Day 3 - API Integration Report – Shop.co

Introduction:

This document provides a concise overview of the process we follow to fetch data from an API, migrate the retrieved data into Sanity, and subsequently display the product information on the frontend. It outlines the key steps and methods used to ensure seamless integration and efficient data management across the system.

API Integration Process:

- **API Endpoints:** The provided API includes the /product endpoint.
- **Fetching Data:** We utilized the Axios library to perform HTTP requests to the API.
- A GET method is called, and the API URL is provided within the code to fetch data from Sanity. The response is then stored in a variable named response for further use. Below is an example of the integration:

```
// Fetch data from external API
const response = await axios.get('https://template1-neon-nu.vercel.app/api/products');
const products = response.data;
```

Adjustments Made to Schemas:

To changes in the product data structure, we incorporated additional fields into the API schema:

- **Category Field:** This field categorizes products, facilitating organized data retrieval and display.
- **Tag Field:** Tags were added to allow for more flexible filtering and searching of products based on specific attributes.
- **Size Field:** Inclusion of size information enables users to select products according to their dimensional preferences.
- **Color Field:** Adding color details provides users with options to choose products based on color variants.

These schema enhancements aim to improve the user experience by providing more detailed and organized product information.

```
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: 'slug',
        type: 'slug',
        title: 'Product slug',
        options: {
            source: 'name',
          }

      },
        {
        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: 'Hoodie', value: 'hoodie'} ,
                    {title: 'Shirt', value: 'shirt'} ,
                ]
            }
        },
        {
            name:"tags",
            title:"tags",
            type: 'string',
            options:{
                list:[
                    {title: 'NewArrivals', value: 'newarrivals'},
                    {title: 'Topselling', value: 'topselling'},
                ]
            }
        },
        {
            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'}
            ]
        }
    ],
})
```

Migration Steps:

Installing

- First step install all the libraries need to import data from api such as axios

Sanity Setup

- Then install the sanity to make product schema in which we will put our data after fetching it from api and display in sanity studio

4. Set Up Sanity Client

- Create a `sanityClient.js` file inside a `sanity-migration` folder to configure the Sanity client.
- This client will be used to interact with your Sanity backend for data migration.

```
import { createClient } from '@sanity/client';

export const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID, // Replace with your project ID
  dataset: 'production',
  apiVersion: '2024-01-04',
  useCdn: false,
  token: process.env.SANITY_TOKEN,
});
```

. Create `importData.js` for Data Migration

- Write an `importData.js` script to fetch data from the API using Axios.
- Transform the fetched data into a format compatible with your Sanity schema and integrate it into Sanity.

```
import axios from 'axios';
import { client } from './sanityClient.js';

async function uploadImageToSanity(imageUrl: string): Promise<string> {

  try {
    // Fetch the image from the URL and convert it to a buffer
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);

    // Upload the image to Sanity
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop(), // Extract the filename from URL
    });

    // Debugging: Log the asset returned by Sanity
    console.log('Image uploaded successfully:', asset);

    return asset._id; // Return the uploaded image asset reference ID
  } catch (error) {
    console.error('❌ Failed to upload image:', imageUrl, error);
    throw error;
  }
}

async function importData() {
    try {
      // Fetch data from external API
      const response = await axios.get('https://template1-recon-nu.vercel.app/api/products');
      const products = response.data;

      // Iterate over the products
      for (const product of products) {
        let imageRef = '';

        // Upload image and get asset reference if it exists
        if (product.image) {
          imageRef = await uploadImageToSanity(product.image);
        }

        const sanityProduct = {
          _id: `product-${product.id}`, // Prefix the ID to ensure validity
          _type: 'products',
      name: product.name,
      description: product.description,
      price: product.price,
      image: {
        _type: 'image',
        asset: {
          _ref: imageRef,
        },
      },
      category: product.category,
      discountPercent: product.discountPercent,
      isNew: product.isNew,
      colors: product.colors,
      sizes: product.sizes

        };

        // Log the product before attempting to upload it to Sanity
        console.log('Uploading product:', sanityProduct);

        // Import data into Sanity
        await client.createOrReplace(sanityProduct);
        console.log(`✅ Imported product: ${sanityProduct.name}`);
      }

      console.log('✅ Data import completed!');
    } catch (error) {
      console.error('❌ Error importing data:', error);
    }
}

importData();
```

