Day 3 - API Integration and Data Migration

Introduction

On Day 3, the focus was on integrating APIs and migrating data into Sanity CMS. The purpose was to develop a functional backend for an e-commerce marketplace. This hands-on task replicated real-world scenarios, allowing me to gain practical experience in handling API integrations and managing data migration from other platforms.
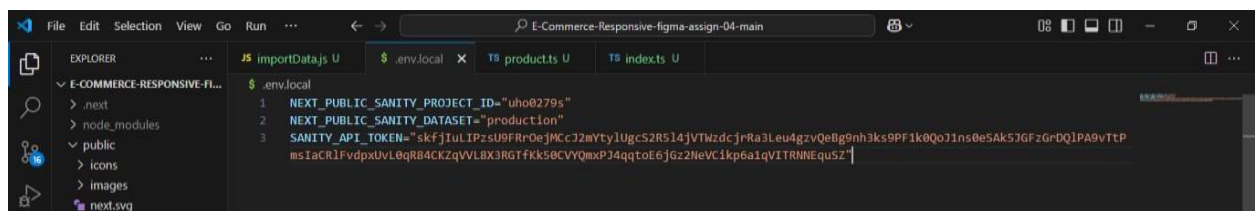
API Overview (Template 1)

- **Migration Script:** https://github.com/developerhammadrehman/template1/blob/main/import Data.js
- **Schema:** https://github.com/developer-hammadrehman/template1/blob/main/src/sanity/schemaTypes/products.t s
- **GitHub Repository:** https://github.com/developer-hammadrehman/template1/tree/main

Step-by-Step Integration Process for Day-3

1. Understand the Provided API:

The first step was to thoroughly review the API documentation provided for Template 1 This involved identifying the key endpoints:
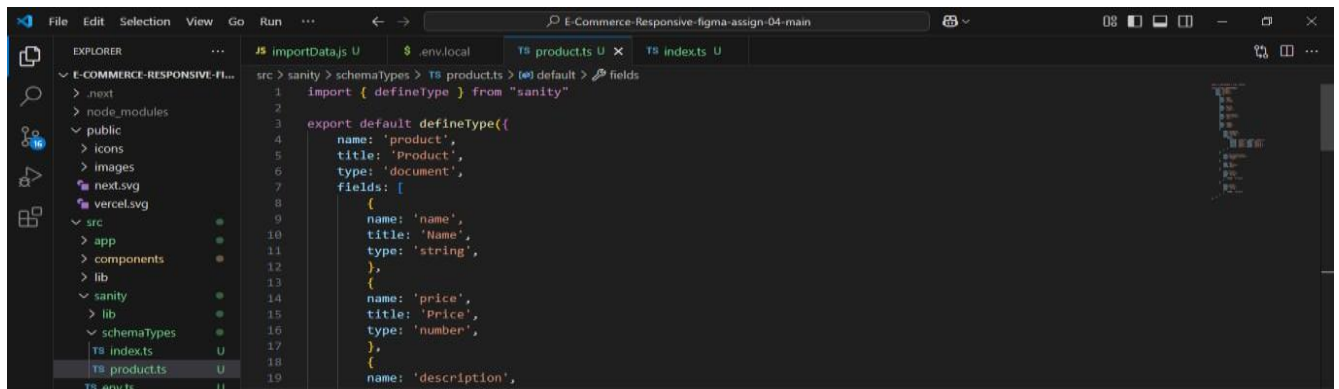
- **Product Listings**: /products
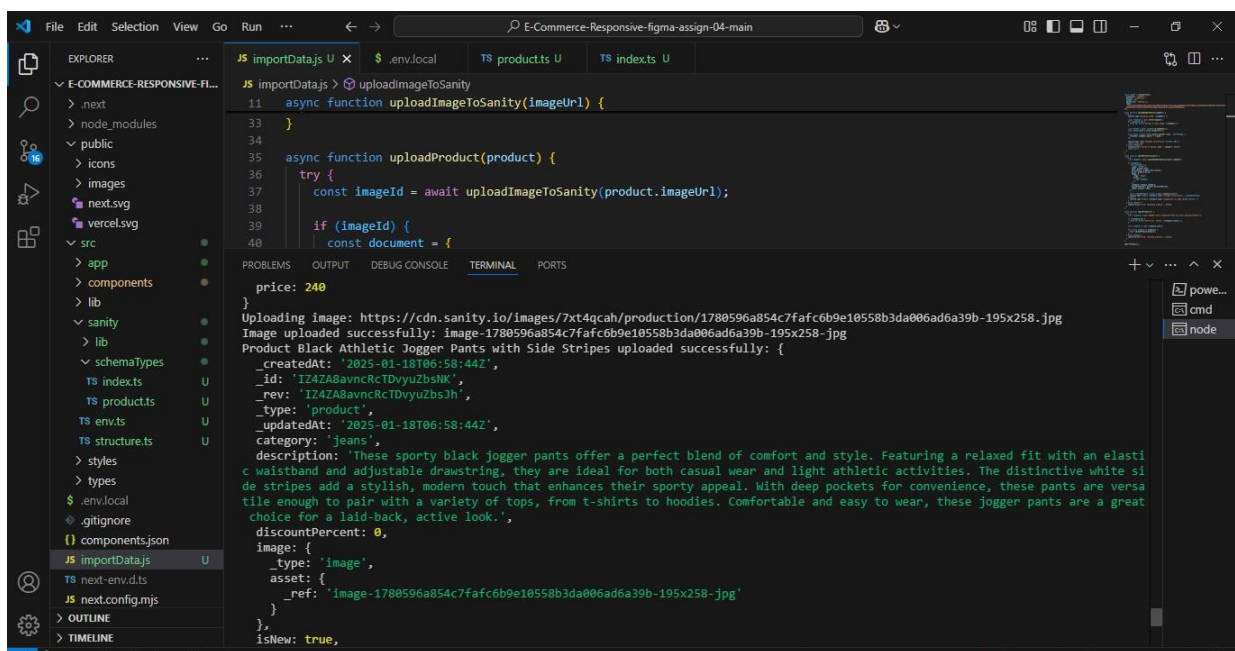- **Categories**: /categories

2. Validate and Adjust Your Schema:

To ensure compatibility, I compared the existing Sanity CMS schema with the API data structure. Adjustments were made to align the fields properly, ensuring smooth data integration. A custom schema file named Product.ts

Product.ts Snippet:


```ts
import { defineType } from "sanity"

export default defineType({
    name: 'product',
    title: 'Product',
    type: 'document',
    fields: [
        {
            name: 'name',
            title: 'Name',
            type: 'string',
        },
        {
            name: 'price',
            title: 'Price',
            type: 'number',
        },
        {
            name: 'description',
```

3. Data Migration Steps and Tools

Tools Used:

- **Sanity Client:** A tool to interact with the CMS.

- **Node.js & Fetch API:** Used for fetching and uploading data.

- **Environment Variables:** Used to store sensitive credentials like project ID and token.

Steps:

1. **Environment setup:** Installed dependencies and configured environment variables.
2. **Sanity Client Setup:** Generated a project id and token.
3. **Addition of ID and Token:** Added ID and Token in env.local .
4. **ImportData.js:** Created a file named ImportData.js to import the data from the /api/products (provided API) endpoint into my Sanity project. Also provided the ID and Token in the code.
5. **Run the Command:** then in cmd I run this command which fetched the data. (node importData.js)
6. **Verify Data:** After migration, I verified data in sanity to ensure the migration is successful.

API Response

Sanity Response

Groq Query

NextJs (Frontend)