DAY 3

API INTEGRATION AND DATA MIGRATION

Name: Farah Shabir

Roll No:00006855

Time: 9:00 AM to 12:00 PM

Slot: Sunday

# Day 3 - API Integration and Data Migration Report - Bandage

## Objective:

On Day 3, the focus was on integrating API data into the Sanity CMS for the Bandage project to enable dynamic product updates for the marketplace. This integration provided a more efficient and scalable alternative to manual data entry, ensuring that product content remains up-to-date effortlessly.

# Sanity CMS Schema Design:

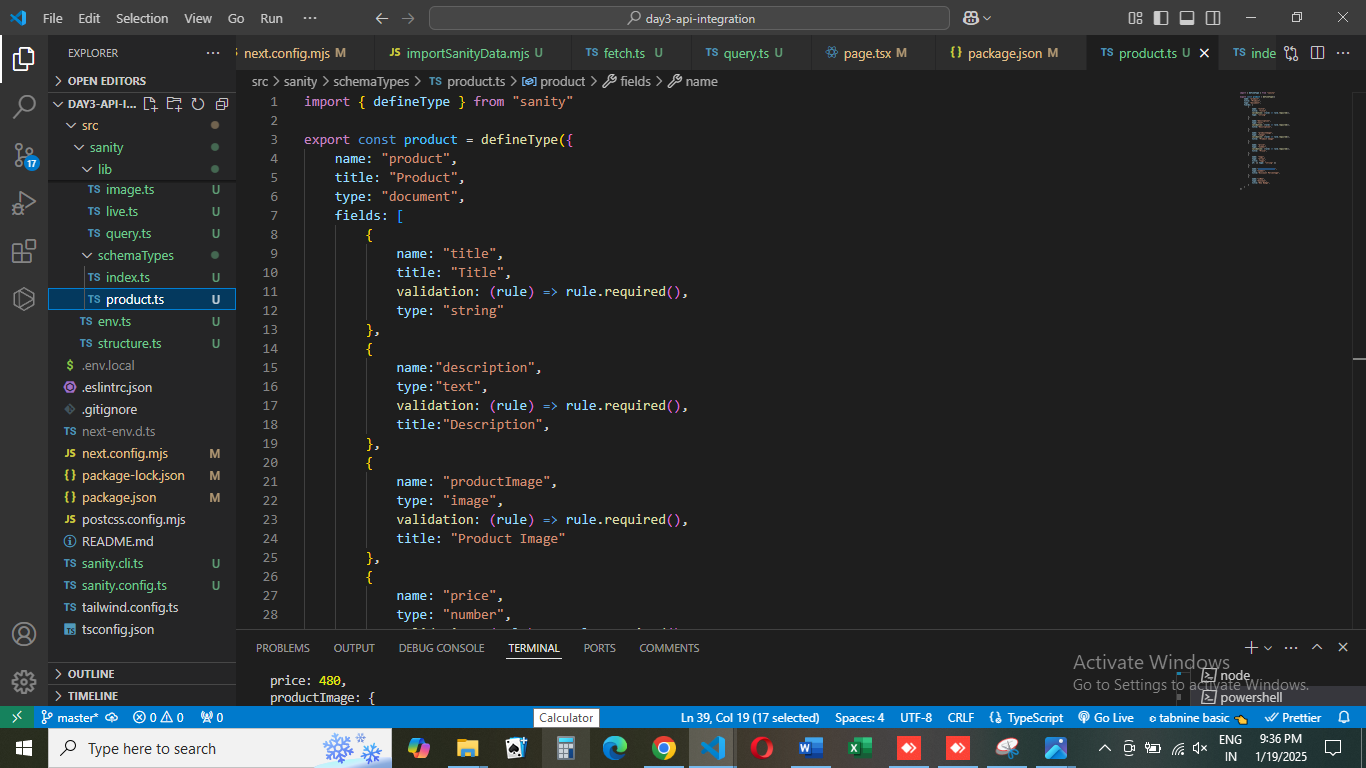
To facilitate seamless handling of product data, I designed a schema named product in Sanity CMS. The schema consists of the following fields:

**Main Fields:**

* **title**: The product title (string type).
* **description**: A detailed description of the product (text type).
* **productImage**: The main product image (image type).
* **price**: The price of the product (number type).
* **tags**: An array of tags to categorize the product (array of strings).
* **discountPercentage**: The discount percentage applied to the product (number type).
* **isNew**: A boolean flag indicating whether the product is new (boolean type).

