Day 3 - API Integration and Data Migration
Sit & Style Studio
Date: 17-1-25
Name: Syeda Hafiza Bibi Amna
Project: API Integration and Data Migration

Overview:
The focus on Day 3 was centered around integrating APIs, particularly with Sanity CMS, by using GROQ queries and updating the schema to accommodate dynamic data needs. Below is a summary of the tasks completed:

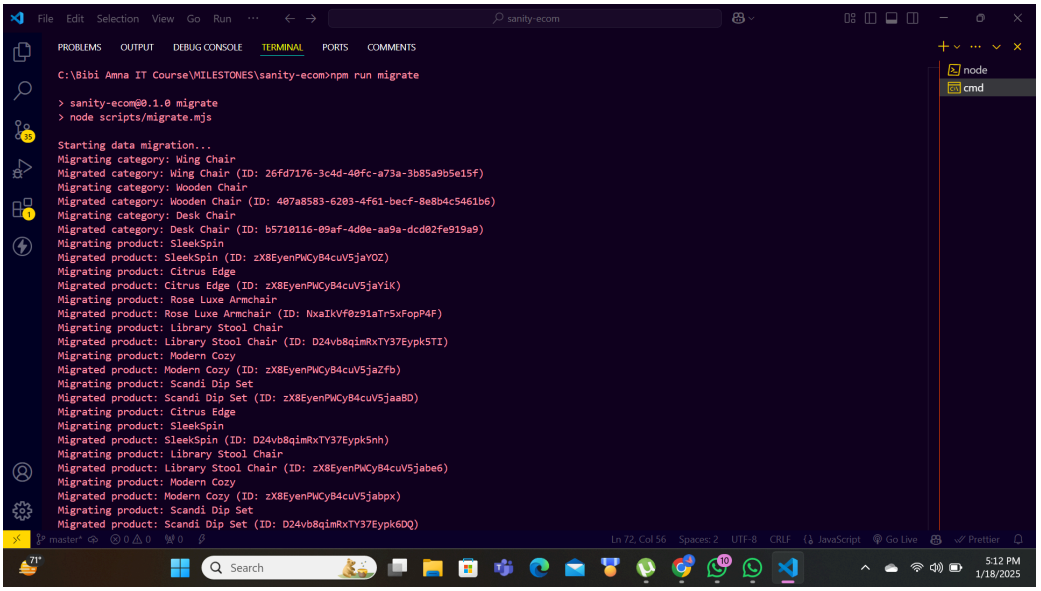
Key Objectives:

1. Establish a connection with Sanity CMS for efficient product data management.
2. Use GROQ queries to dynamically fetch structured data from Sanity.
3. Update the Sanity schema to meet evolving API requirements.
4. Migrate product data, including images, from an external API to Sanity CMS.
5. Test and validate the integration for smooth data flow between backend and frontend.

