Day 3 - API Integration and Data Migration

Nike store

Date: 17-1-25

Name: Mehwish Sajjad

Project: API Integration and Data Migration

Overview:

The primary focus of Day 3 was on integrating APIs, specifically with **Sanity CMS**, using **GROQ queries**, and updating the schema to support dynamic data requirements. Here's a summary of the tasks accomplished:

Key Objectives:

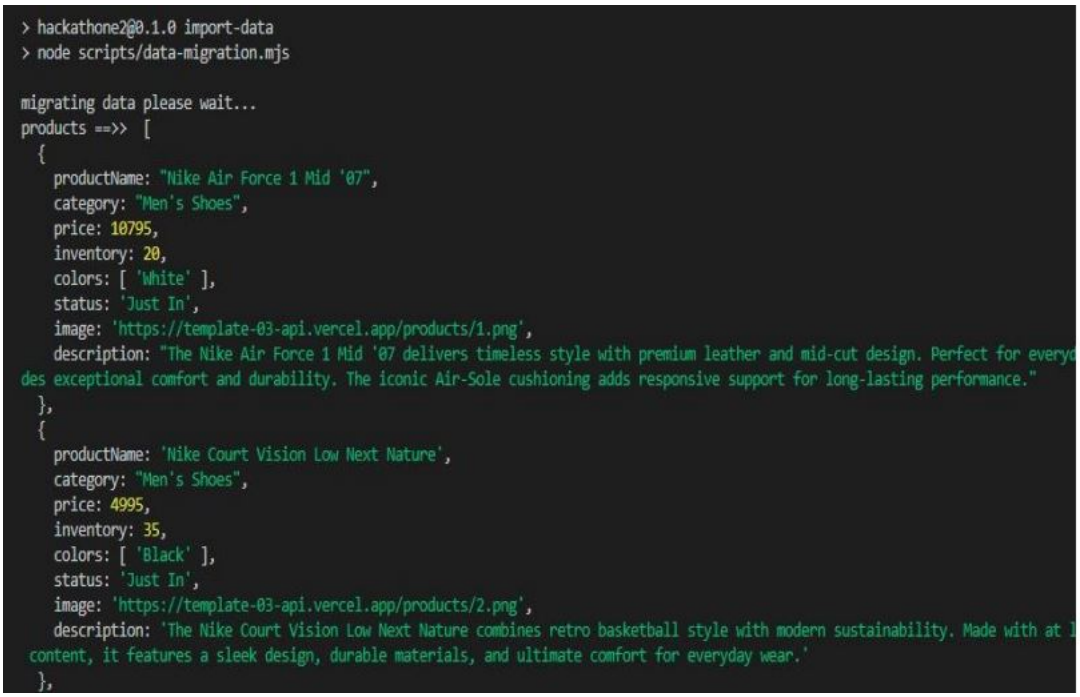
1. Establish a connection with Sanity CMS for managing product data efficiently.
2. Implement **GROQ queries** to fetch structured data dynamically from Sanity.
3. Update the **Sanity schema** to align with the evolving requirements of the api.
4. Migrate product data from an external API to Sanity CMS, including image uploads.
5. Test and validate the integration to ensure smooth data flow between the backend and frontend.