This schema ensures that all necessary product information is captured and properly structured within the Sanity CMS, making it easy to manage and update products dynamically.

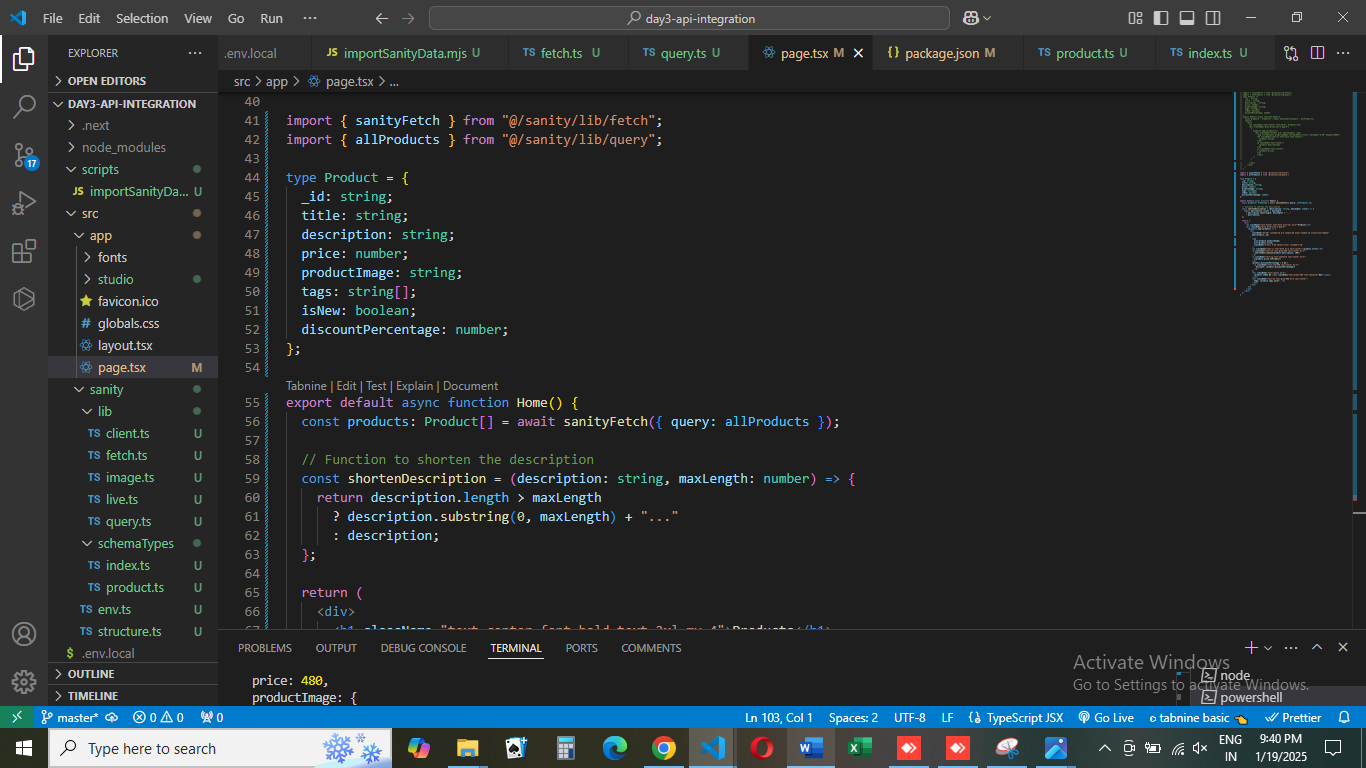
**Code Snippet:**

****

# API Integration and Data Migration:

## API Data Fetching:

I retrieved product data from an external API, including images, titles, descriptions, prices, and tags. This data was then mapped to the appropriate fields in the Sanity CMS schema, enabling smooth integration and dynamic content management.

