Mehdi Abbas - 00461734

Day 3 - API Integration Report

1. Introduction

This report is about integrating an external API into a marketplace project. The goal was to fetch product data, including images, from the API and save it in a Sanity CMS. This report will also outline some adjustments made on the Sanity schema, migration steps, and the tools that were used.

2. API Integration Process

a) Fetching Data from the API:

Product data was retrieved using Axios, a promise-based HTTP client.

The API provided details such as product name, image URL, price, stock level, and category.

b) Handling Images:

Images were hosted externally, and their URLs were fetched for each product.

Sanity's asset pipeline was employed to upload these images into the Sanity media library. This involved converting the image URLs into binary buffers using Axios.

c) Importing Data into Sanity:

Once images were uploaded, the product data (including image references) was imported into Sanity CMS via the Sanity client.

Every product document included fields like name, description, price, category, and a reference to the uploaded image.

3. Sanity Schema Adjustments

The Sanity schema was updated to accommodate the advanced product model, which contains the following fields:

Basic Fields: ID, name, image, price, description, etc.

Additional Fields:

Star rating

Rating count

isTrending (boolean)

isLatest (boolean)

Slug references (for dynamic routing).

Schema Structure Below:

```
1    import { title } from "process";
2
3    export default {
4        name: 'product',
5        type: 'document',
6        title: 'Product',
7        fields: [
8            {
9                name: 'name',
10               type: 'string',
11               title: 'Name',
12               validation: (Rule: any) => Rule.required().error('Name is required'),
13           },
14           {
15               title: 'Slug',
16               name: 'slug',
17               type: 'slug',
18               options: {
19                   source: 'name',
20                   maxLength: 200, // will be ignored if slugify is set
21                   slugify: (input: string) => input
22                       .toLowerCase()
23                       .replace(/\s+/g, '-')
24                       .slice(0, 200)
25               }
26           },
27           {
28               name: 'image',
29               type: 'image',
30               title: 'Image',
31               options: {
32                   hotspot: true,
```

4. Migration Steps and Tools Used

Migration Steps:

Fetching Product Data:

Axios was used to fetch product data from the external API endpoint.

Uploading Images to Sanity:

Images were fetched by their URLs and uploaded to Sanity using client.assets.upload().

Creating Product Documents:

Each product document was created in Sanity CMS using client.create(), where the product is associated with its corresponding image.

Validation:

The imported data was validated to ensure it was being rendered correctly in Sanity CMS and the frontend.

Tools Used:

Sanity Client: To interact with the Sanity API and manage product documents.

Axios: For making HTTP requests to fetch API data and download images.

dotenv: For securely managing environment variables such as the Sanity API token.

Next.js: This is used for rendering the product data on the frontend.

5. API Data Preview

template-03-api.vercel.app/api/products

Pretty-print ☑

{
  "success": true,
  "data": [
    {
      "productName": "Nike Air Force 1 Mid '07",
      "category": "Men's Shoes",
      "price": 10795,
      "inventory": 20,
      "colors": [
        "White"
      ],
      "status": "Just In",
      "image": "https://template-03-api.vercel.app/products/1.png",
      "description": "The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for ever durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance."
    },
    {
      "productName": "Nike Court Vision Low Next Nature",
      "category": "Men's Shoes",
      "price": 4995,
      "inventory": 35,
      "colors": [
        "Black"
      ],
      "status": "Just In",
      "image": "https://template-03-api.vercel.app/products/2.png",
      "description": "The Nike Court Vision Low Next Nature combines retro basketball style with modern sustainability. Made with a durable materials, and ultimate comfort for everyday wear."
    },
    {
      "productName": "Nike Air Force 1 PLT.AF.ORM",
      "category": "Women's Shoes",
      "price": 8695,
      "inventory": 18,
      "colors": [
        "White"
      ],
      "status": "Just In",

