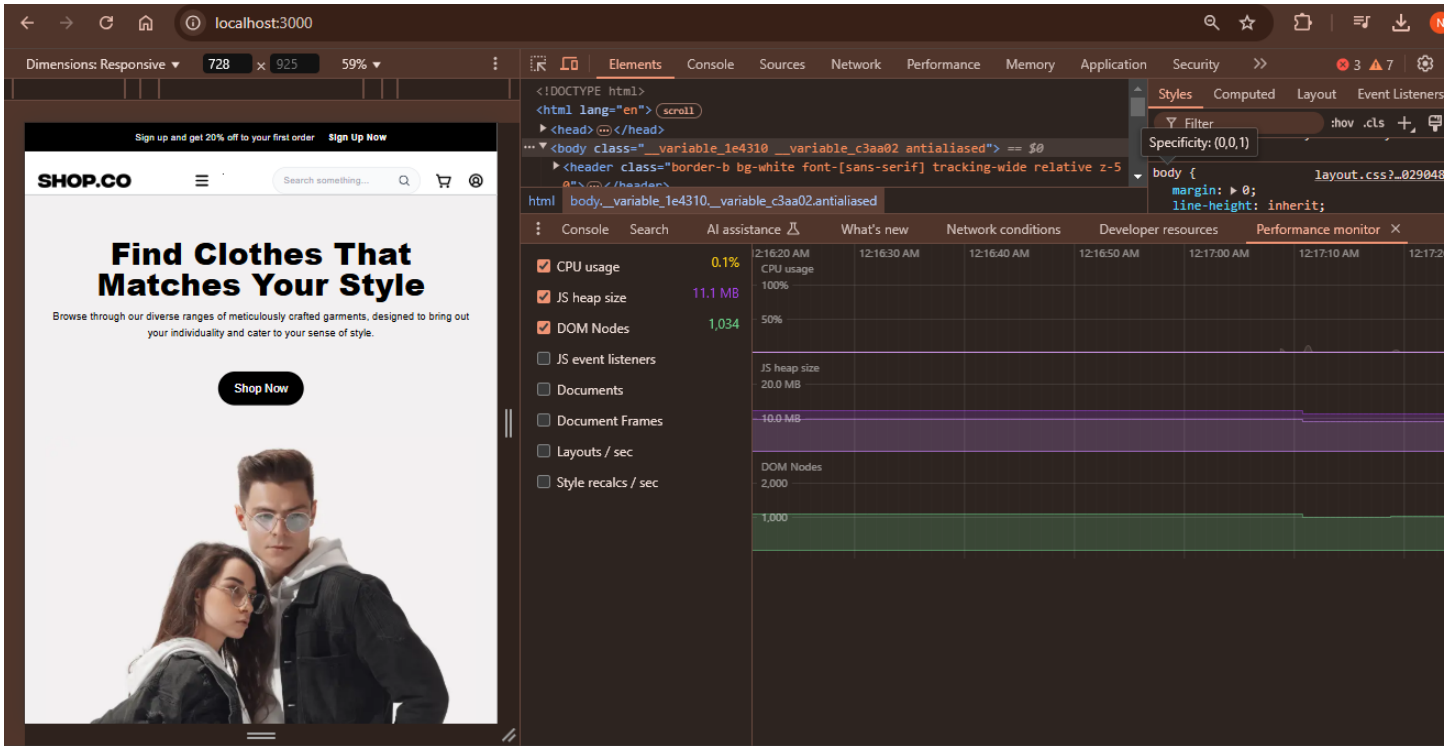
SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more about Google Search Essentials](#).

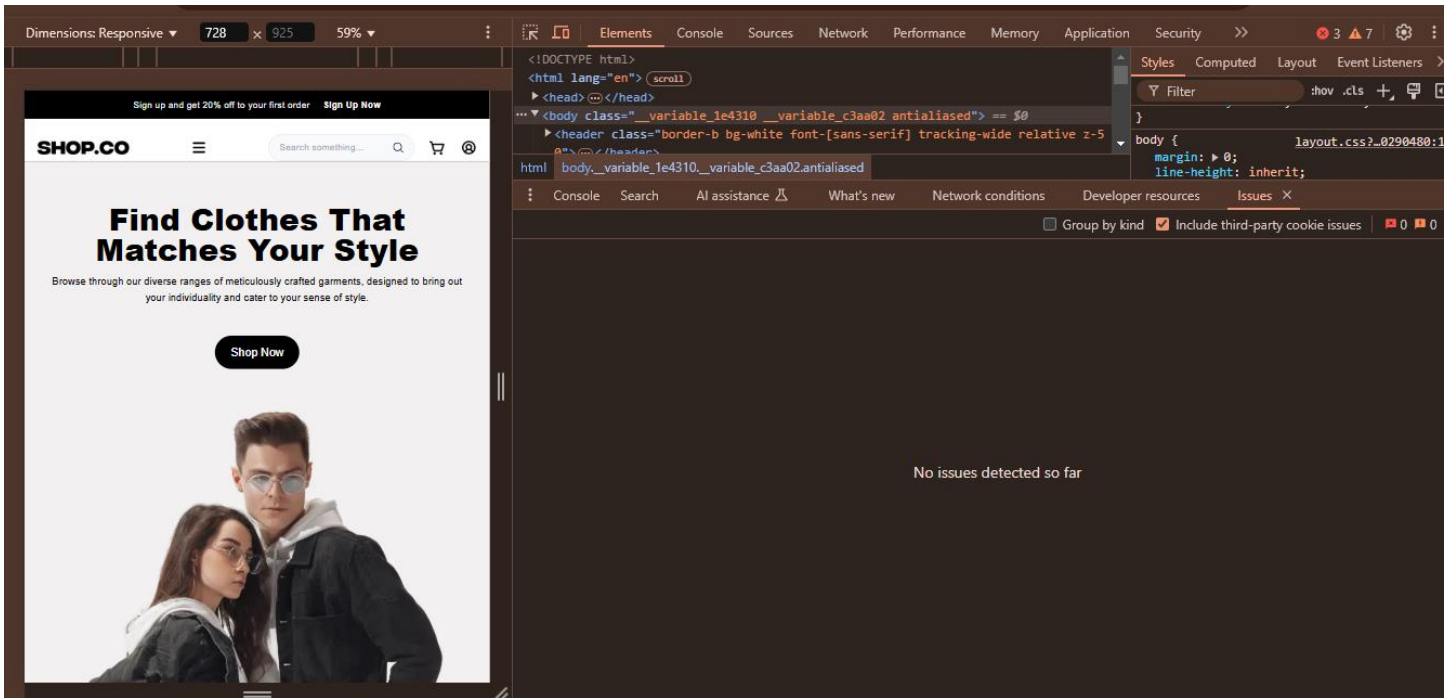
Developer Resources



Performance Monitor



Issues



Security