---

1. API Data Migration to Sanity:

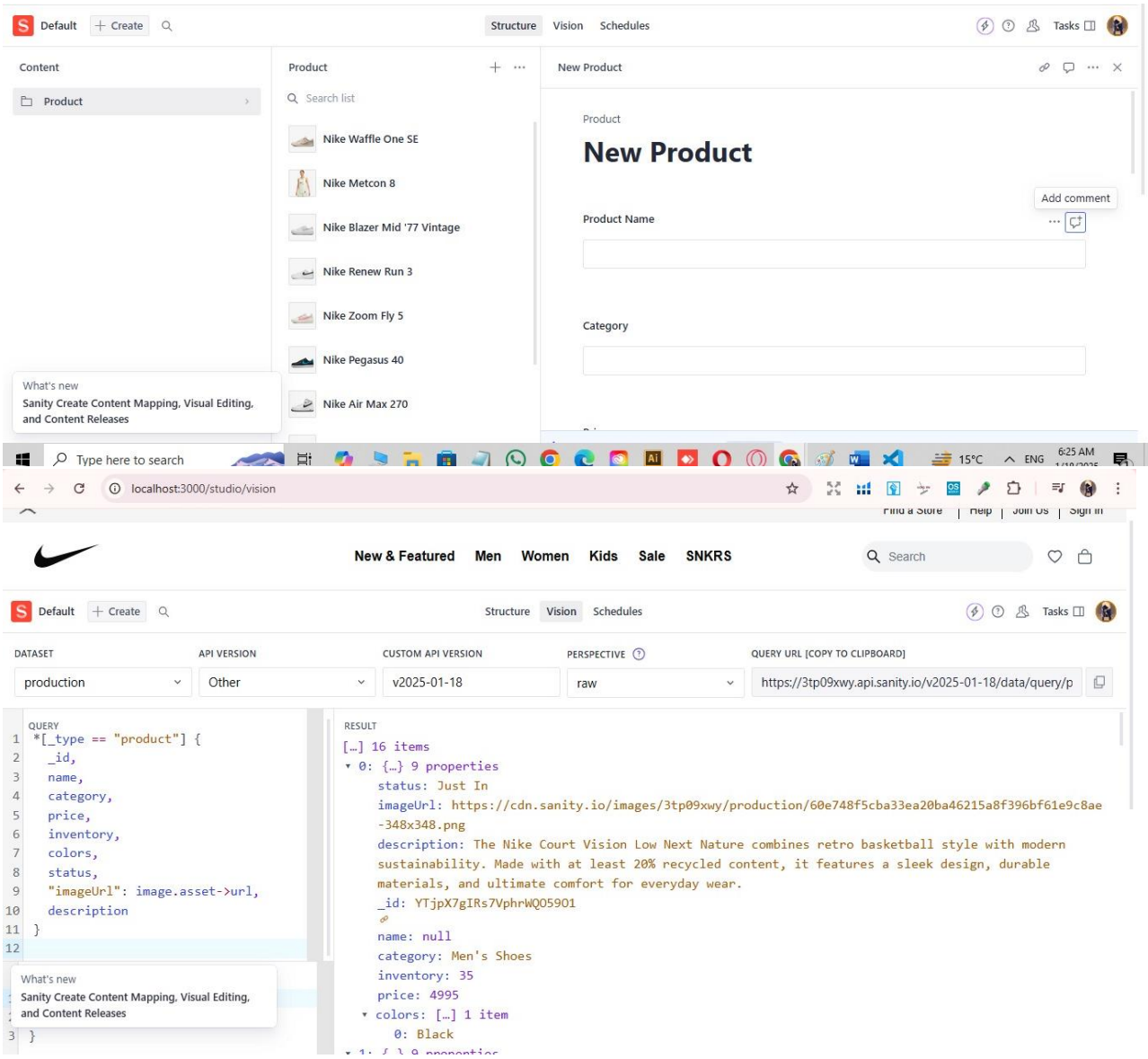
- **Objective: ○ Migrate product data, including images, from an external API to Sanity CMS.**
- **Steps Taken:**

1. **Fetch Product Data:**
   - **Used Axios to retrieve product data from an external API.**
   - **Example API Endpoint: https://template-03api.vercel.app/api/products.**
2. **Upload Images to Sanity:**
   - **Uploaded product images using the client.assets.upload method.**
   - **Ensured that each image was linked to its respective product.**
3. **Create Product Documents in Sanity:**
   - **Iterated over the fetched data and created a corresponding product document in Sanity using the client.create() method.**



2. Sanity CMS Integration:

- **Connection Setup:**
    - Integrated the @sanity/client library into the project to enable communication with the Sanity CMS.
    - Configured the client using project-specific credentials, including the projectId, dataset, and apiVersion.
- **Purpose:** To centralize product management and leverage Sanity's features, such as realtime updates, robust content modeling, and optimized image handling



importSanityData.mjs > client

```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

// Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

// Create Sanity client
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2021-08-31'
});


async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop()
    });
    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}
```

