AMEER
HAMZA

↘

ABOUT

Name: Ameer Hamza
Roll Number: 00054341
Email: ahm401613@gmail.com

Day 3 Hackathon

API Integration and Data Migration Process

Objectives:

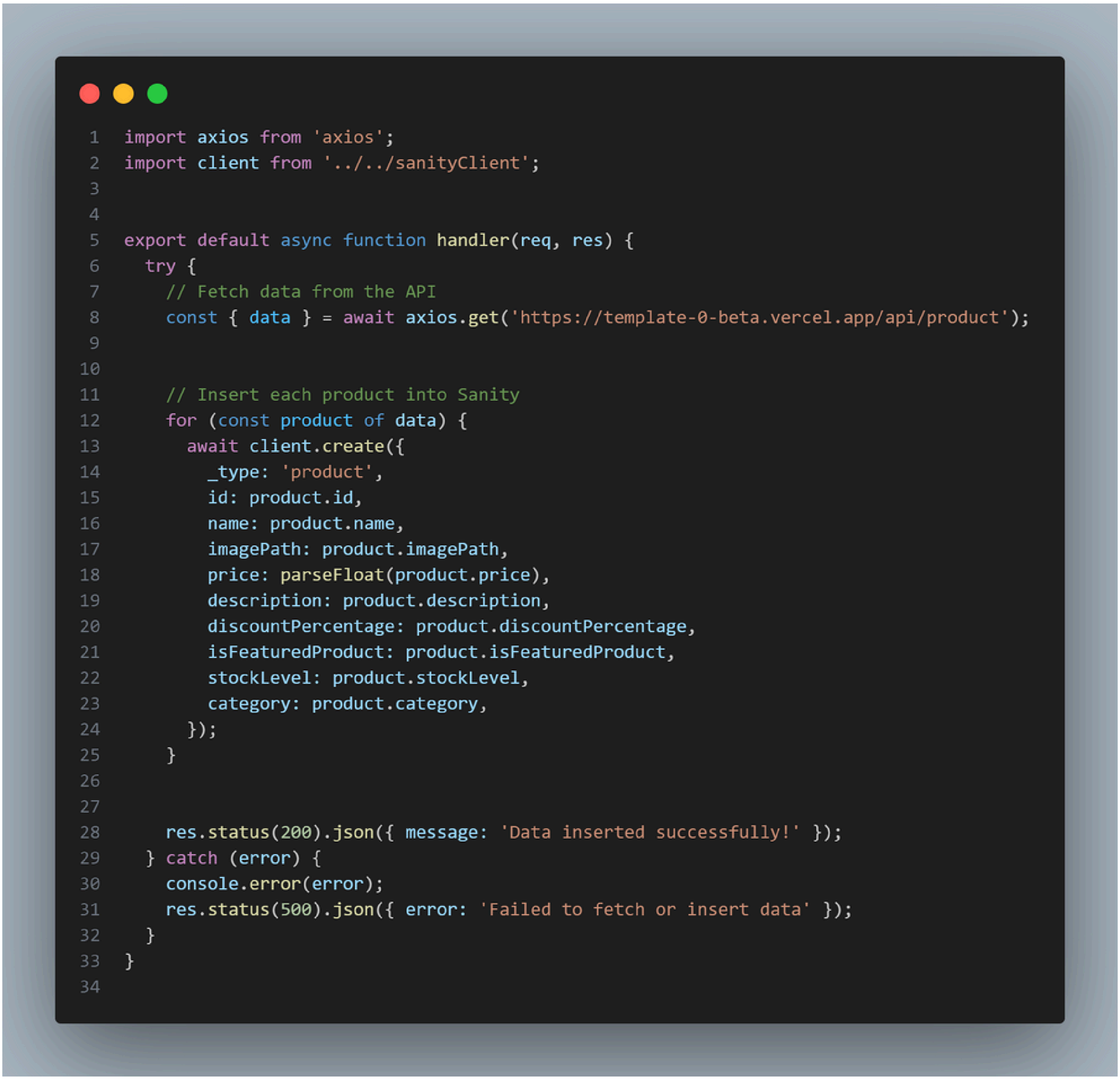
- Set up API integration in a Next.js app.
- Move data from external APIs into Sanity CMS.
- Make sure the schemas are validated and properly matched with the data sources.