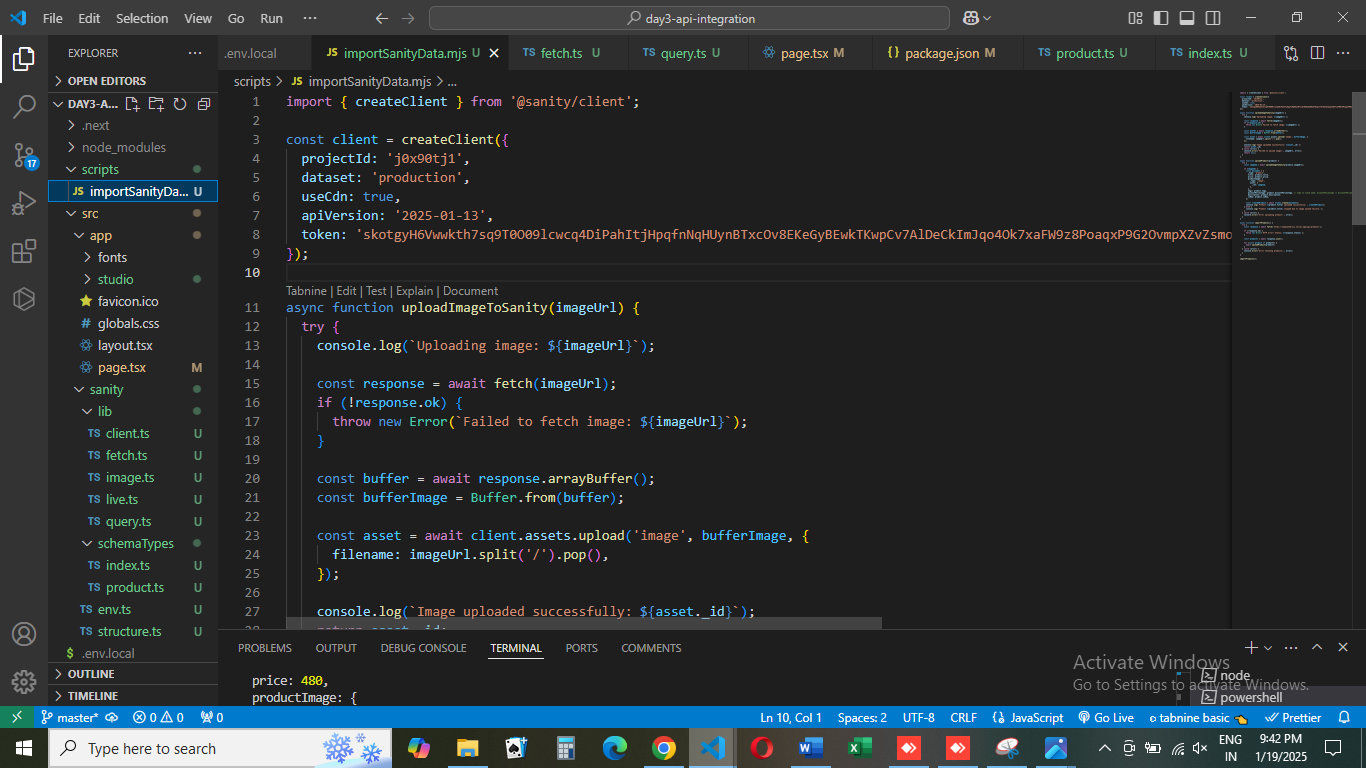


## Data Population in Sanity CMS:

## After fetching the API data, I dynamically populated the product fields in Sanity CMS. This automated the process of filling in product information, ensuring consistent and accurate data across the platform.

