"Day 3 - API Integration Report - [E-Commerce]"

Prepared by: Kanwal Rafiqe

---

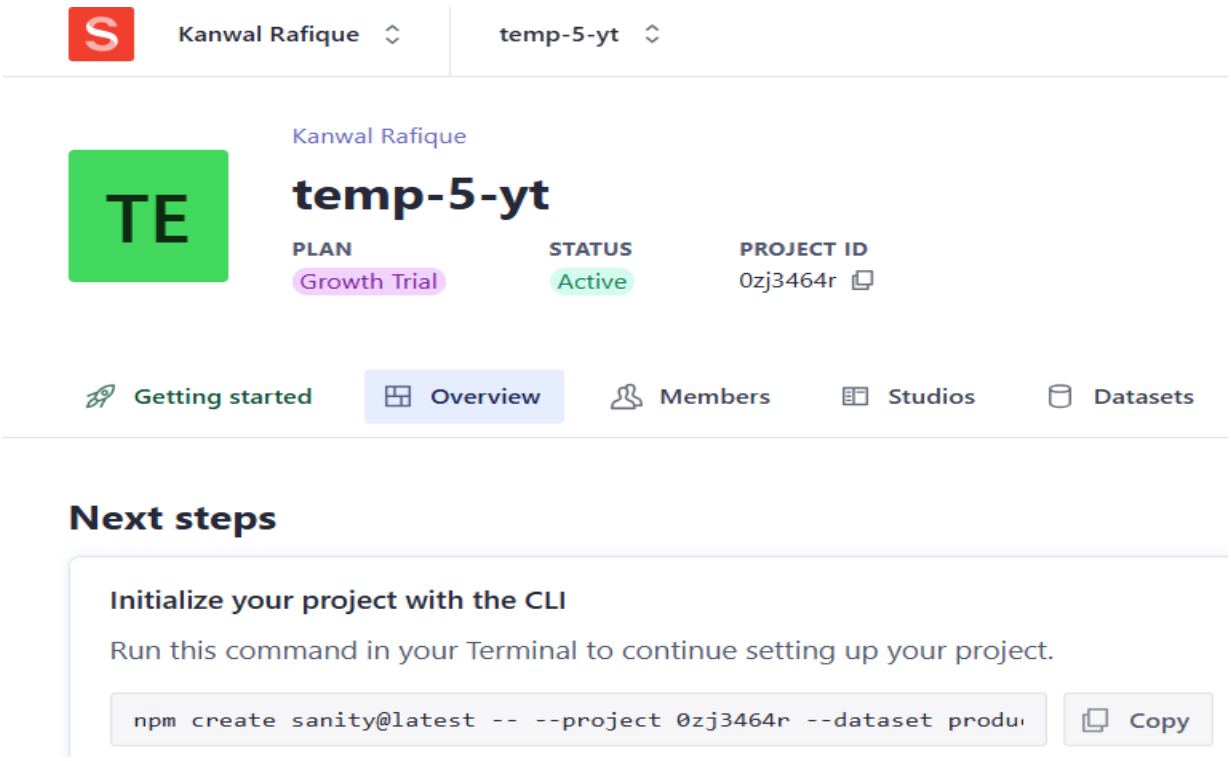
1. Introduction

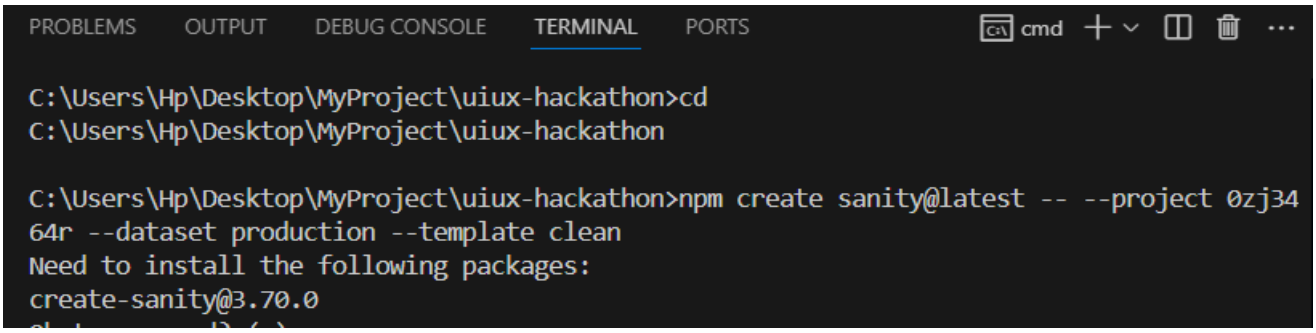
On Day 3 of the hackathon, the objective was to integrate APIs and migrate data into the Sanity CMS. The focus was on utilizing external APIs, such as the one provided for product data, and integrating them with the Sanity CMS backend to create a functional marketplace. The process mimics real-world scenarios where APIs are used to pull data and display it within the frontend, as well as ensuring compatibility with existing templates.