---

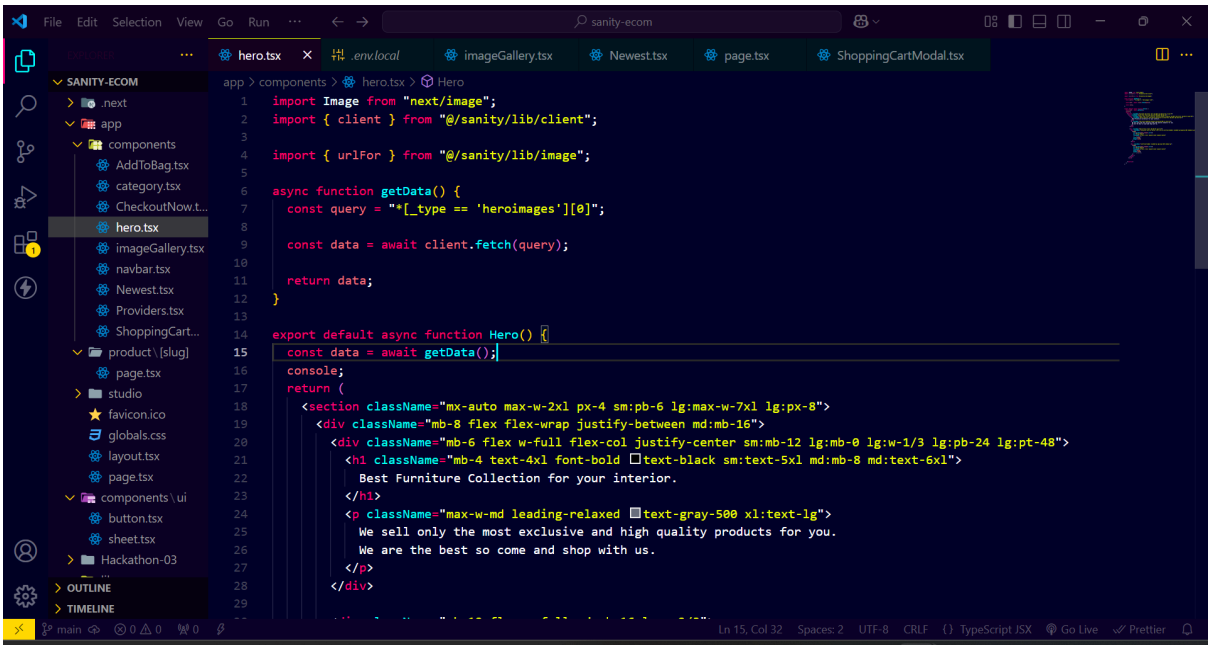
1. API Data Migration to Sanity

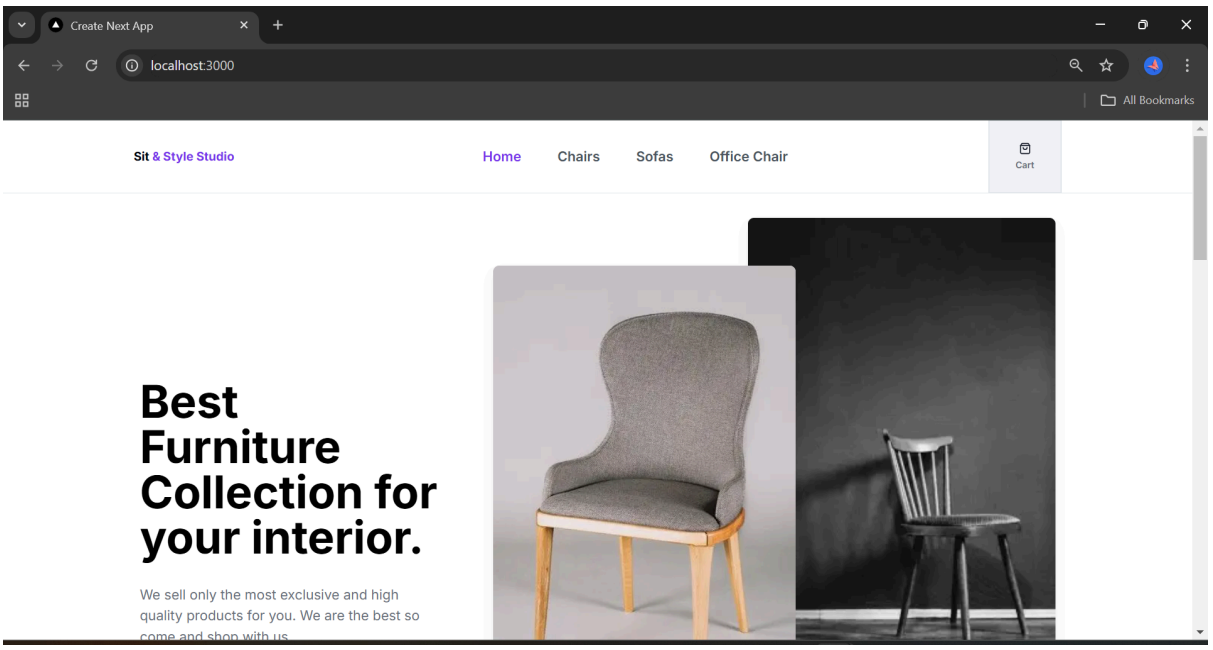
- **Objective**: Migrate product data (including images) from an external API to Sanity CMS.
- **Steps Taken**:
    1. **Fetch Product Data**:
        - Utilized Axios to fetch product data from an external API.
        - Example API Endpoint: https://template03api.vercel.app/api/products.
    2. **Upload Images to Sanity**:
        - Used client.assets.upload to upload product images to Sanity.
        - Linked each image to its corresponding product.
    3. **Create Product Documents in Sanity**:
        - Iterated through the fetched data and created product documents in Sanity using client.create().

