Name: Muhammad Hammad

Father Name: Muhammad Siddique

Roll No: 00413850

Days/Time: Saturday -07:00PM – 10:00PM

Day 3 - API Integration Report – Furniture E-commerce Marketplace

Step 1: Understand the Provided API

Tasks Completed:

Key Endpoint Identified:

1. **/api/product: This endpoint provides product listings with details like name, category, price, stock level, discount percentage, and image paths.**

1. **Used the import-data.mjs script from Template 4 as a base.**
2. **Made two significant changes to adapt it for Template 0 API:**
    - **API URL Updated:**
      Changed the API endpoint to:

const response = await axios.get("https://template-0-beta.vercel.app/api/product");

    - **Response Structure Adjustment:**
      The API response for Template 0 provides an array of objects directly, not nested under products. Updated the script accordingly:

const products = response.data;

3. **Verified the API response structure by testing with the sample data, which included product attributes like id, name, imagePath, price, description, discountPercentage, isFeaturedProduct, stockLevel, and category.**

Outcome:

- **The script now fetches data correctly from Template 0's API and uploads it seamlessly to Sanity CMS, including product metadata and images.**

Step 2: Validating and Adjusting the Sanity Schema

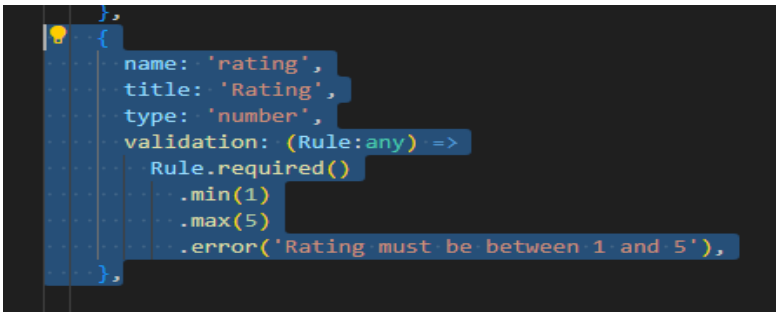
Tasks Completed:

1. **Reviewed the Sanity CMS schema created on Day 2 to ensure it aligns with the API data structure for Template 0.**
2. **Adjustments Made:**

- **Updated the category field from a simple string to an options list for better consistency and user experience:**

```
{
  name: 'category',
  type: 'string',
  title: 'Category',
  options: {
    list: [
      { title: 'Chair', value: 'Chair' },
      { title: 'Sofa', value: 'Sofa' },
      { title: 'Table', value: 'Table' },
      { title: 'Bed', value: 'Bed' },
    ],
  },
  validation: (Rule: any) => Rule.required().error('Category is required'),
},
],
```

- **Added a new field, rating, to the Sanity CMS schema to enhance the product details for Template 0.**



Step 3: Data Migration

Task: Migrating the product data from the external API into Sanity CMS using a migration script.

Migration Steps:

1. **Preparing the Migration Script:**
    - **A Node.js script was created to automate the migration process, making it efficient and scalable. This script used Sanity's client to interact with the CMS and create documents for each product retrieved from the external API.**