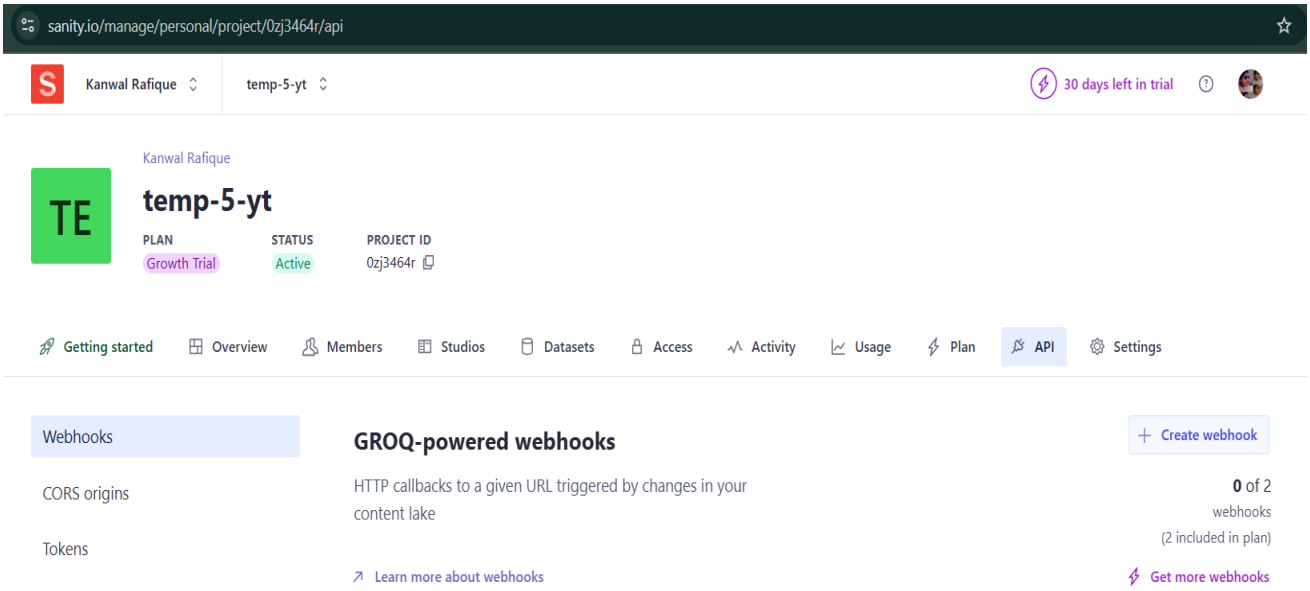
---

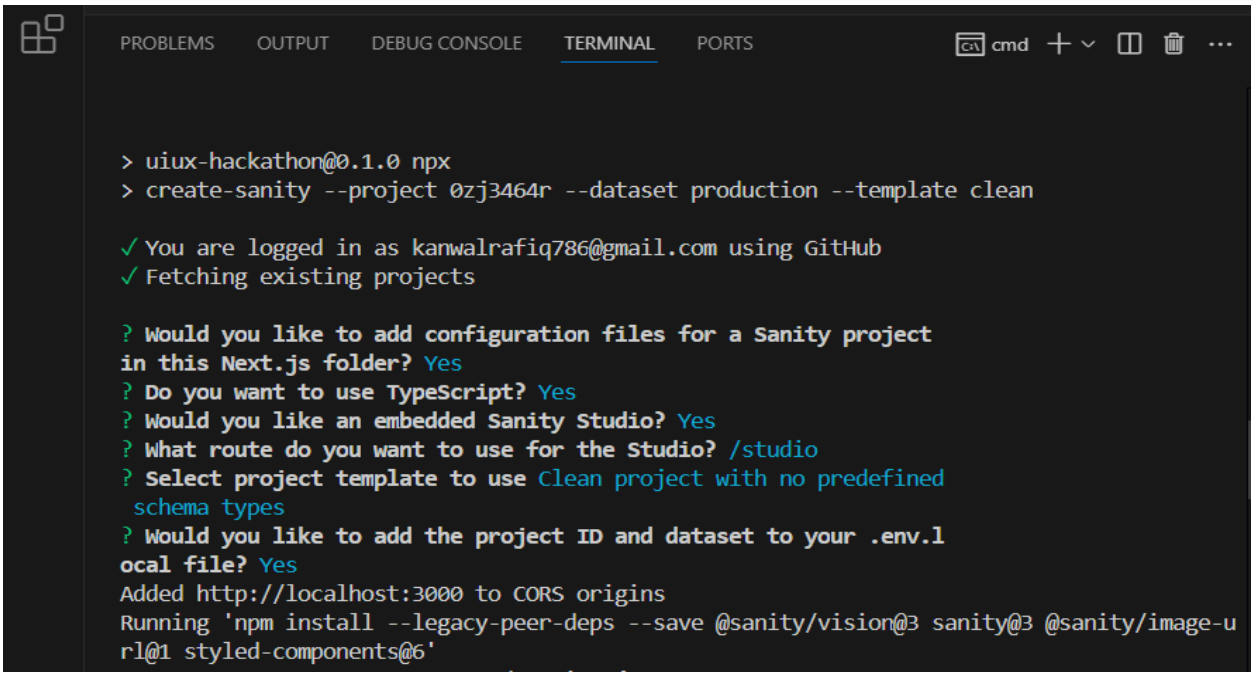
2. Creating Project in Sanity, Generating Token

1. **Clone Repository:** The repository was cloned to get started with the project.
2. **Insert Sanity CMS:** Sanity CMS was integrated into the existing repo to handle the backend management.
3. **API Integration:** The product data API was called to fetch products from the external source.
4. **Data Import:** Data was imported into the Sanity CMS using the provided migration script.
5. **Data Verification:** After migration, the data was verified in Sanity CMS to ensure accuracy and completeness.




PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
cmd
C:\Users\Hp\Desktop\MyProject\uiux-hackathon>cd
C:\Users\Hp\Desktop\MyProject\uiux-hackathon
C:\Users\Hp\Desktop\MyProject\uiux-hackathon>npm create sanity@latest -- --project 0zj34
64r --dataset production --template clean
Need to install the following packages:
create-sanity@3.70.0