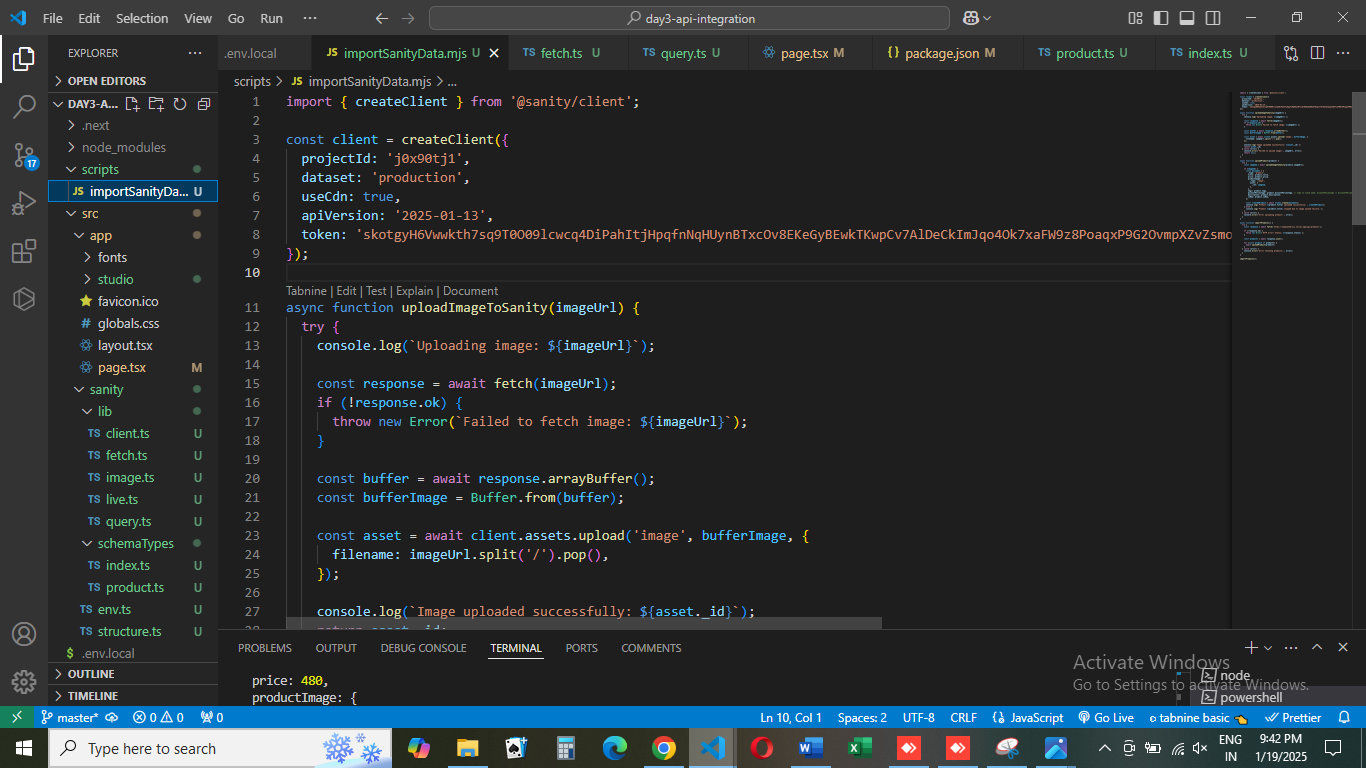
## Data Migration:

Using the Sanity CLI, I exported the dataset from Sanity CMS for backup and later re-imported it for testing purposes. This migration process ensured that all the data was correctly structured and displayed as expected on the frontend.



# Steps Taken for Data Migration:

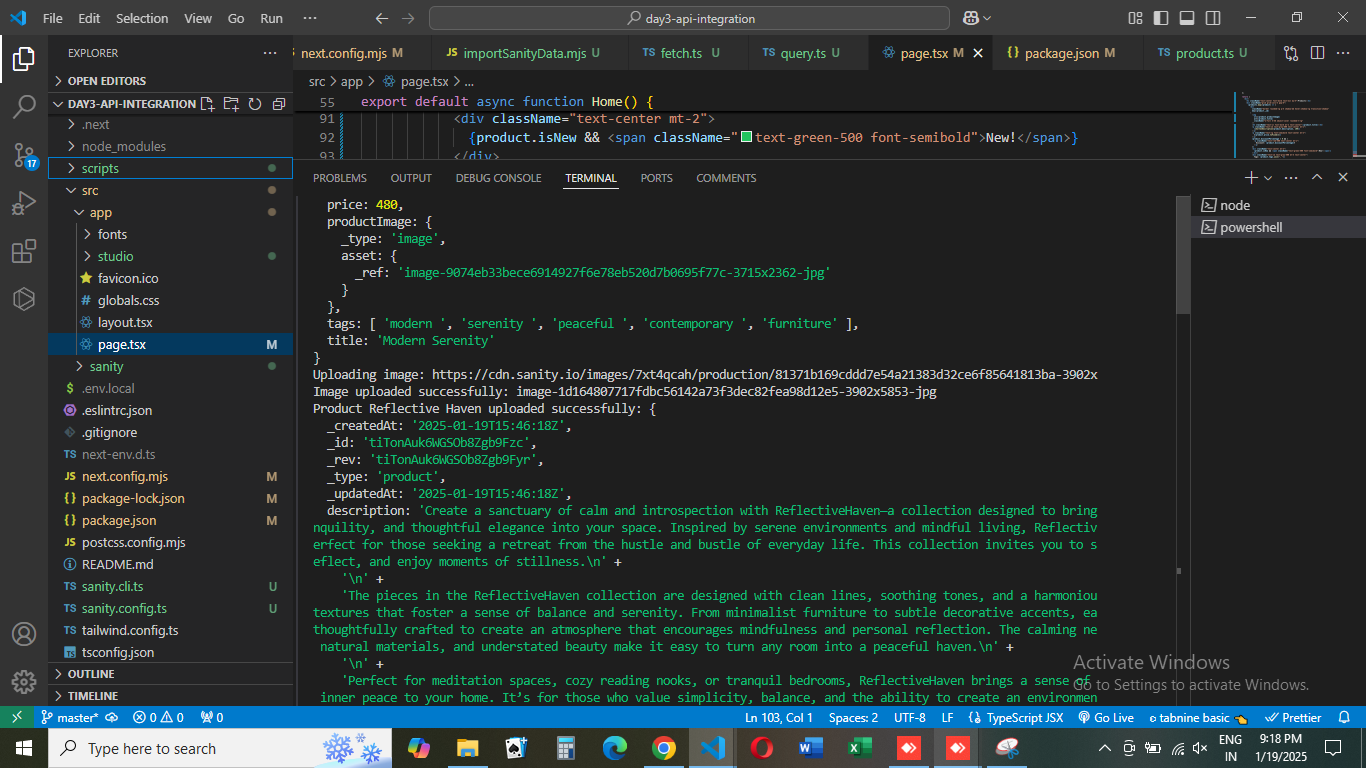
## Exporting Data:

The initial step was to export the data from Sanity CMS using the Sanity CLI. This action created a secure backup of all product data, ensuring its safety before continuing with any subsequent operations.

## Verification of Data:

The exported JSON structure was reviewed to ensure that all fields were populated correctly. This step ensured that the data would be accurately displayed when fetched and rendered on the frontend.

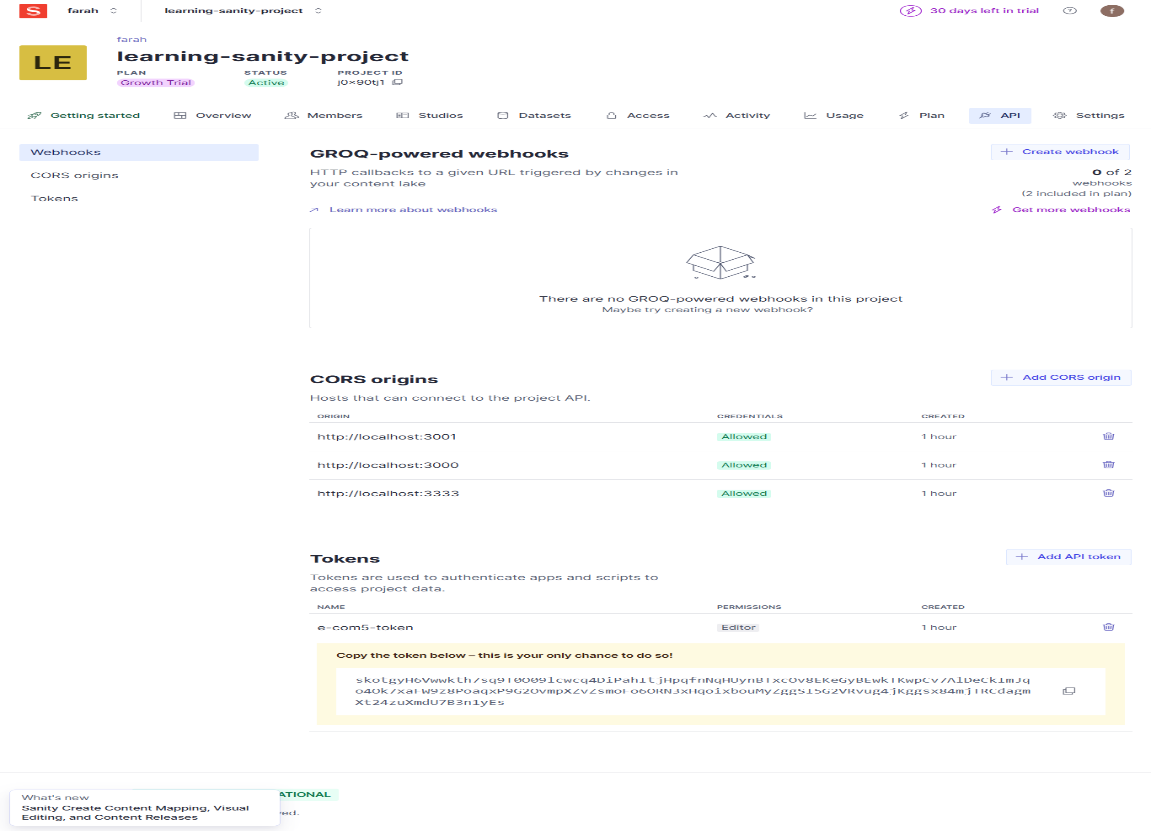
## Re-importing Data:

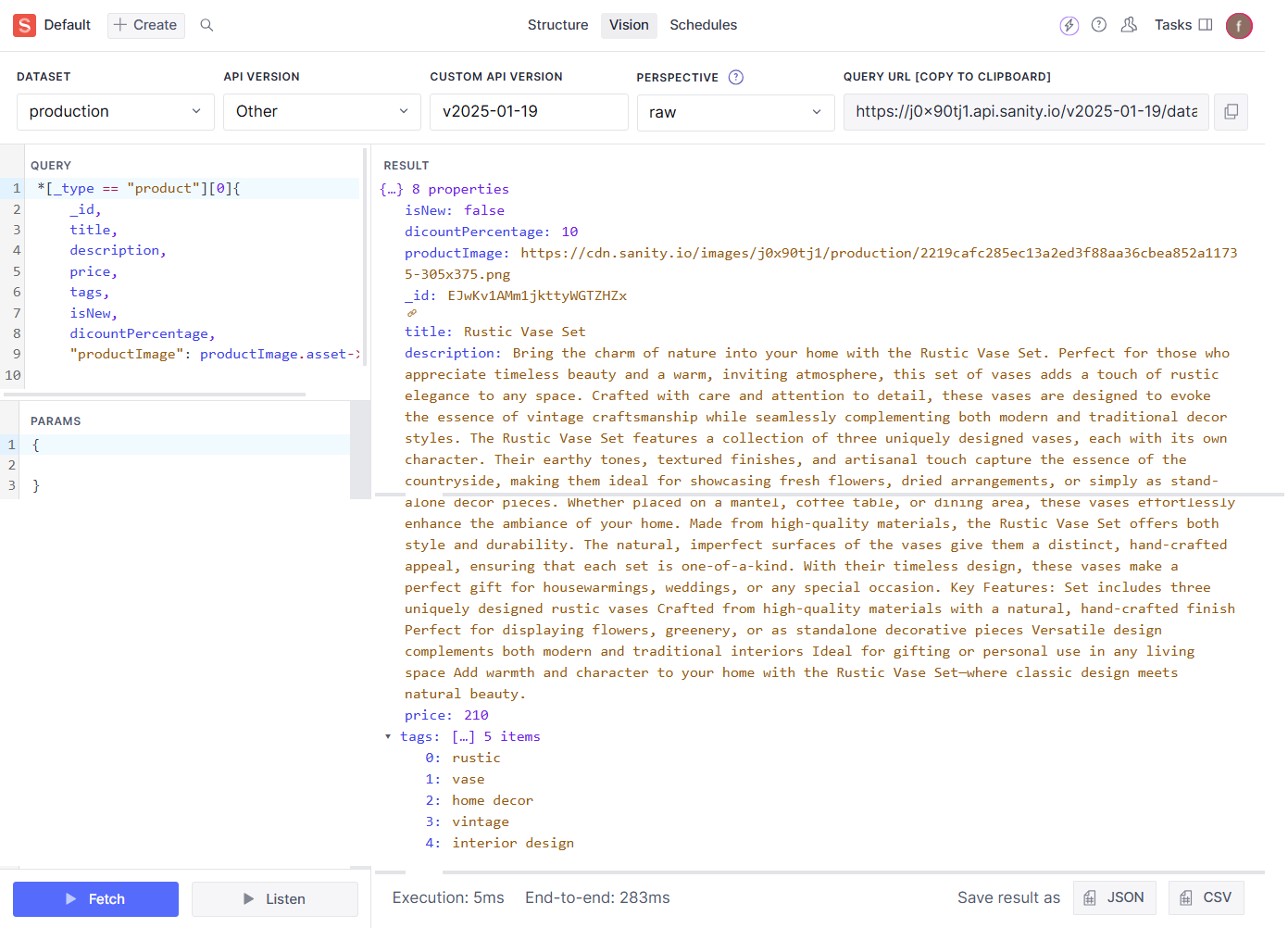
After verification, the dataset was re-imported into Sanity CMS. This confirmed that the data migration was successful and the system was working as expected.