src > app > fetch > page.tsx > ...

```
import React from 'react'
import{ client } from "@/sanity/lib/client"


async function getdata() {
  const fetchdata = client.fetch(`*[_type == "product"] {
    _id,
    name,
    category,
    price,
    inventory,
    colors,
    status,
    "imageUrl": image.asset->url,
    description
  }`);
  return fetchdata;
}

export default async function Products() {
  const data = await getdata();
  console.log(data);
  return (
    <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-8 ml-8">
      {data.map((val: any, index: number) => (
        <div
          key={val._id || index} // Using _id as the unique key, fallback to index
          className="w-full max-w-sm bg-white border border-gray-200 rounded-lg shadow dark:bg-gray-800 dark:border-gra
        >
          {/* Image */}
          <a href="#">
            <img
              className="p-8 rounded-t-lg"
              src={val.imageUrl}
              alt={val.name || "Product image"}
            />
          </a>

          {/* Product Details */}
          <div className="px-5 pb-5">
            <a href="#">
              <h5 className="text-xl font-semibold tracking-tight text-gray-900 dark:text-white">
                {val.name}
              </h5>
            </a>
            <div className="flex items-center mt-2.5 mb-5">
              <div className="flex items-center space-x-1 rtl:space-x-reverse">
                {/* Stars */}
                {[...Array(4)].map((_, starIndex) => (
                  <svg
                    key={starIndex}
                    className="w-4 h-4 text-yellow-300"
                    aria-hidden="true"
                    xmlns="http://www.w3.org/2000/svg"
                    fill="currentColor"
                    viewBox="0 0 22 20"
                  >
                    <path d="M20.924 7.625a1.523 1.523 0 0 0-1.238-1.044l-5.051-.734-2.259-4.577a1.534 1.534 0 0 0-2.75
                  </svg>
                ))}
```

Our Products

Nike Air Force 1 Mid '07

The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance.

Category: Men's Shoes
Colors: White
Status: Just In

$10795

Add to Cart

Nike Court Vision Low

The Nike Court Vision Low blends basketball-inspired style with everyday comfort. Featuring a sleek design and durable materials, this versatile shoe is perfect for any casual wardrobe.

Category: Men's Shoes
Colors: White
Status: Just In

$5695

Add to Cart

Nike Metcon 8

The Nike Metcon 8 is built for intense training sessions. Featuring a stable base, durable materials, and excellent traction, it's the perfect companion for weightlifting, CrossFit, or HIIT workouts.

Category: Men's Training Shoes
Colors: Black
Status: Best Seller

$10295

Add to Cart

Nike Air Force 1 PLT.AF.ORM

The Nike Air Force 1 PLT.AF.ORM elevates the iconic AF1 silhouette with bold design updates. Featuring a platform sole and premium materials, this women's shoe redefines style and comfort in a fresh, modern way.

Category: Women's Shoes
Colors: White
Status: Just In

$8695

Add to Cart

Nike Air Force 1 React

The Nike Air Force 1 React takes the classic AF1 to the next level with React technology. Experience unparalleled cushioning and modern design details for all-day comfort and a bold statement of style.

Category: Men's Shoes
Colors: White
Status: Just In

$13295

Add to Cart

Air Jordan 1 Elevate Low

The Air Jordan 1 Elevate Low features a clean, minimal design with premium materials. Its platform sole adds a touch of height and modern edge, while ensuring comfort and durability for all-day wear.

Category: Women's Shoes
Colors: White
Status: Promo Exclusion

$11895

Add to Cart

Nike Dri-FIT UV Hyverse

The Nike Dri-FIT UV Hyverse graphic top offers comfort and UV protection in style. Made from soft, breathable fabric, this fitness top keeps you cool and dry during workouts or casual outings.

Category: Men's Short-Sleeve Graphic Fitness Top
Colors: Teal
Status: Sustainable Materials

$2495

Add to Cart

Nike Air Max 270

The Nike Air Max 270 features the largest Air unit yet for exceptional cushioning. Its bold design, breathable mesh upper, and vibrant colors make it a standout choice for style and comfort.

Category: Men's Shoes
Colors: Black
Status: Trending

$13295

Add to Cart

Nike Pegasus 40

The Nike Pegasus 40 delivers responsive cushioning and durability for everyday runners. Its lightweight construction and secure fit make it the perfect choice for training or casual runs.