Key Takeaways:

- Learn how to implement API integrations.
- Understand how to transfer data from APIs to a CMS.
- Customize and validate schemas in Sanity CMS to ensure smooth data compatibility.

Custom Data Migration Script

This script allows you to easily transfer data from Sanity CMS to an E-commerce Marketplace database. The process follows these main steps:



```
import axios from 'axios';
import client from '../../sanityClient';


export default async function handler(req, res) {
  try {
    // Fetch data from the API
    const { data } = await axios.get('https://template-0-beta.vercel.app/api/product');


    // Insert each product into Sanity
    for (const product of data) {
      await client.create({
        _type: 'product',
        id: product.id,
        name: product.name,
        imagePath: product.imagePath,
        price: parseFloat(product.price),
        description: product.description,
        discountPercentage: product.discountPercentage,
        isFeaturedProduct: product.isFeaturedProduct,
        stockLevel: product.stockLevel,
        category: product.category,
      });
    }


    res.status(200).json({ message: 'Data inserted successfully!' });
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Failed to fetch or insert data' });
  }
}

```

Proposed Solution

1 - Sanity API Configuration:

- Connect to Sanity's API using a preset dataset and project ID, secured with an API token.
- Store sensitive information like the project ID and dataset securely in environment variables.

2 - Retrieving Data from Sanity:

- Use GROQ queries to pull organized content from Sanity CMS.
- This includes important product details like categories, descriptions, and pricing.

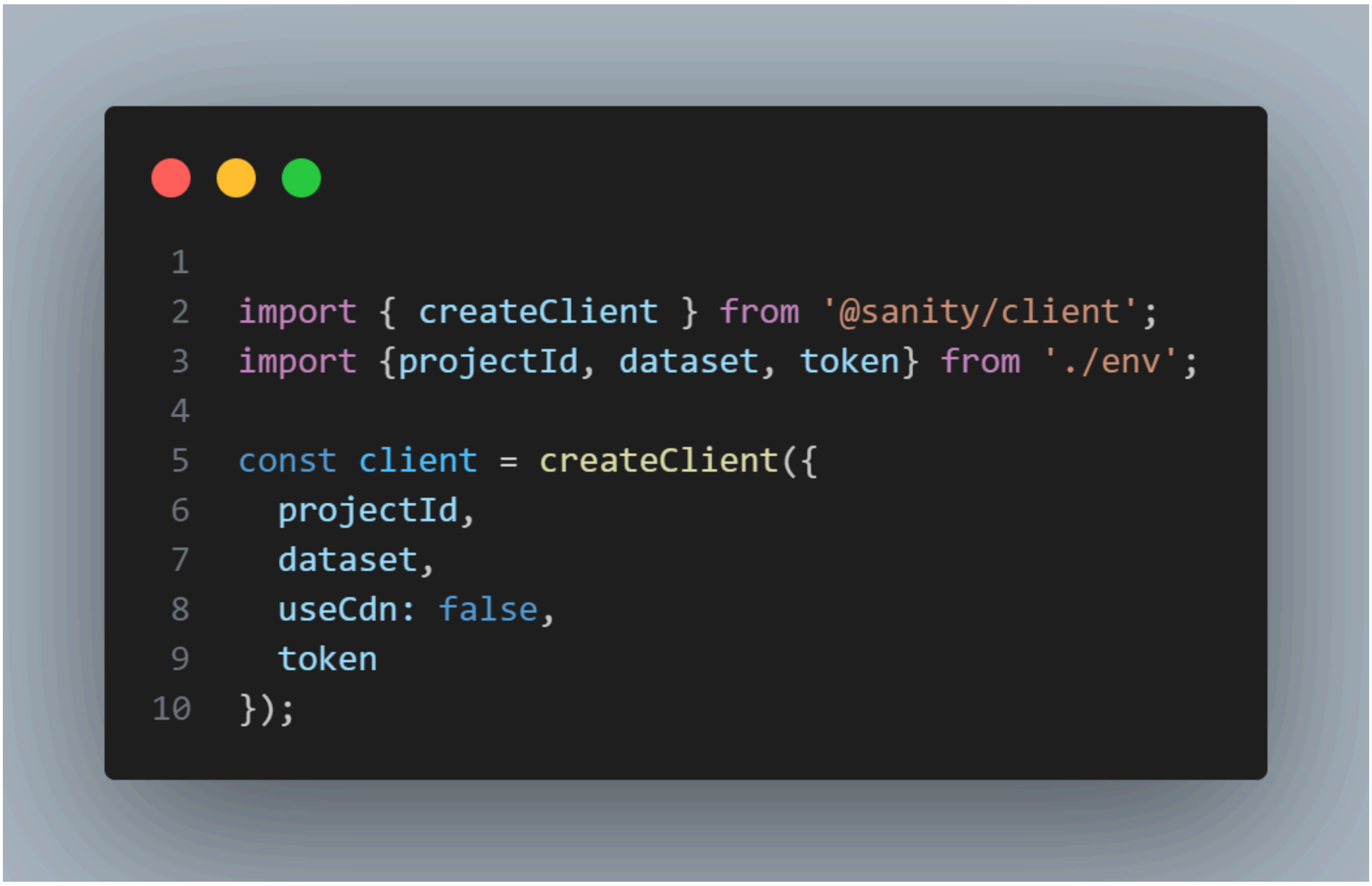
3 - Mapping and Adjusting Data:

- **Restructure the fetched data to make it compatible with the app's schema.**

4 - Inserting Data into the Database:

- **Insert the data into the database through a REST API or direct commands.**
- **Error handling ensures everything moves smoothly without issues.**

CLIENT PAGE CODE

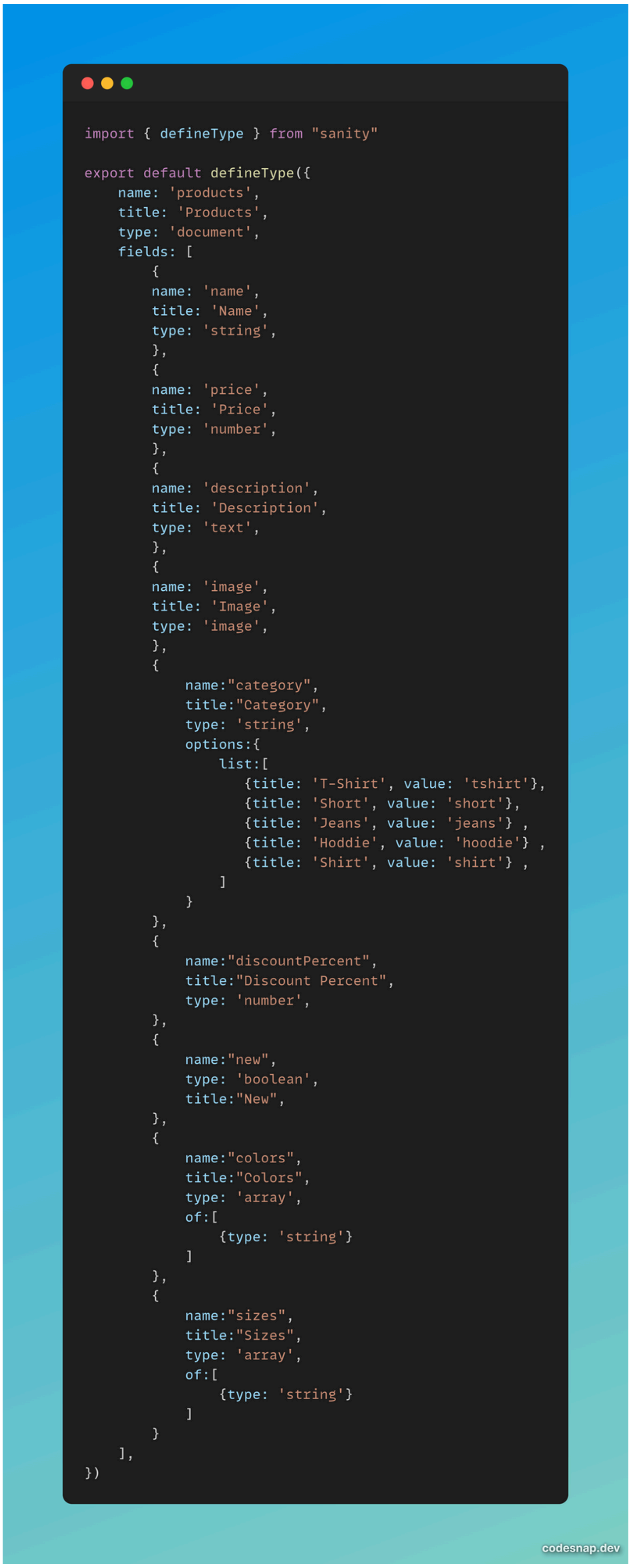


```

import { createClient } from '@sanity/client';
import {projectId, dataset, token} from './env';

const client = createClient({
  projectId,
  dataset,
  useCdn: false,
  token
});
```