# Tools Used:

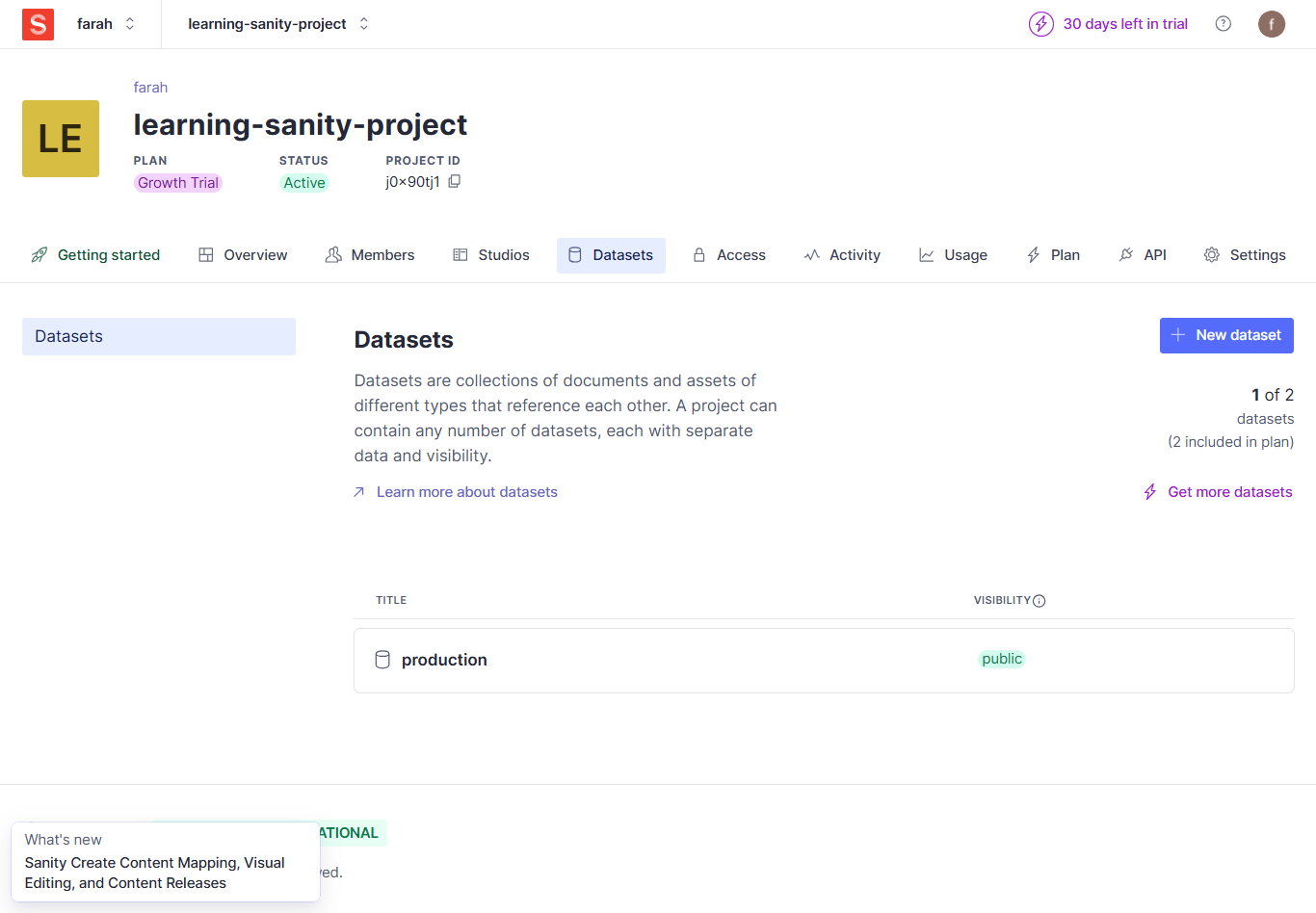
## Sanity Studio:

Used for schema creation, content management, and displaying product data.