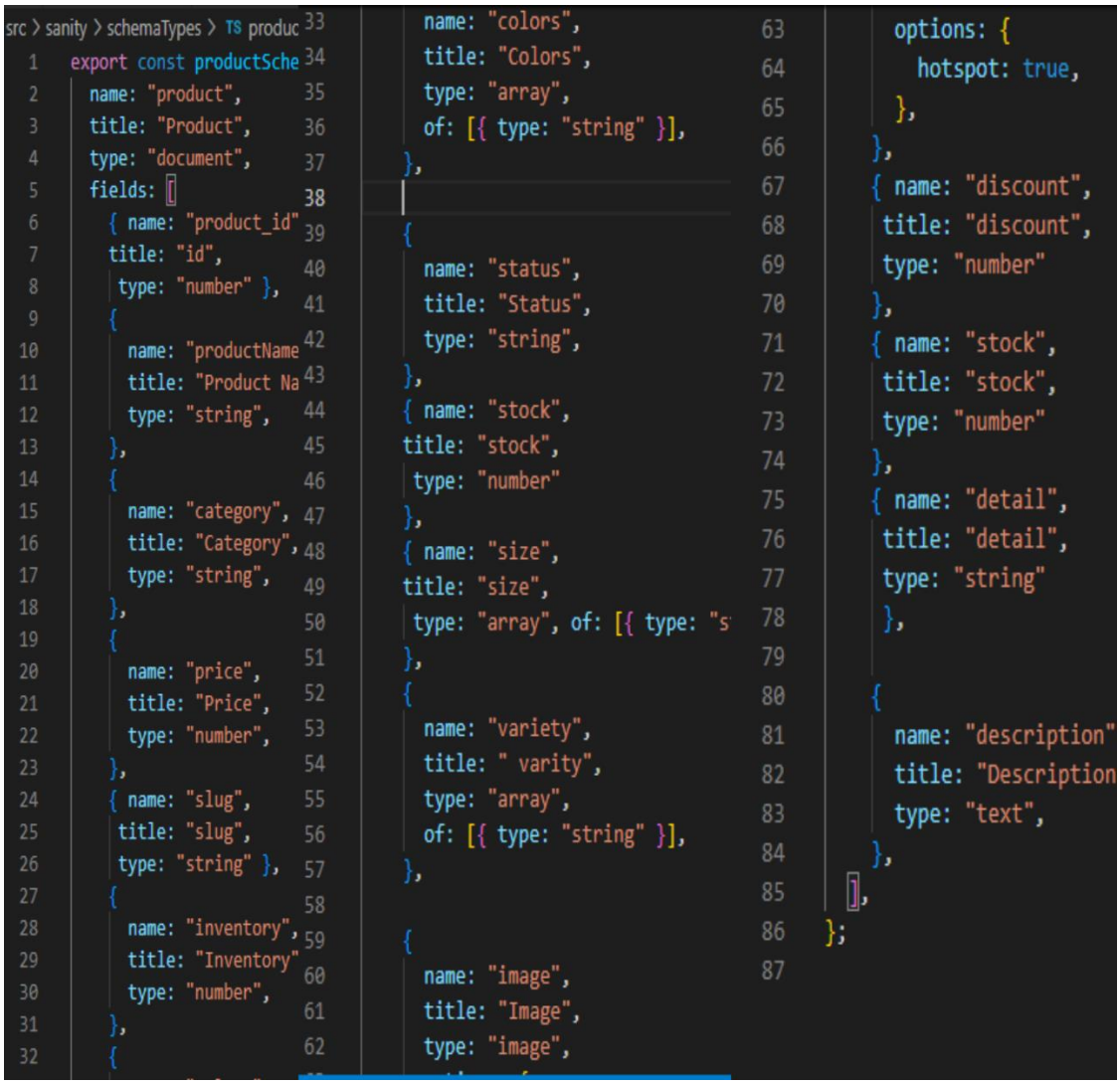
Category: Men's Running Shoes
Colors: Gray
Status: Best Seller

$9795

Add to Cart

3. Schema Enhancements in Sanity CMS:

- Updated the product schema to align with the requirements of the application, ensuring it could handle new fields and validations.
- **Fields Added/Updated:**
    - **inventory**: Tracks stock levels for products.
    - **colors**: Stores available color options as an array.



Self-Validation Checklist:

API Understanding:	Schema Validation:	Data Migration:	API Integration in Next.js	Submission Preparation:
✓	✓	✓	✓	✓

Outcomes of Day 3:

- Successfully integrated **Sanity CMS** with the application.
- Implemented dynamic **GROQ queries** to fetch and display product data.
- Enhanced the Sanity schema to support extended product attributes.
- Migrated data and images from an external API to Sanity CMS.
- Validated and tested the integration to ensure seamless backend-to-frontend data flow.

This concludes my Day 3 Hackathon work by **Mehwish Sajjad** I will continue to work diligently to turn these features into reality for my e-commerce website