6. Data in Sanity CMS

localhost:3000/studio/structure/product/NxalkVf0z91aTr5xFotyLl

☐ mhhasanul@gmail.com   English ˅   USD ˅   Login 👤   Wishlist ♡

Hekto   Home   Products   Blog   Shop   Contact   [Search   ] 🔍

S Default   + Create   🔍   Structure Vision Schedules   ⚡ ⑦ 🖧 Tasks 🔲 👤

Content   Product   + ⋯   Luxury Flower Shell Sofa Chair   🔗 💬 ⋯ ✕

📁 Product   ›   🔍 Search list

                         🪑 Luxury Flower Shell Sofa Chair   Product

                         🪑 Cantilever Chair   **Luxury Flower Shell Sofa Chair**

                         🪑 Nordic Net Red Chair
                                                                                              Add comment
                         🪑 Sobuy Blue Folding Chair Woo… •   Name   ⋯ ⟳

                         🪑 Varmora Plastic Chair Solid   [Luxury Flower Shell Sofa Chair   ]

                         🪑 Matilda Velvet Chair – Pink

                         🪑 Diondre Chair - Tuft Button - Ac…   Slug
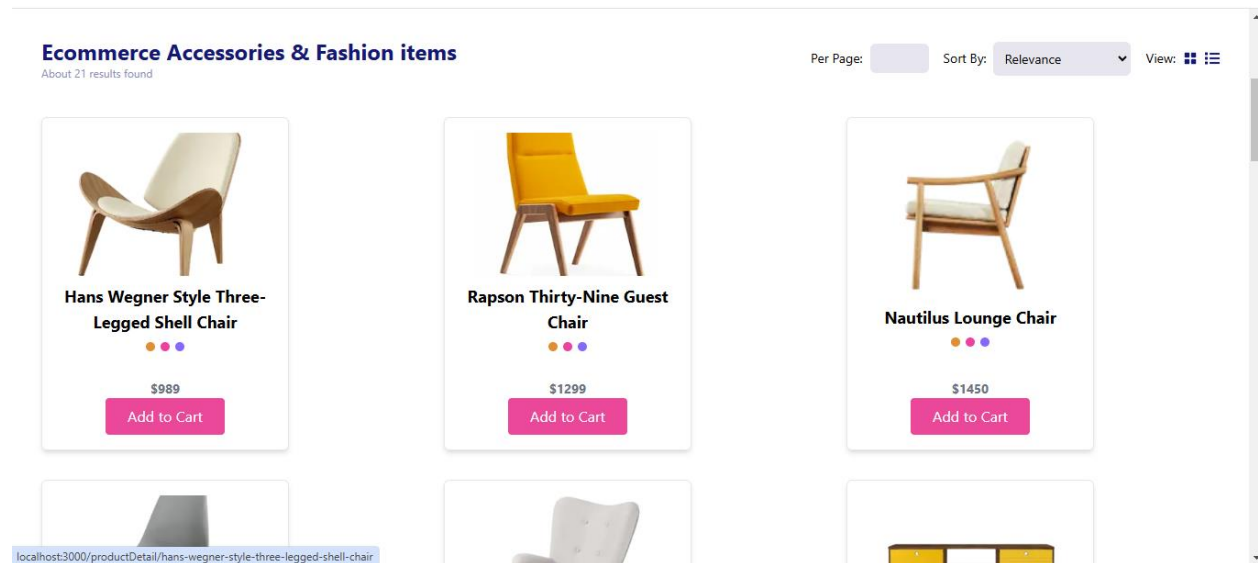
                                                                    [luxury-flower-shell-sofa-chair   ] Generate

7. Products on Frontend

7. Conclusion

This report documents the API integration process, schema adjustments, migration steps, and tools used for importing product data into Sanity CMS. It also provides code snippets for fetching API data and uploading product information. The project successfully integrates product data from an external API, handles image uploads, and renders the data in a Next.js frontend. 8. Conclusion

This report details the end-to-end process of how to integrate product data from an external API into Sanity CMS. Key accomplishments include:

Fetching and processing product data.

Uploading externally hosted images into the Sanity media library.

Creating and validating product documents in Sanity CMS.

Successfully rendering the imported data on a Next.js frontend.

The project demonstrates efficient API integration, schema customization, and seamless data migration, resulting in a fully functional marketplace application.