

Hackathon Day 3: API Integration and Data Migration for Ecommerce platform

INTRODUCTION

This document outlines the process of **API integration**, **schema adjustments**, and **data migration** into **Sanity CMS** for the **Comforty Marketplace**. The goal was to seamlessly integrate external product data, enhance the CMS structure, and ensure optimal marketplace functionality.

API INTEGRATION PROCESS

1. Identifying API Endpoints

We began by analyzing the external API providing product data. The key endpoints included:

- **Product Details:** Information like title, price, description, inventory, etc.
- **Product Images:** Endpoint for retrieving product image data.
- **Category Data:** Details about product categories.

2. Creating API Fetch Functions

To retrieve data, we implemented fetch functions using the `fetch()` method in JavaScript. This allowed dynamic retrieval of product information from the external API.

Example Code Snippet:

```
useEffect(() => {  
  fetchProducts()  
    .then((data) => setProducts(data))  
    .catch((error) => console.error('Error fetching products:', error));  
}, []);
```

3. Integrating API Calls in the Project

API calls were integrated into the **Next.js** lifecycle methods, particularly using React hooks like `useEffect`. This enabled dynamic display of product data on the frontend.

Example Usage on the Product Detail Page:

- Displayed product details such as **title**, **price**, **description**, and **inventory status**.

- Integrated a fallback mechanism to handle errors during data fetch.

Error Handling:

To ensure a smooth user experience, error handling was implemented during API calls.

Example:

```
if (loading) return <div>Loading...</div>;  
if (error) return <div>Error: Unable to fetch products</div>;
```

SCHEMA ADJUSTMENTS

To ensure compatibility with external API data, several schema changes were made within **Sanity CMS**.

1. Product Schema Adjustments

New fields were added to the **Product schema**:

- **priceWithoutDiscount**: Numeric field to store the original price of a product before discounts.
- **tags**: Array field to store product labels such as "featured" or "on sale".
- **rating**: Number field to store the product's rating.

2. Category Schema Enhancements

A reference field was added to the product schema to link products to their respective categories, ensuring proper categorization and display.

3. Image Handling

Product images were fetched from the external API and uploaded as assets in **Sanity CMS**. An **asset reference field** was used to link images with corresponding products in the CMS.

DATA MIGRATION

The migration process involved transferring product data from the external API into **Sanity CMS** for seamless management.

1. Preparing the Migration Script

A Node.js script was written using **Sanity's client library** to automate product data migration into the CMS.

2. Script to Migrate Product Data

The script iterated over the product data fetched from the external API and dynamically created documents within **Sanity CMS**.

Sample Migration Script:

```
const sanityClient = require('@sanity/client');
const client = sanityClient({
  projectId: 'your-project-id',
  dataset: 'your-dataset',
  useCdn: false
});

const products = fetchProductData(); // Hypothetical function to fetch data

products.forEach(product => {
  client.create({
    _type: 'product',
    title: product.title,
    price: product.price,
    description: product.description,
    category: product.category,
    rating: product.rating,
    images: product.images
  });
});
```

Conclusion

The **API integration** and **data migration** for the **Comforty Marketplace** were successfully completed. The **schema adjustments** ensured the CMS is fully compatible with external product data, and the **migration script** effectively populated **Sanity CMS** with product details. The marketplace is now fully capable of dynamically displaying and managing products, offering a seamless and efficient user experience.

-Rumaisa Ayaz