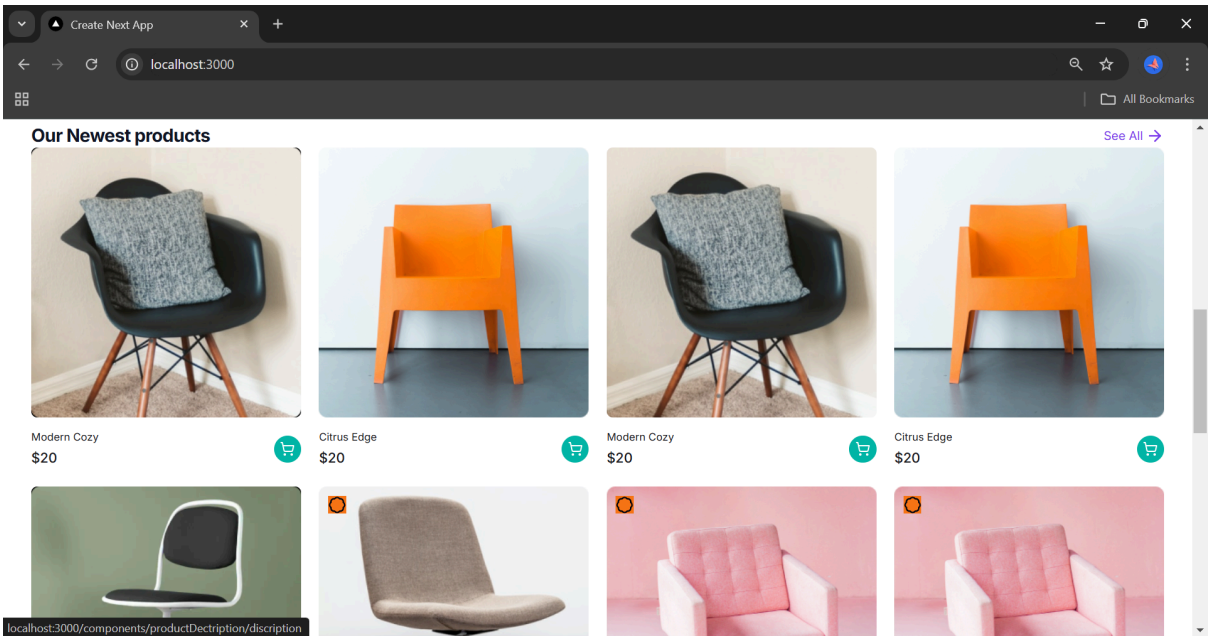


2. Sanity CMS Integration

- **Connection Setup**:
  - Integrated the @sanity/client library to enable communication with Sanity CMS.
  - Configured the client with project-specific credentials (projectId, dataset, apiVersion).
- **Purpose**: Centralized product management by leveraging Sanity's features, including real-time updates, robust content modeling, and optimized image handling.

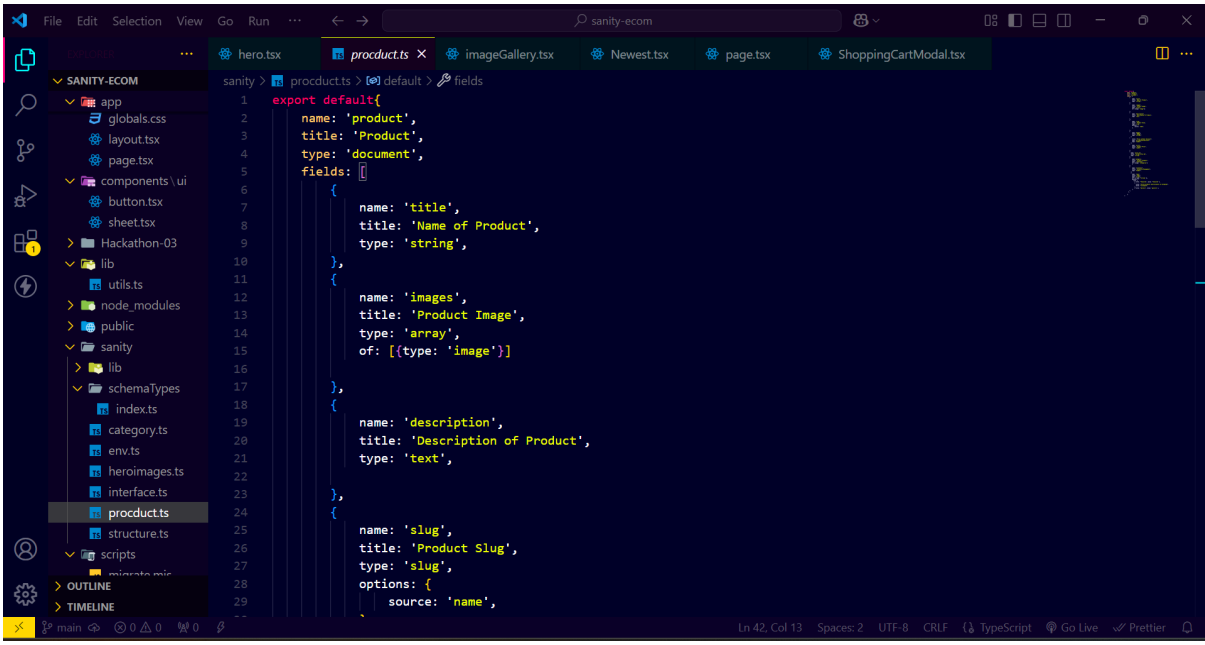






3. Schema Enhancements in Sanity CMS

- Updated the product schema to ensure it could accommodate new fields and validations.
- **Fields Added/Updated**:
    - **inventory**: Tracks product stock levels.
    - **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 into the application.
- Implemented dynamic GROQ queries to retrieve and display product data.
- Enhanced the Sanity schema to accommodate extended product attributes.
- Migrated product data and images from an external API to Sanity CMS.
- Validated and tested the integration to ensure seamless data flow from backend to frontend.

This concludes my Day 3 Hackathon progress. I will continue to work diligently to bring these features to life for my e-commerce website.