Tools Used

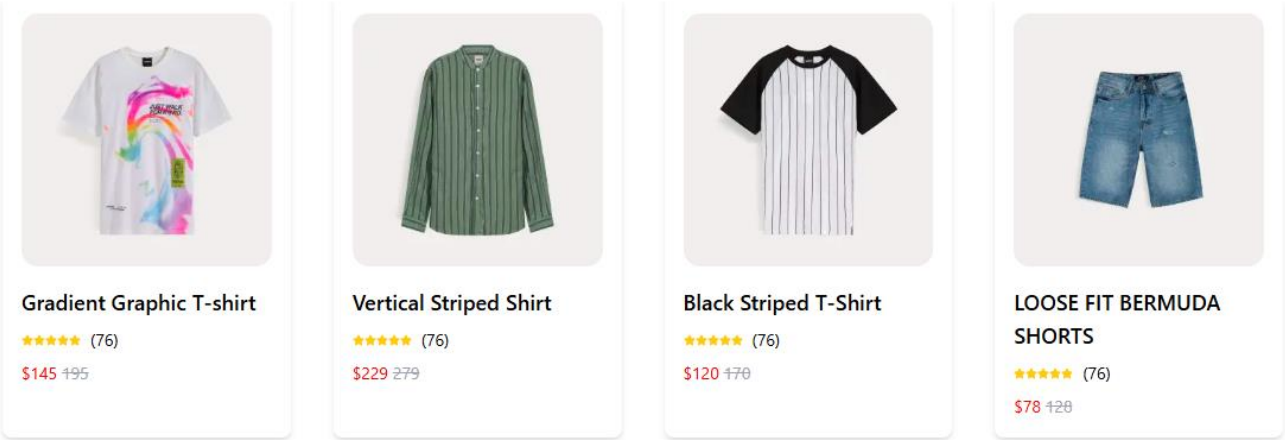
1. **Axios**
    - Used for fetching data from APIs, including Sanity if needed for read operations.
    - Allows making HTTP requests (GET, POST, etc.) seamlessly.
2. **Dotenv**
    - Used to manage and store sensitive configuration details like API keys, tokens, and URLs in `.env` file.
    - Keeps environmental variables secure and out of the main codebase.
3. **Sanity Client**
    - A JavaScript client library for interacting with the Sanity CMS.
    - Used for writing data to Sanity, fetching data, and other CRUD operations.

