HACKATHON 3

Day 3 - API Integration Report - [SHOP.CO]

Prepared By: MEHMIL ZEESHAN

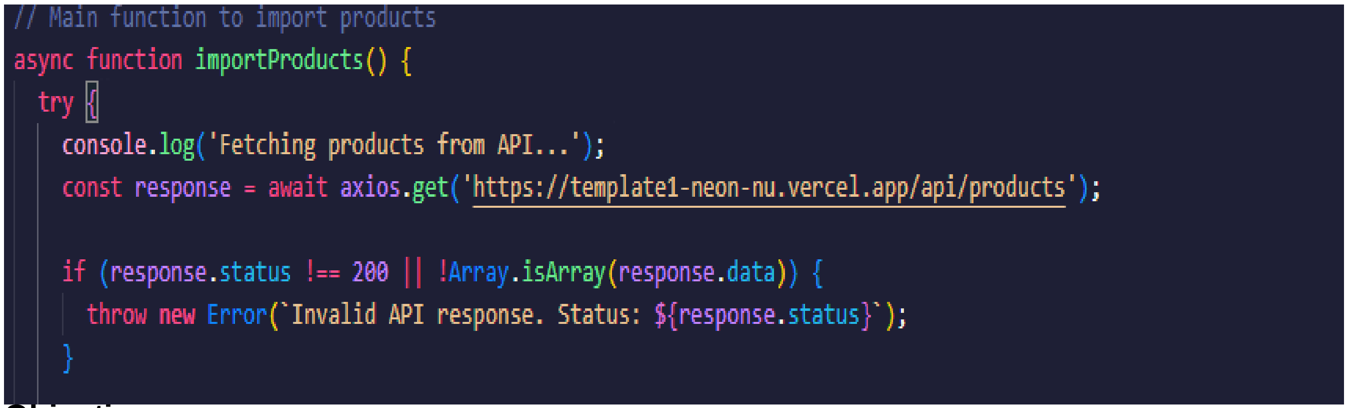
1. API Integration Process

Objective:

Integrating external APIs into the marketplace to fetch product data, process it, and display it on the frontend.

Steps Taken:

- **API Selection:**
- I chose the `https://template1-neon-nu.vercel.app/api/products` API to retrieve product data because it provides detailed product information for my marketplace.
- **API Integration in Next.js:** I used **`axios`** to fetch data from the API. The API call is made inside **`getServerSideProps`** to ensure server-side rendering and real-time data fetching.
- This API call ensures that product data is fetched and passed to the page for rendering.



```
// Main function to import products
async function importProducts() {
  try {
    console.log('Fetching products from API...');
    const response = await axios.get('https://template1-neon-nu.vercel.app/api/products');

    if (response.status !== 200 || !Array.isArray(response.data)) {
      throw new Error(`Invalid API response. Status: ${response.status}`);
    }
```

Objective:

To align the Sanity CMS schemas with the API data for smooth migration and data population.