**Sanity Vision**

**Sanity Dataset:**

****

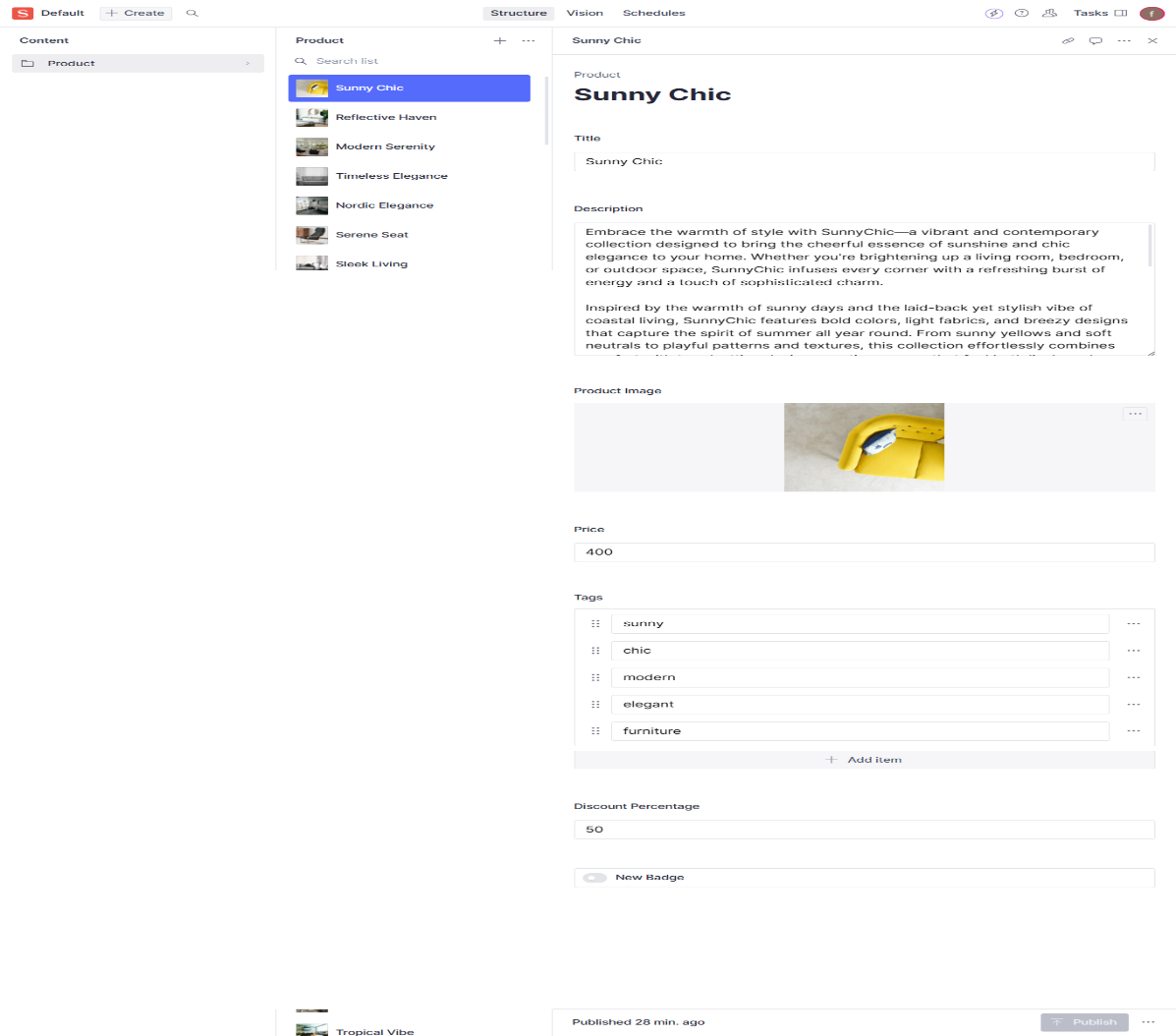
**Sanity CLI:**

Utilized for exporting and importing the dataset, ensuring data consistency and backup.

# Screenshots and Frontend Display:

## Sanity CMS Fields:

Screenshot showing the populated fields in Sanity Studio, displaying product details like images, descriptions, and prices.



## Frontend Display:

* Here’s a simple explanation of how the product data is displayed dynamically on the frontend of the marketplace:
* The product details, such as image, title, description, price, and tags, are fetched from the Sanity CMS and displayed in a clean, structured format.
* Each product appears in a card layout, showing the product image, title, and price, with a brief description and any relevant tags or discounts.
* The data is automatically updated whenever changes are made in Sanity CMS, ensuring the marketplace always reflects the most current information.

# 

# Conclusion:

The API integration and data migration were successfully executed, significantly enhancing the efficiency and scalability of the Bandage project. The integration streamlined the process of adding and updating product data in the marketplace, while the migration steps ensured data consistency and accuracy across the system. With this setup, the Bandage project is now more dynamic, scalable, and easier to maintain, offering a more seamless experience for both developers and users.