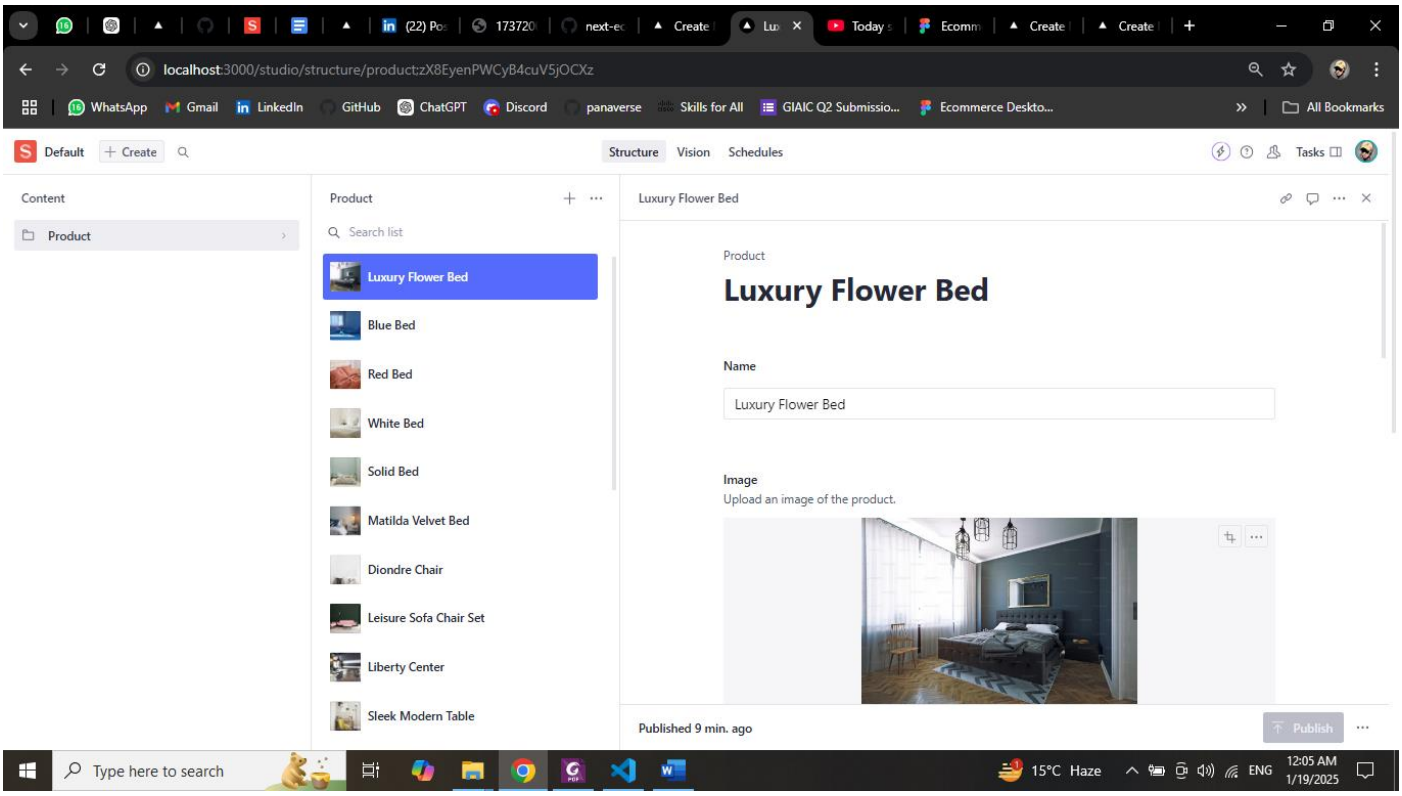
Script to Migrate Product Data: 1. The script iterated over the product data and pushed it to Sanity's CMS using the following code:

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

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

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2025-01-15',
  useCdn: false,
});

