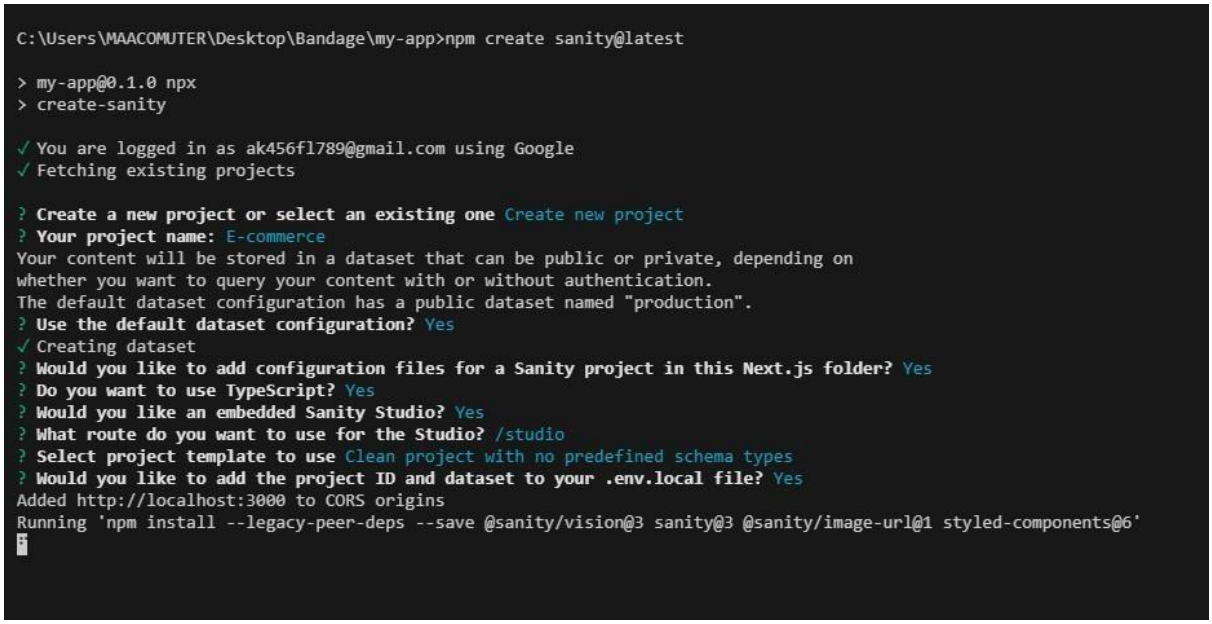
Name: Hammad Noor Khan

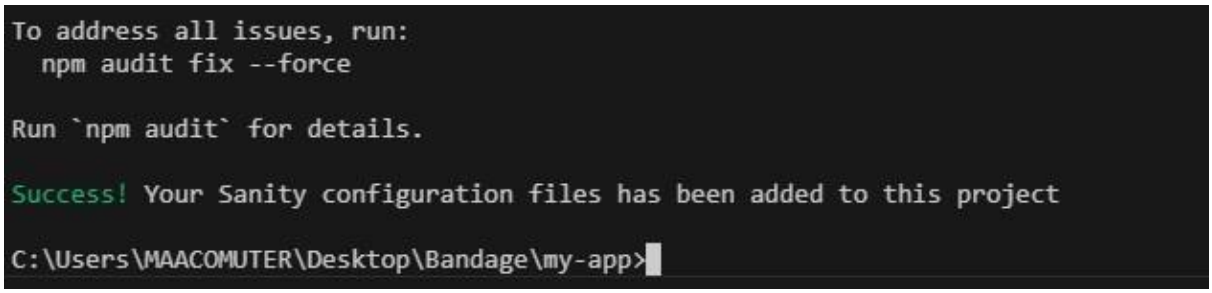
Roll No: 00457174

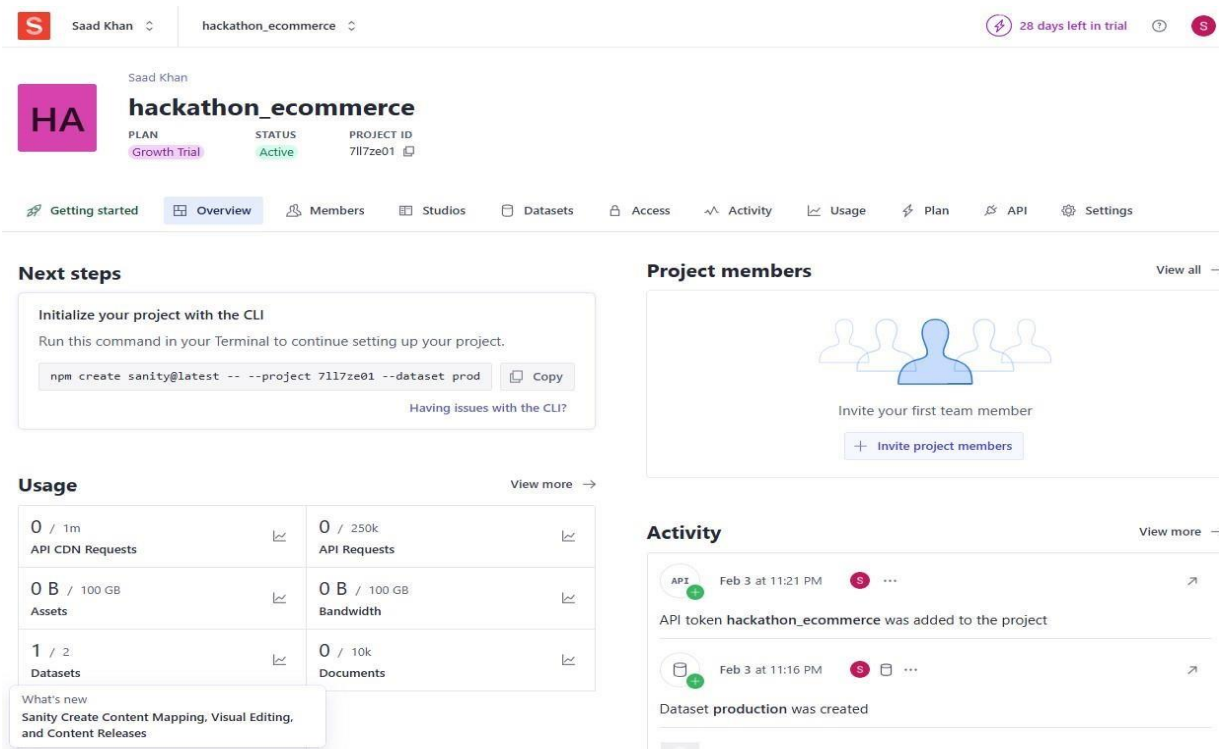
Day 3 - API Integration and Data Migration Report - [Hekto E-Commerce Marketplace]

Overview

This document summarizes the process I followed to integrate APIs and migrate data into the Sanity CMS while ensuring a functional Next.js frontend. The objective was to populate the CMS with accurate data, align schemas, and build a robust frontend for data display.







Process Overview 1. **Understanding the Provided API** ○

API Documentation Review: Identified key endpoints:

✦ /products: For product listings.

○ **Tools Used**: browser developer tools to test endpoints and responses.

2. **Schema Validation and Adjustments** ○ Compared the Sanity CMS schema with the API data structure.

○ **Example Adjustments**:

- **API Field**: slug **Schema Field**: slug

```
src > sanity > schemaTypes > product.ts > product
import { defineType } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        {
            name: "title",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        },
        {
            name:"description",
            type:"text",
            validation: (rule) => rule.required(),
            title:"Description",
        },
        {
            name: "productImage",
            type: "image",
            validation: (rule) => rule.required(),
            title: "Product Image"
        },
        {
            name: "price",
            type: "number",
            validation: (rule) => rule.required(),
            title: "Price",
        },
        {
            name: "tags",
            type: "array",
            title: "Tags",
            of: [{ type: "string" }]
        },
        {
            name:"dicountPercentage",
            type:"number",
            title:"Discount Percentage",
        },
        {
            name:"isNew",
            type:"boolean",
            title:"New Badge",
        }
    ]
})
```

Action: Added slug field to the schema to align with the API data and ensure proper URL routing.

```
src > sanity > schemaTypes > index.ts > schema > types
import { type SchemaTypeDefinition } from 'sanity'
import { product } from './Product'

export const schema: { types: SchemaTypeDefinition[] } = {
  types: [product],
}
```

3. **Data Migration Methods**
    - **Provided API**:

Wrote scripts to fetch, transform, and upload data into Sanity CMS. 4.

