Day 3 - API Integration Report

General Ecommerce
Prepared By: Muzamil Ahmed

API Integration Process Overview:

This document outlines the API integration process, focusing on fetching and migrating data from an external API into **Sanity CMS** for the marketplace backend. The external API used is available at: https://hackathon-apis.vercel.app/api/products.

Steps Followed:

1. Understanding API Documentation:

- Reviewed the provided external API endpoints.
- Identified key endpoints such as product listings and categories.

2. Authentication Setup:

- The API did not require authentication, simplifying the integration process.

3. Endpoint Testing:

- Tested API calls using **Postman**.
- Verified the structure and accuracy of the data returned.

4. Backend Integration:

- Implemented functions in the backend to fetch and process API data.
- Established reusable utility functions for querying **Sanity CMS**.

Adjustments Made to Schemas:

1. Sanity Schema Updates:

- Adjusted the product schema to include fields such as:
  - Tags
  - Dimensions
  - Category

- Added a **category schema** to manage product categories.

2. Field Mappings:

- **API Field:** `title` → **Schema Field:** `name`
- **API Field:** `price` (number) → **Schema Field:** `price` (float)
- **API Field:** `image` → **Schema Field:** `image.asset` (reference)

3. Relationships:

- Linked **category** in the product schema as a reference field to the **category schema**.

Migration Steps and Tools Used:

Tools:

- **Axios:** For making API calls.
- **Sanity Client:** For interacting with the Sanity CMS.
- **Slugify:** For generating slugs from product names.

Steps:

1. **Fetch Data from API:**
   - Used Axios to retrieve product data from the external API.
2. **Data Transformation:**
   - Mapped API data fields to match the Sanity schema.
   - Created a utility function to upload images to Sanity and return their references.
3. **Sanity Data Upload:**
   - Iterated over the transformed data to create or update records in Sanity.
4. **Verification:**
   - Verified imported data by querying it via the **Sanity CMS Studio**.

Challenges Faced and Solutions:

1. Data Format Mismatch:

- **Challenge:** The API returned inconsistent field names compared to the Sanity schema.
- **Solution:** Used a mapping layer in the migration script to standardize field names.

2. Image Upload Errors:

- **Challenge:** Timeout issues occurred while uploading large images.

- **Solution:** Implemented error handling and retries for image uploads.

3. Data Duplication:

- **Challenge:** Duplicate entries in the API response.
- **Solution:** Added logic to check for existing records in Sanity before creating new ones.

Screenshots:

1. **API Calls:**
   - Screenshot of successful GET request to the external API.
2. **Frontend Data Display:**
   - Screenshot showing products rendered in the marketplace frontend.
3. **Sanity CMS:**
   - Screenshot of products and categories populated in Sanity CMS.

Code Snippets:

- **API Integration Functions**
- **Migration Script**

Summary:

This report provides a comprehensive overview of the API integration and data migration process for the marketplace backend. By utilizing tools such as **Axios**, **Sanity CMS**, and **Slugify**, we successfully:

- Imported data from an external API.
- Transformed it to match the Sanity schema.
- Verified its integrity.

The reusable backend functions and scripts ensure scalability and flexibility for future development. This workflow demonstrates a practical approach to real-world API integration and CMS management challenges, equipping the team with essential skills for dynamic marketplace creation.