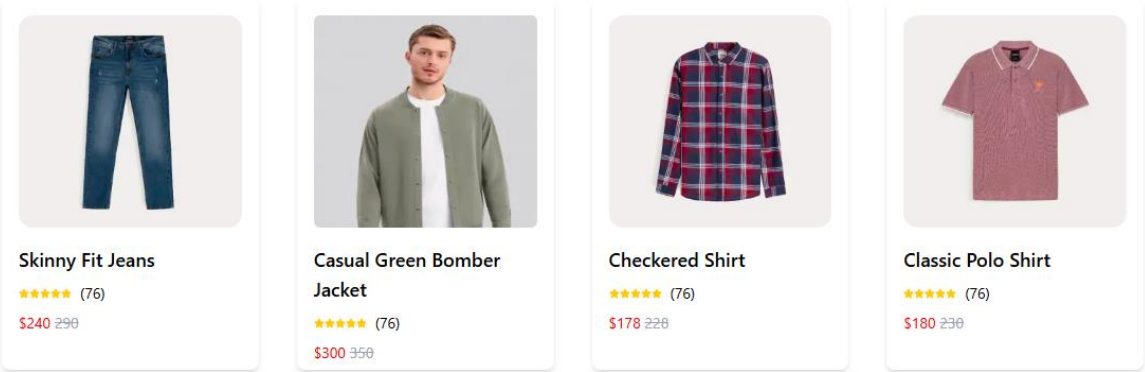
Screenshot of

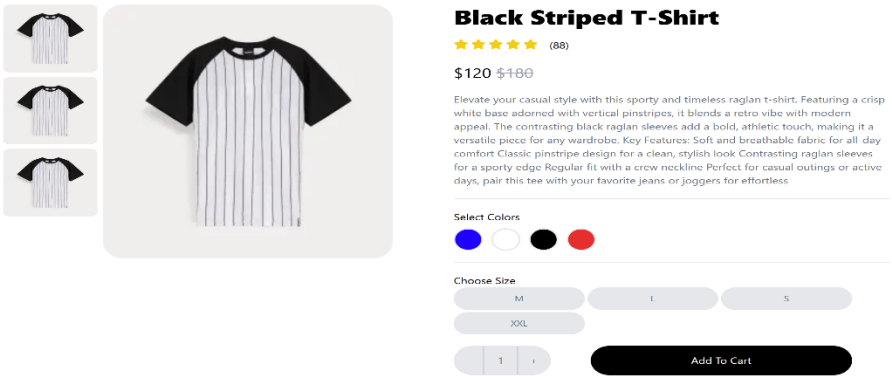
1. Api calls output:

```
}
✅ Imported product: Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops
Uploading product: {
  _id: 'product-2',
  _type: 'product',
  name: 'Mens Casual Premium Slim Fit T-Shirts ',
  price: 22.3,
  discountPercentage: 0,
  tags: [ "men's clothing" ],
  image: {
    _type: 'image',
    asset: {
      _type: 'reference',
      _ref: 'image-e1d220cc41c863e93dff756780aefc931799ac1a-663x879-jpg'
    }
  },
  description: 'Slim-fitting style, contrast raglan long sleeve, three-button henley placket, light weight & soft fabric for
 breathable and comfortable wearing. And Solid stitched shirts with round neck made for durability and a great fit for casua
l fashion wear and diehard baseball fans. The Henley style round neckline includes a three-button placket.',
  colors: [],
  sizes: []
}
✅ Imported product: Mens Casual Premium Slim Fit T-Shirts
Uploading product: {
  _id: 'product-3',
  _type: 'product',
  name: 'Mens Cotton Jacket',
  price: 55.99,
  discountPercentage: 0,
  tags: [ "men's clothing" ],
  image: {
    _type: 'image',
    asset: {
      _type: 'reference',
      _ref: 'image-5519a5fa0fee4d3f36f7c3e3b9e4c81e501c8c61-679x755-jpg'
    }
```

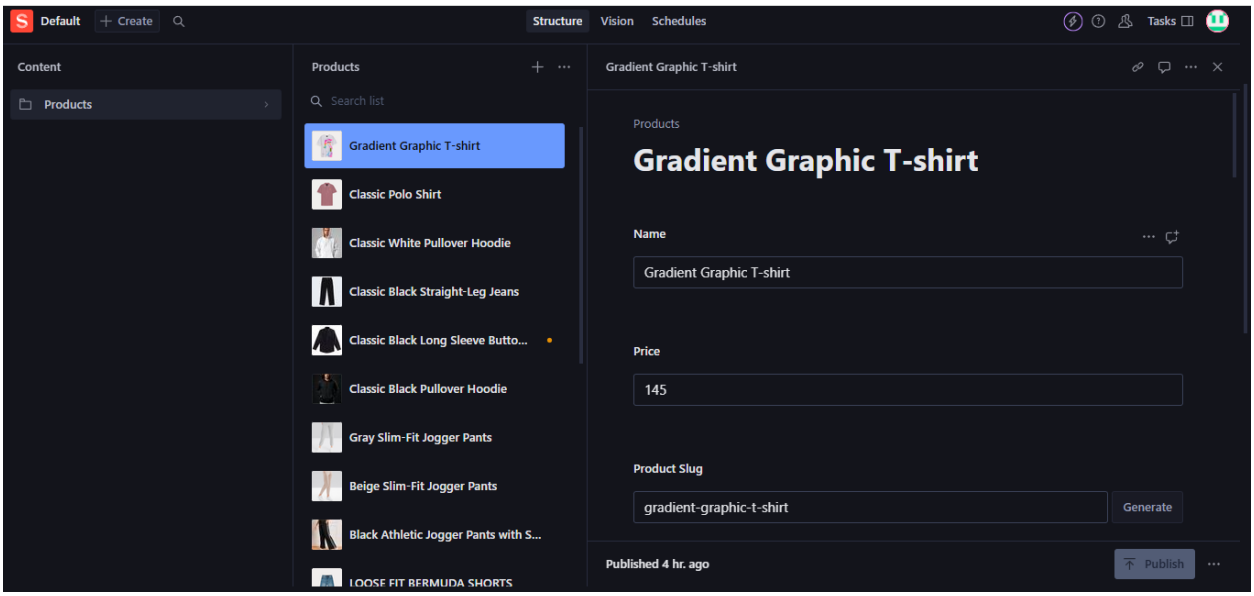
2. Data Display In Frontend







3. Sanity CMS Fields:



Conclusion:

We successfully fetched data from an external API, transformed it to match the required structure, and integrated it with Sanity CMS. The data was migrated to Sanity, making it accessible for content management. Using Sanity's querying capabilities, we displayed the data dynamically on the frontend.