Schema Code

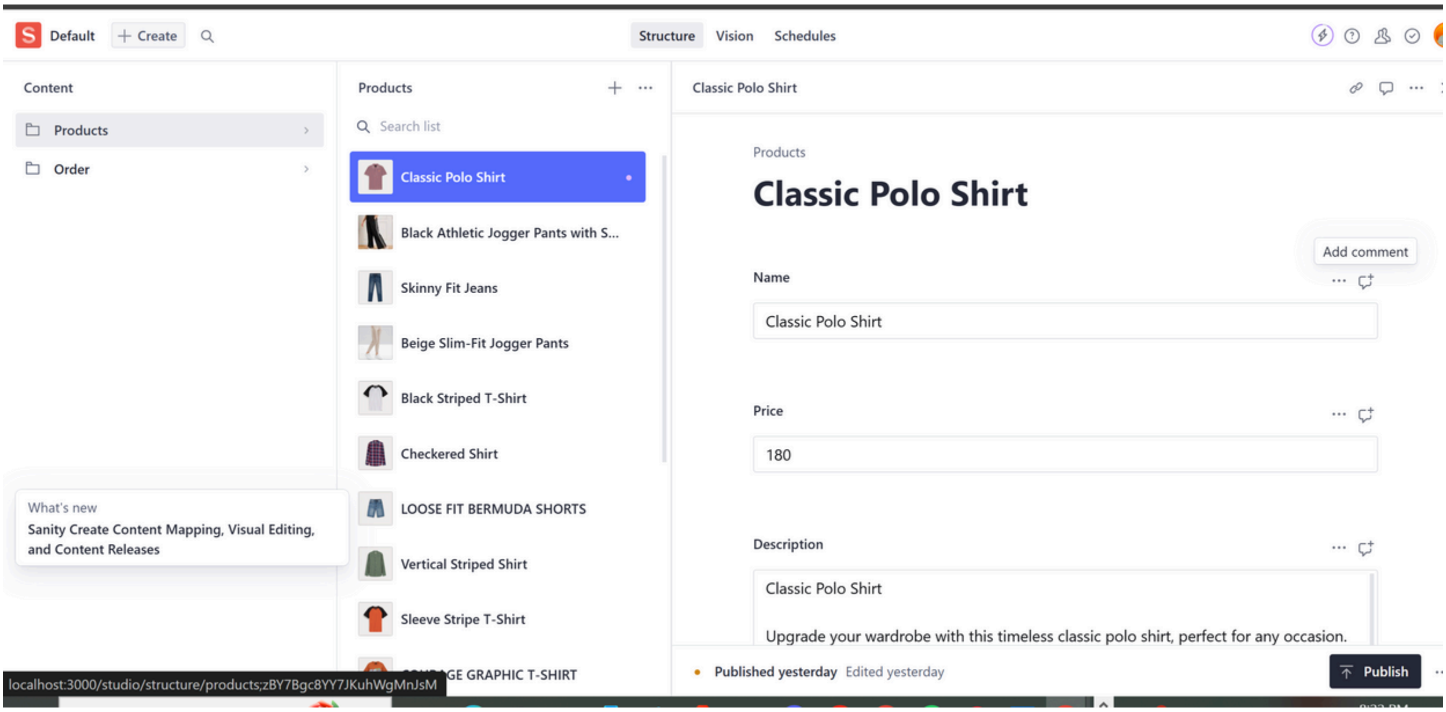
The schema defines how content is structured in Sanity CMS. Main parts of the schema include:

- **Title**: The product name.
- **Slug**: A unique identifier for product URLs.
- **Description**: A detailed description of the product.
- **Price**: The product’s price.
- **Image**: A field to store product images.



```
import { defineType } from "sanity"

export default defineType({
    name: 'products',
    title: 'Products',
    type: 'document',
    fields: [
        {
        name: 'name',
        title: 'Name',
        type: 'string',
        },
        {
        name: 'price',
        title: 'Price',
        type: 'number',
        },
        {
        name: 'description',
        title: 'Description',
        type: 'text',
        },
        {
        name: 'image',
        title: 'Image',
        type: 'image',
        },
        {
            name:"category",
            title:"Category",
            type: 'string',
            options:{
                list:[
                   {title: 'T-Shirt', value: 'tshirt'},
                   {title: 'Short', value: 'short'},
                   {title: 'Jeans', value: 'jeans'} ,
                   {title: 'Hoddie', value: 'hoodie'} ,
                   {title: 'Shirt', value: 'shirt'} ,
                ]
            }
        },
        {
            name:"discountPercent",
            title:"Discount Percent",
            type: 'number',
        },
        {
            name:"new",
            type: 'boolean',
            title:"New",
        },
        {
            name:"colors",
            title:"Colors",
            type: 'array',
            of:[
                {type: 'string'}
            ]
        },
        {
            name:"sizes",
            title:"Sizes",
            type: 'array',
            of:[
                {type: 'string'}
            ]
        }
    ],
})
```

Sanity Output



Content

- Products
- Order

What's new
Sanity Create Content Mapping, Visual Editing, and Content Releases

Products

Search list

- Classic Polo Shirt
- Black Athletic Jogger Pants with S...
- Skinny Fit Jeans
- Beige Slim-Fit Jogger Pants
- Black Striped T-Shirt
- Checkered Shirt
- LOOSE FIT BERMUDA SHORTS
- Vertical Striped Shirt
- Sleeve Stripe T-Shirt
- ...GE GRAPHIC T-SHIRT

