

## **Day3 of ”marketplacebuilderhackathon“**

### **2. API Integration Process**

#### **Overview**

The goal of this task was to integrate external APIs into our marketplace application to facilitate data retrieval, manipulation, and display. The integration enhances our system's functionality by automating tasks and streamlining operations.

#### **Steps Involved:**

### **3. Adjustments Made to Schemas Product Schema Updates**

#### **1. Identifying API Requirements**

\* The first step involved identifying the key functionalities required for the marketplace (e.g fetching product data, processing transactions).

- \* Chose APIs based on data needs and ensured they met the marketplace's requirements.

## **2. Integrating APIs**

- \* Installed necessary libraries (e.g., axios, fetch) to handle API requests in the application.

- \* Set up environment variables to securely store API keys and credentials.

- \* Developed functions to fetch data from the external API endpoints and map it to the marketplace data structure.

## **3. Handling Data in the Application**

- \* The data fetched via the API was integrated into the existing components of the marketplace.

- \* Used state management (e.g., React's useState) to store API data and display it dynamically on the frontend.

## **4. Error Handling & Debugging**

- \* Implemented error handling to manage failed API requests or missing data.

- \* Logs were set up to track API response status and pinpoint potential issues.

- \* Added New Fields for API Data Custom Data Types

## **5. Adjustments Made to Schemas Product Schema**

Updates

### **\* Added New Fields for API Data**

Introduced new fields such as `image_url`, `tags`, and `price_per_day` to the product schema accommodate the data fetched from the external API. Example

Example: javascript CopyEdit { name: 'image\_url', type: 'string', title: 'Product Image URL' }

Revised Existing Fields Updated the price field to handle different pricing models (e.g., hourly, dai Example: javascript CopyEdit { name: 'price\_per\_day', type: 'string', title: 'Price per Day' }

# **Custom Data Types**

## **4. Migration Steps and Tools Used**

Migration Process:

### **1. Database Backup**

Prior to initiating the migration, a full backup of the existing database was performed to avoid data loss.

### **2. Data Mapping and Transformation**

Data from the external API was mapped to the existing database schema using a script that matched API fields with the marketplace data file

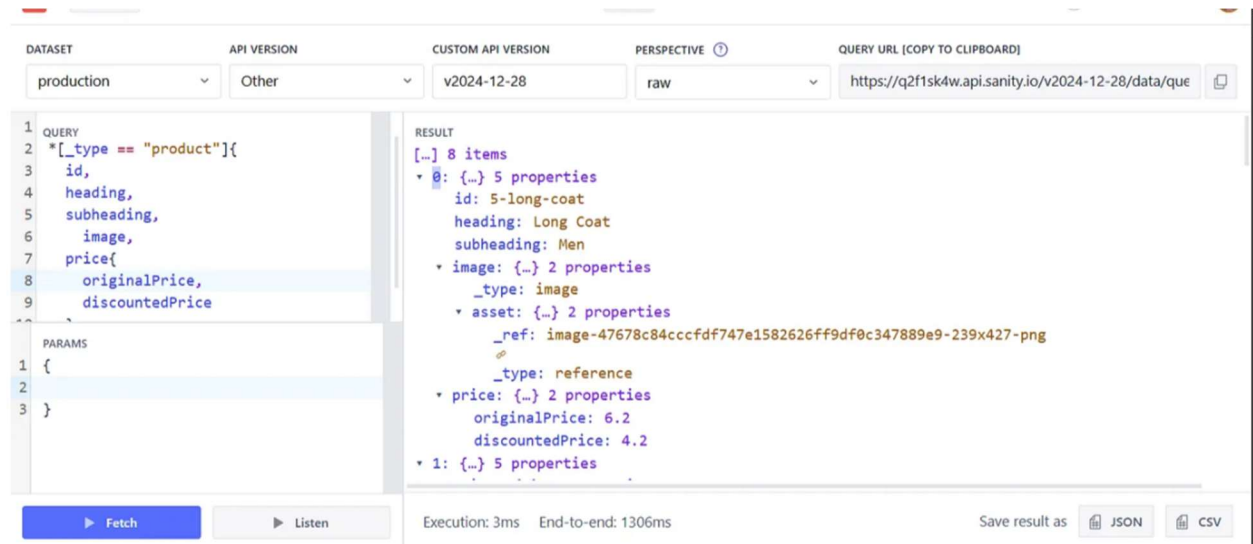
### **3. Data Insertion The transformed**

data was inserted into the marketplace's database, ensuring that the new fields were populated accurately. For large data sets, batch processing was used to minimize performance issues.

## 4. Validation After the migration

all product data was validated to ensure it was correctly formatted and visible on the frontend

Code snippets for API integration and migration scripts



Output after fetch display product in frontend



## Long Coat

Men

★★★★☆ (10 Reviews)

**\$4.2** ~~\$6.2~~

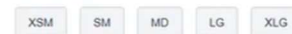
Availability: **In Stock**

Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquam possimus cum quidem tempora sint et eaque, temporibus alias provident hic nobis sed velit voluptate voluptatibus, quia ullam esse eligendi vitae.

Color Family:



Size Family:



Select Options



**”Develop By”**

Name : ”Muhammad Owais Shah”

Class : ”Saturday 2pm to 5pm”

Roll : 00183255