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, {
```

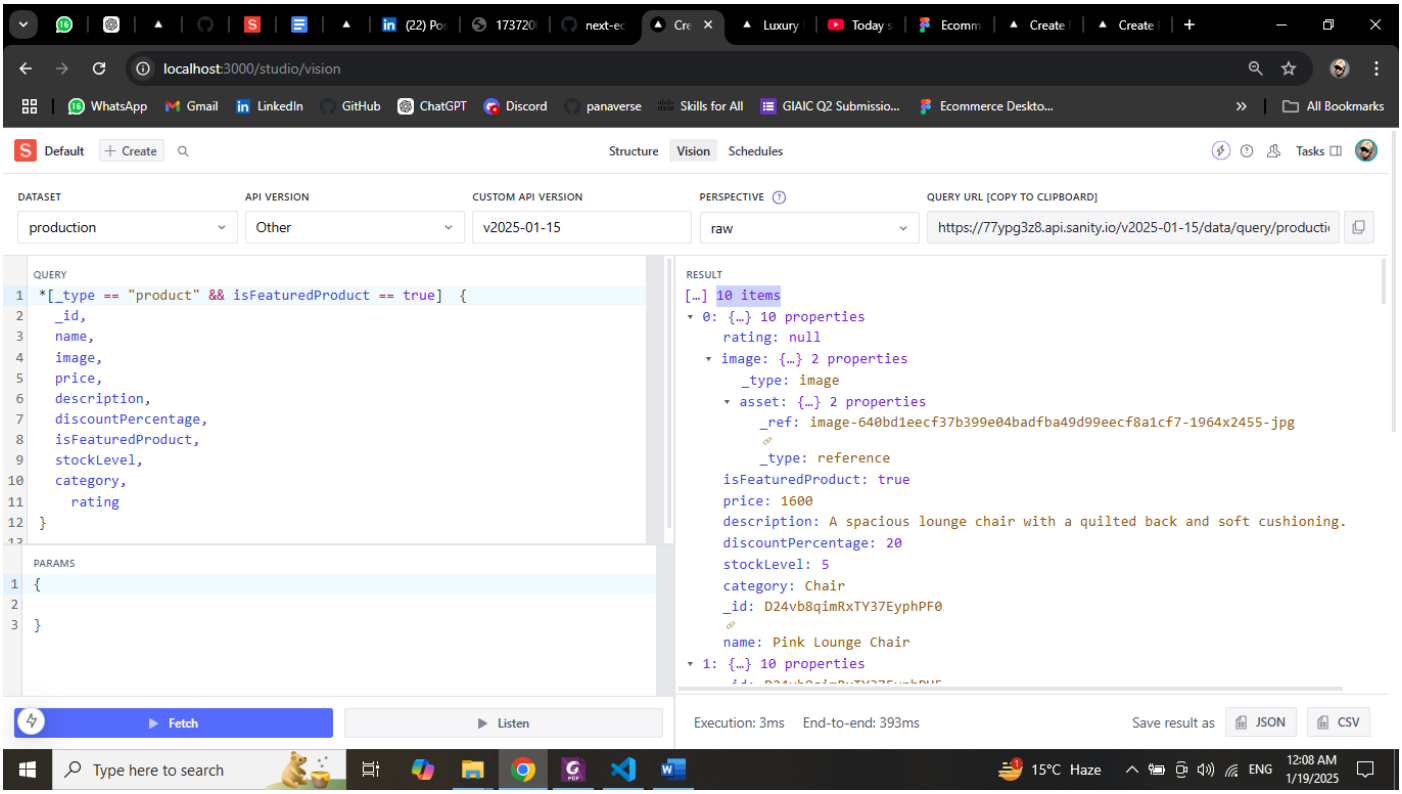
```
      filename: imageUrl.split('/').pop(),
    });
    console.log(`Image Uploaded Successfully : ${asset._id}`);
    return asset._id;
  }
  catch (error) {
    console.error('Failed to Upload Image:', imageUrl, error);
    return null;
  }
}
async function importData() {
  try {
    console.log('Fetching Product Data From API ...');

    const response = await axios.get("https://template-0-beta.vercel.app/api/product")
    const products = response.data;

    for (const item of products) {
      console.log(`Processing Item: ${item.name}`);

      let imageRef = null;
      if (item.imagePath) {
        imageRef = await uploadImageToSanity(item.imagePath);
      }
      const sanityItem = {
        _type: 'product',
        name: item.name,
        category: item.category || null,
        price: item.price,
        description: item.description || '',
        discountPercentage: item.discountPercentage || 0,
        stockLevel: item.stockLevel || 0,
        isFeaturedProduct: item.isFeaturedProduct,
        image: imageRef
          ? {
              _type: 'image',
              asset: {
                _type: 'reference',
                _ref: imageRef,
              },
            }
          : undefined,
      };