API Integration in Next.js

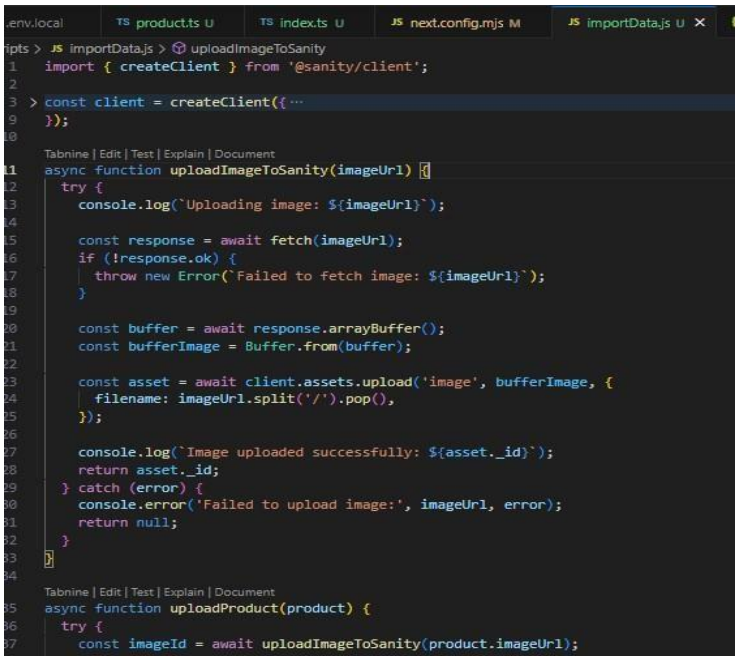
```
async function importProducts() {
  try {
    const response = await fetch('https://template6-six.vercel.app/api/products');

    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }

    const products = await response.json();

    for (const product of products) {
      await uploadProduct(product);
    }
  } catch (error) {
    console.error('Error fetching products:', error);
  }
}

importProducts();
```



Results

1. **Sanity CMS**: Successfully populated with data using automated and manual methods.
2. **Next.js Frontend**: Functional API integration displaying product listings and categories with fallback mechanisms.

Timeless Elegance

Product

Timeless Elegance

Title

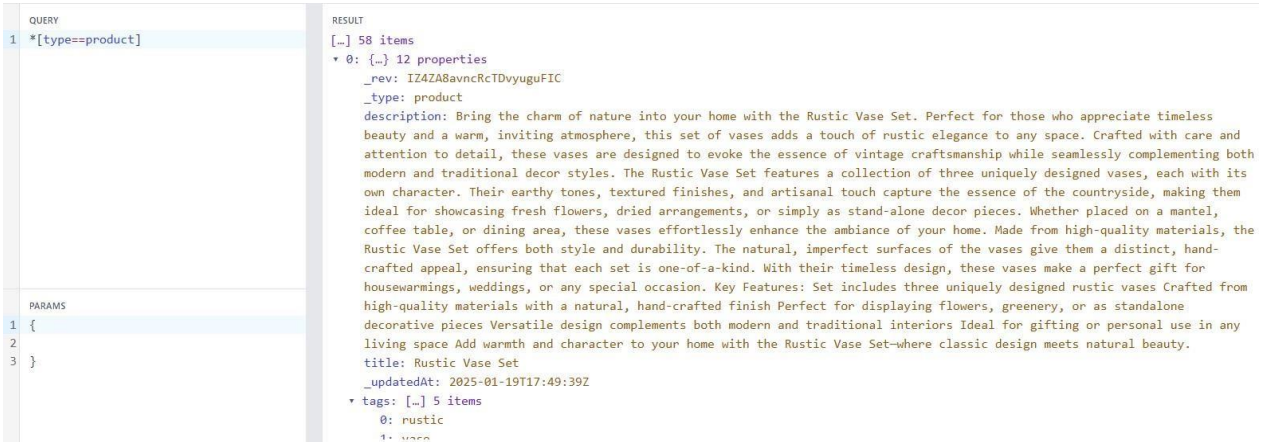
Timeless Elegance

Description

Introducing TimelessElegance—a collection that embodies the perfect fusion of classic beauty and modern sophistication. Designed for those who appreciate the enduring appeal of refined style, TimelessElegance brings grace, charm, and unparalleled quality to any space. Each piece in this collection is crafted to stand the test of time, offering not only lasting durability but also a sense of sophistication that never goes out of style.

With its clean lines, luxurious materials, and subtle detailing, TimelessElegance seamlessly complements any décor, from contemporary urban apartments to traditional homes. Whether you're adding a statement piece to your living room,

Product Image



Best Practices Followed

- **Sensitive Data Management**: Stored API keys in .env files.
- **Clean Code**: Used modular functions and descriptive variables with comments.
- **Data Validation**: Ensured data alignment with the schema before migration.
- **Version Control**: Committed frequently with detailed messages.
- **Thorough Testing**: Addressed edge cases and validated endpoints with Postman.

Conclusion

Through meticulous planning and execution, I ensured the Sanity CMS was accurately populated and fully integrated into a functional Next.js application. Robust practices and thorough testing were key to achieving a scalable and efficient.