localhost:3000/studio/structure/products;zBY7Bgc8YY7JKuhWgMnJsM

Classic Polo Shirt

Products

Classic Polo Shirt

Name
Classic Polo Shirt

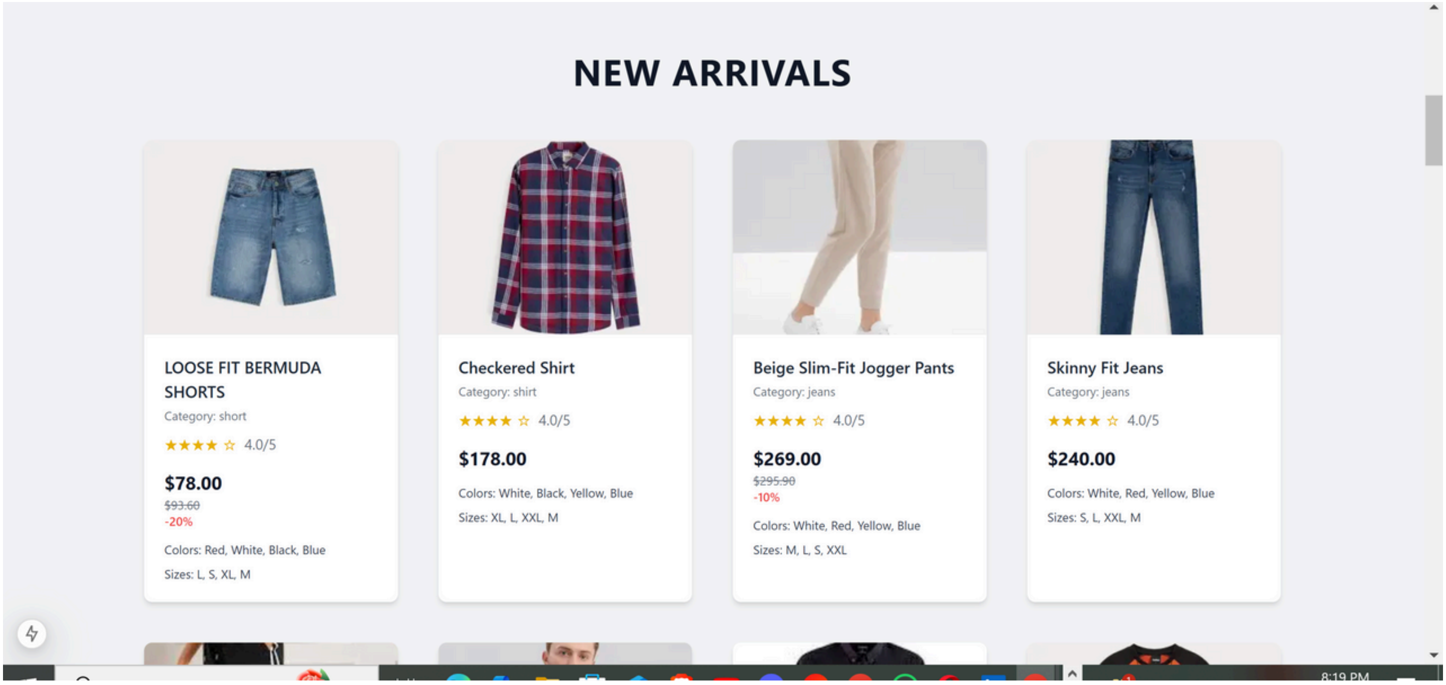
Price
180

Description
Classic Polo Shirt

Upgrade your wardrobe with this timeless classic polo shirt, perfect for any occasion.

Published yesterday Edited yesterday

Products on UI



NEW ARRIVALS

LOOSE FIT BERMUDA SHORTS
Category: short
★★★★☆ 4.0/5
$78.00
~~$93.60~~
-20%
Colors: Red, White, Black, Blue
Sizes: L, S, XL, M

Checkered Shirt
Category: shirt
★★★★☆ 4.0/5
$178.00
Colors: White, Black, Yellow, Blue
Sizes: XL, L, XXL, M

Beige Slim-Fit Jogger Pants
Category: jeans
★★★★☆ 4.0/5
$269.00
~~$295.90~~
-10%
Colors: White, Red, Yellow, Blue
Sizes: M, L, S, XXL

Skinny Fit Jeans
Category: jeans
★★★★☆ 4.0/5
$240.00
Colors: White, Red, Yellow, Blue
Sizes: S, L, XXL, M

Day 3 - Success!

The API integration and data migration are now complete. This includes setting up the schema, fetching data from Sanity, and implementing dynamic routing for the marketplace.