      console.log(`Uploading ${sanityItem.category} - ${sanityItem.name} to Sanity !`);
      const result = await client.create(sanityItem);
      console.log(`Uploaded Successfully: ${result._id}`);
      console.log("-----------------------------------------------------")
      console.log("\n\n")
    }
    console.log('Data Import Completed Successfully !');
  } catch (error) {
    console.error('Error Importing Data : ', error);
  }
}
```

importData();

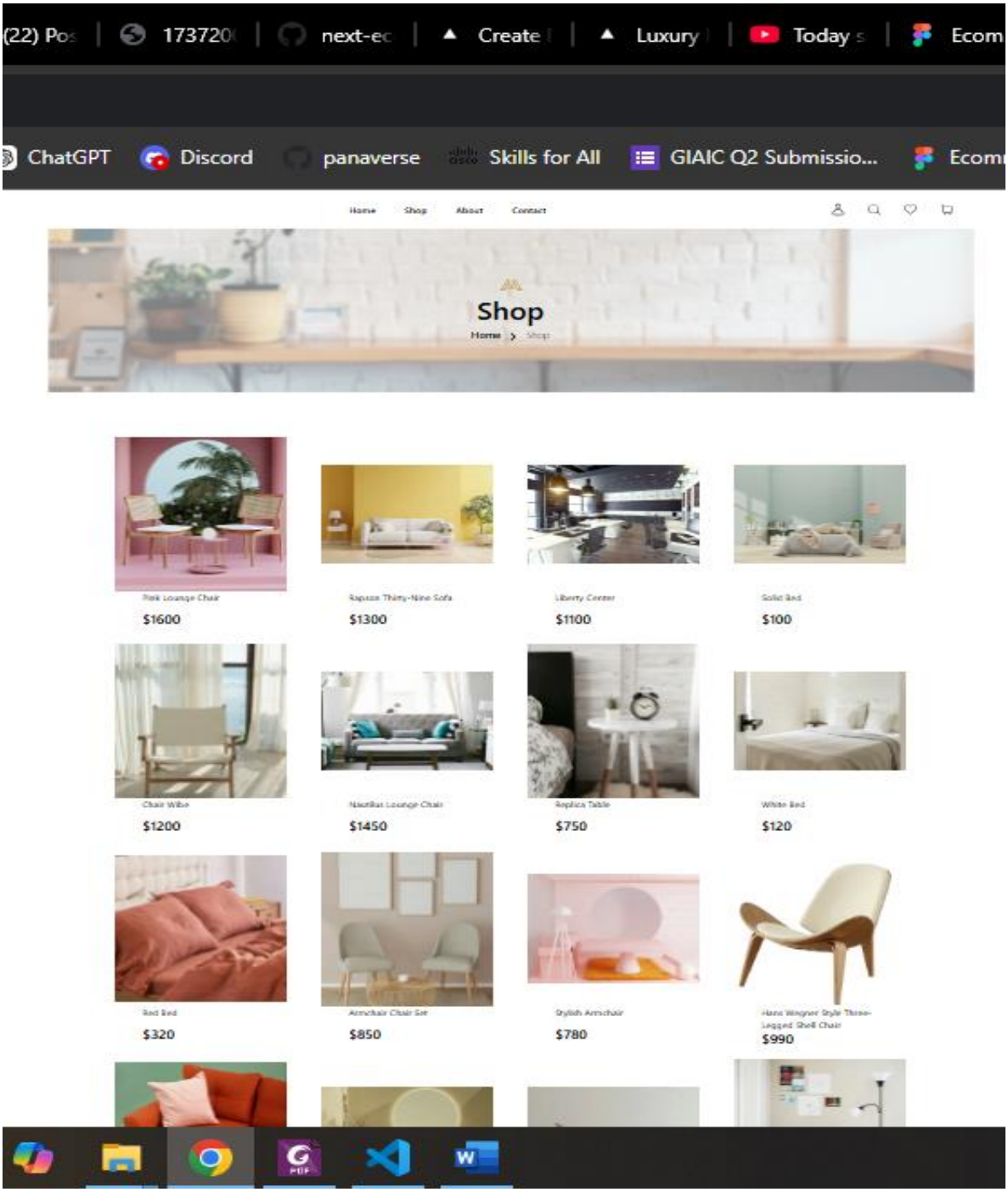
Verification: After migration, the data was verified in the Sanity Studio interface, ensuring that all product data, including pricing, description, and categories, were correctly added.



Add a screenshot demonstrating the usage of GROQ queries to fetch product data in vision ,



Above is a screenshot of the Shop page, where the product data is successfully displayed on the UI using the GROQ query from Sanity.



Conclusion:

The API integration process was successfully carried out, allowing the product data to be fetched, stored, and displayed on the frontend seamlessly. Key modifications were made to the Sanity CMS schema to support the new product details. The migration of data was automated using a custom script, ensuring a smooth and efficient transition from the external API to the CMS.