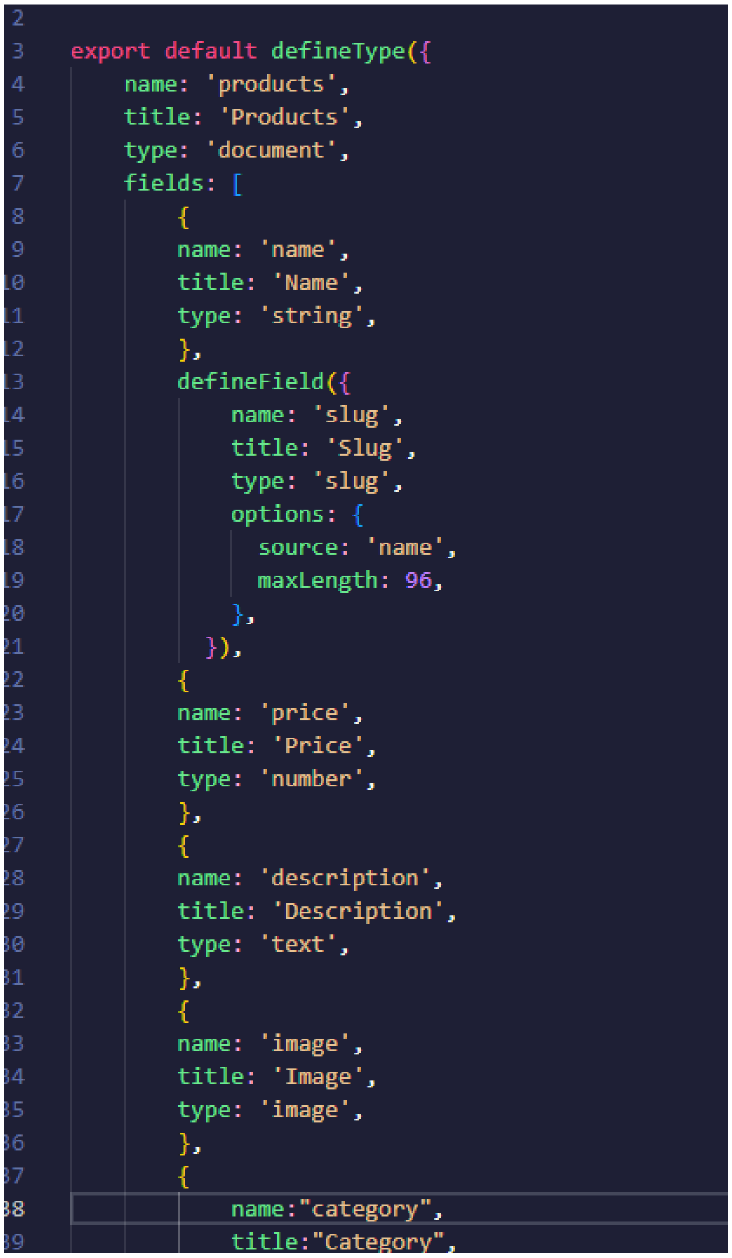
- **Identified Schema Mismatches:**

There were a few mismatches in field names (e.g., "product_name" vs. "name") and data types (e.g., "price" as a string in the API vs. number in Sanity).

I adjusted my schema in Sanity CMS to accommodate these differences.

Updated Sanity Schema:

This schema update ensures proper mapping of the API data to the CMS.



```
export default defineType({
    name: 'products',
    title: 'Products',
    type: 'document',
    fields: [
        {
        name: 'name',
        title: 'Name',
        type: 'string',
        },
        defineField({
            name: 'slug',
            title: 'Slug',
            type: 'slug',
            options: {
              source: 'name',
              maxLength: 96,
            },
          }),
        {
        name: 'price',
        title: 'Price',
        type: 'number',
        },
        {
        name: 'description',
        title: 'Description',
        type: 'text',
        },
        {
        name: 'image',
        title: 'Image',
        type: 'image',
        },
        {
            name:"category",
            title:"Category",
```

3. Migration Steps and Tools Used

Migration Process:

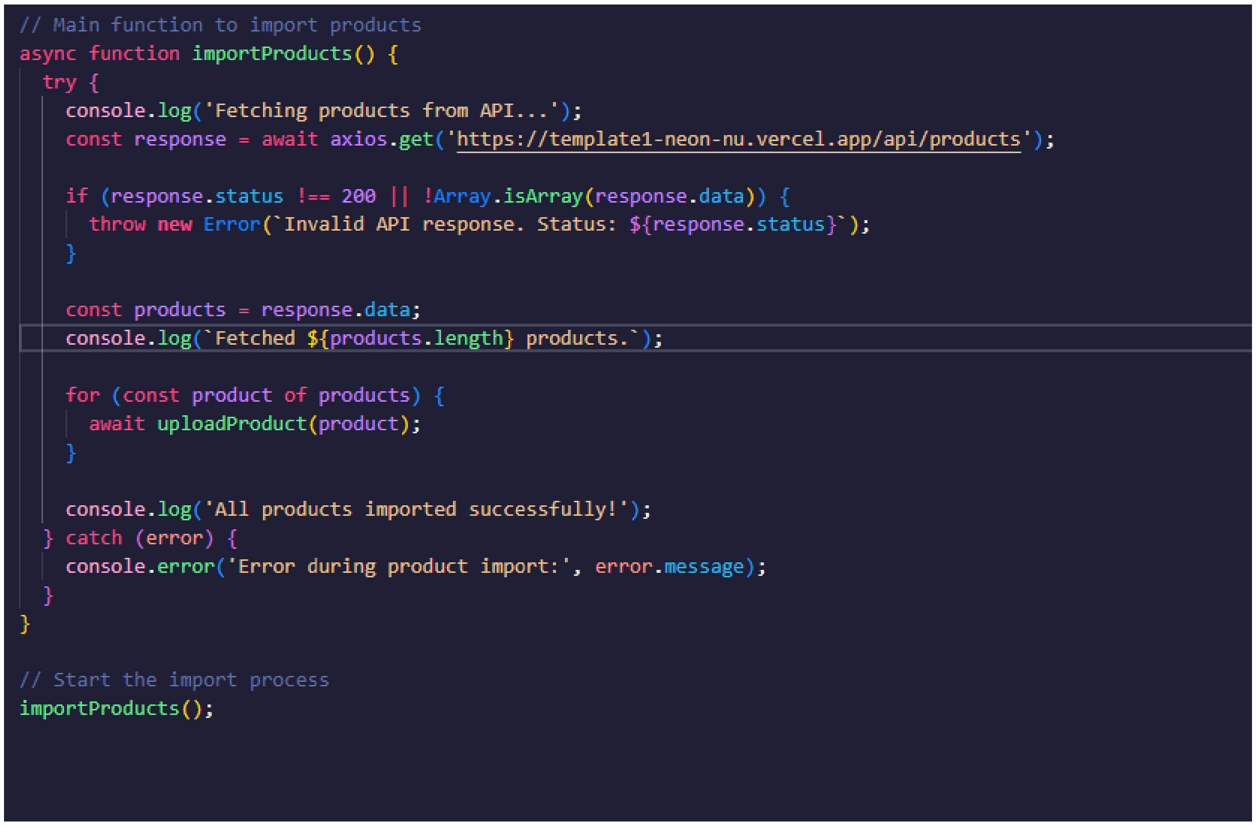
- **Tool Used:**

I used **Sanity CLI** to perform the migration. This tool helps to import data into the Sanity CMS from an external source.

- **Data Transformation:**

The data fetched from the API was transformed to match the Sanity schema format before being imported.

Migration Script:



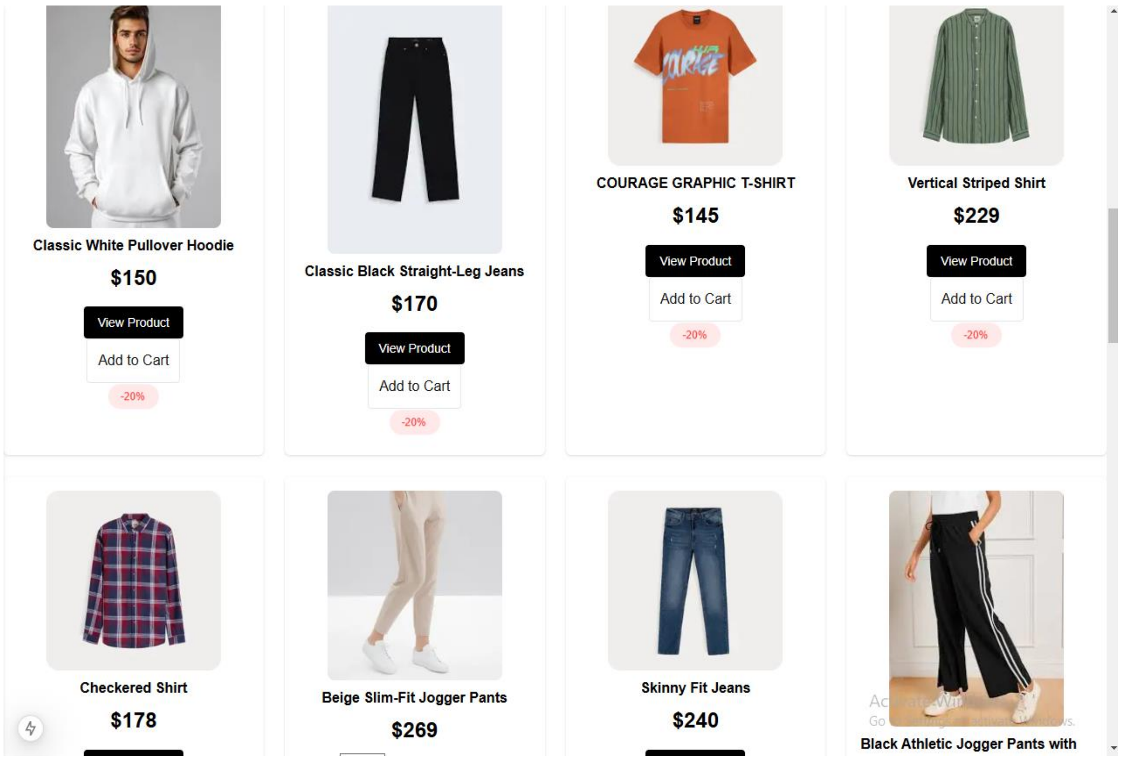
```
// Main function to import products
async function importProducts() {
  try {
    console.log('Fetching products from API...');
    const response = await axios.get('https://template1-neon-nu.vercel.app/api/products');

    if (response.status !== 200 || !Array.isArray(response.data)) {
      throw new Error(`Invalid API response. Status: ${response.status}`);
    }

    const products = response.data;
    console.log(`Fetched ${products.length} products.`);

    for (const product of products) {
      await uploadProduct(product);
    }

    console.log('All products imported successfully!');
  } catch (error) {
    console.error('Error during product import:', error.message);
  }
}

// Start the import process
importProducts();
```

Migration Process Validation:

I validated the data by checking for any missing fields or mismatches in the API response and Sanity CMS, and logged discrepancies for further investigation.

4. Screenshots

1. **Data Displayed in Frontend:**



Classic White Pullover Hoodie
$150
View Product
Add to Cart
-20%

Classic Black Straight-Leg Jeans
$170
View Product
Add to Cart
-20%

COURAGE GRAPHIC T-SHIRT
$145
View Product
Add to Cart
-20%

Vertical Striped Shirt
$229
View Product
Add to Cart
-20%

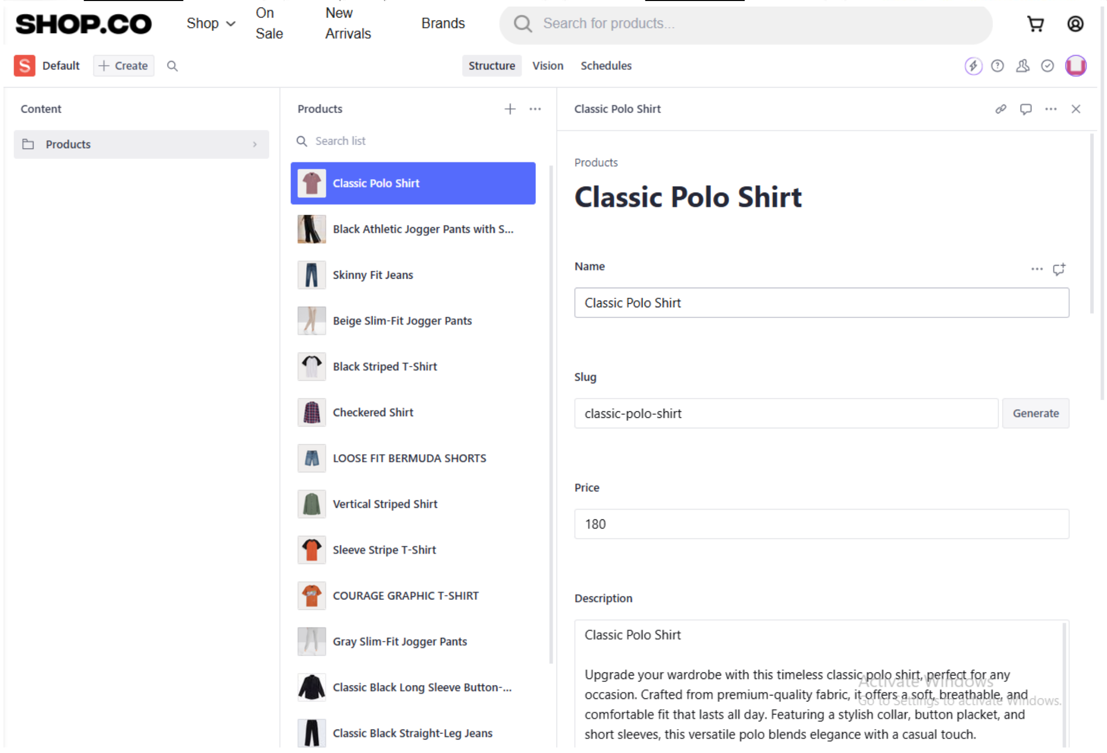
Checkered Shirt
$178

Beige Slim-Fit Jogger Pants
$269

Skinny Fit Jeans
$240

Black Athletic Jogger Pants with

a.



SHOP.CO Shop On Sale New Arrivals Brands Search for products...

Default + Create Structure Vision Schedules

Content
Products

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
COURAGE GRAPHIC T-SHIRT
Gray Slim-Fit Jogger Pants
Classic Black Long Sleeve Button-...
Classic Black Straight-Leg Jeans

Classic Polo Shirt
Products
Classic Polo Shirt

Name
Classic Polo Shirt

Slug
classic-polo-shirt Generate

Price
180

Description
Classic Polo Shirt

Upgrade your wardrobe with this timeless classic polo shirt, perfect for any occasion. Crafted from premium-quality fabric, it offers a soft, breathable, and comfortable fit that lasts all day. Featuring a stylish collar, button placket, and short sleeves, this versatile polo blends elegance with a casual touch.

b.

5. Best Practices Followed

1. **Sensitive Data Handling:**

   a. I stored the API keys in `.env` files to ensure they were not exposed in version control.

2. **Clean Code Practices:**
   a. Descriptive variable names and modular functions were used.
   b. I added comments for complex logic.
3. **Data Validation:**
   a. I used schema validation to check that data types and constraints matched during migration.
4. **Version Control:**
   a. Frequent commits were made, with meaningful messages to document each step of the process.

6. Conclusion

The API integration was successful, and the data was properly migrated to Sanity CMS, with all fields matching the required schema. The frontend now displays the data dynamically from the API, and the CMS contains all necessary product details.