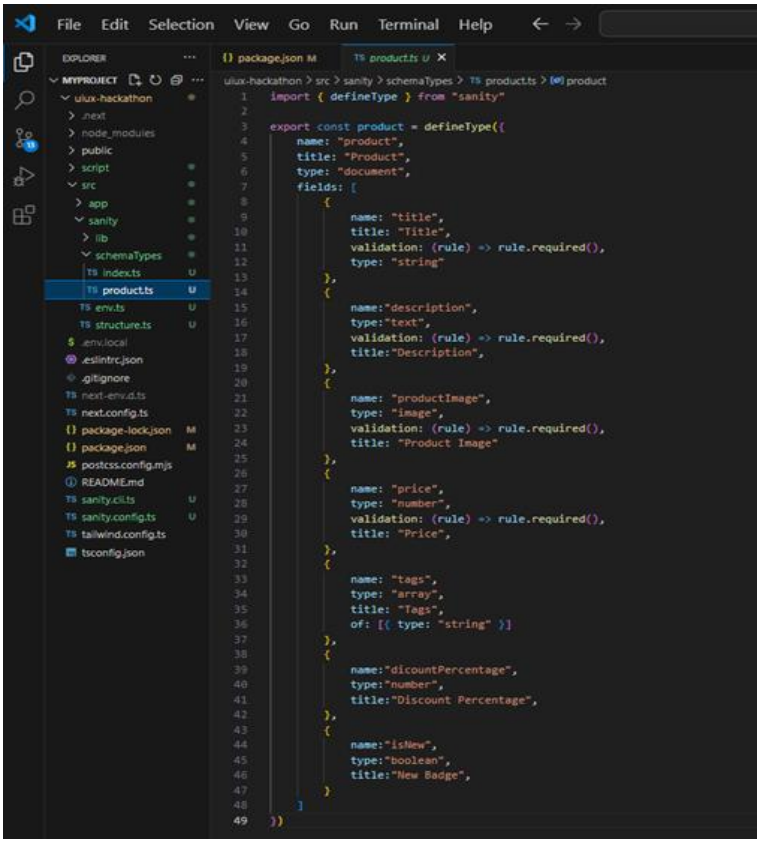

sanity.io/manage/personal/project/0zj3464r/api
Kanwal Rafique
temp-5-yt
30 days left in trial
Kanwal Rafique
TE
temp-5-yt
PLAN
Growth Trial
STATUS
Active
PROJECT ID
0zj3464r
Getting started
Overview
Members
Studios
Datasets
Access
Activity
Usage
Plan
API
Settings
Webhooks
CORS origins
Tokens
GROQ-powered webhooks
HTTP callbacks to a given URL triggered by changes in your content lake
Learn more about webhooks
Create webhook
0 of 2
webhooks
(2 included in plan)
Get more webhooks

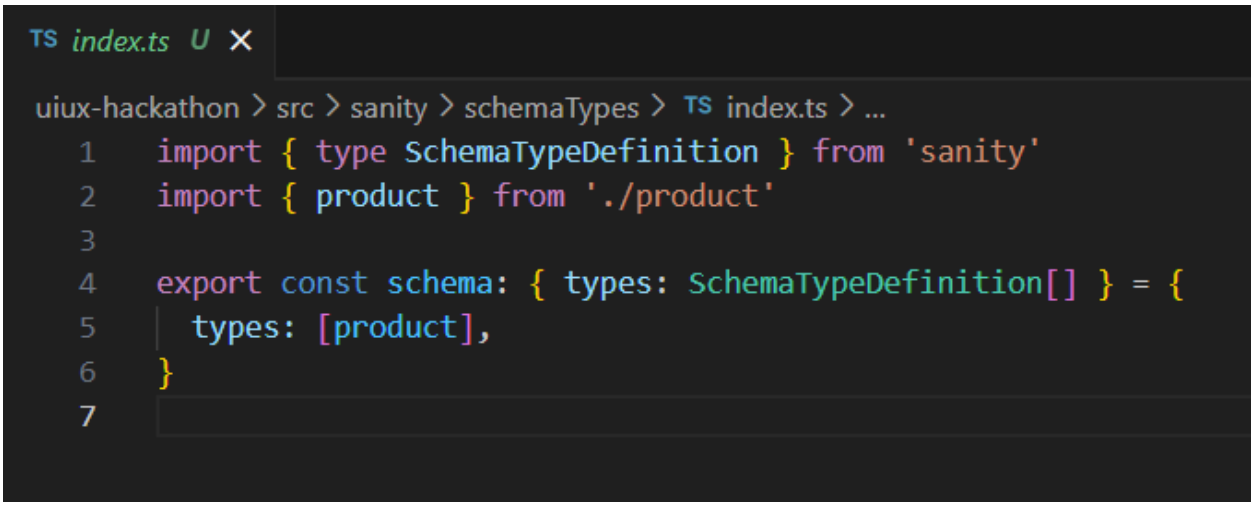

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
cmd
> uiux-hackathon@0.1.0 npx
> create-sanity --project 0zj3464r --dataset production --template clean
✓ You are logged in as kanwalrafiq786@gmail.com using GitHub
✓ Fetching existing projects
? Would you like to add configuration files for a Sanity project
in this Next.js folder? Yes
? Do you want to use TypeScript? Yes
? Would you like an embedded Sanity Studio? Yes
? What route do you want to use for the Studio? /studio
? Select project template to use Clean project with no predefined
schema types
? Would you like to add the project ID and dataset to your .env.l
ocal file? Yes
Added http://localhost:3000 to CORS origins
Running 'npm install --legacy-peer-deps --save @sanity/vision@3 sanity@3 @sanity/image-u
rl@1 styled-components@6'

3. API Integration Process

The API integrated during Day 3 is hosted at the following endpoint:

- **API URL:** https://template6-six.vercel.app/api/products

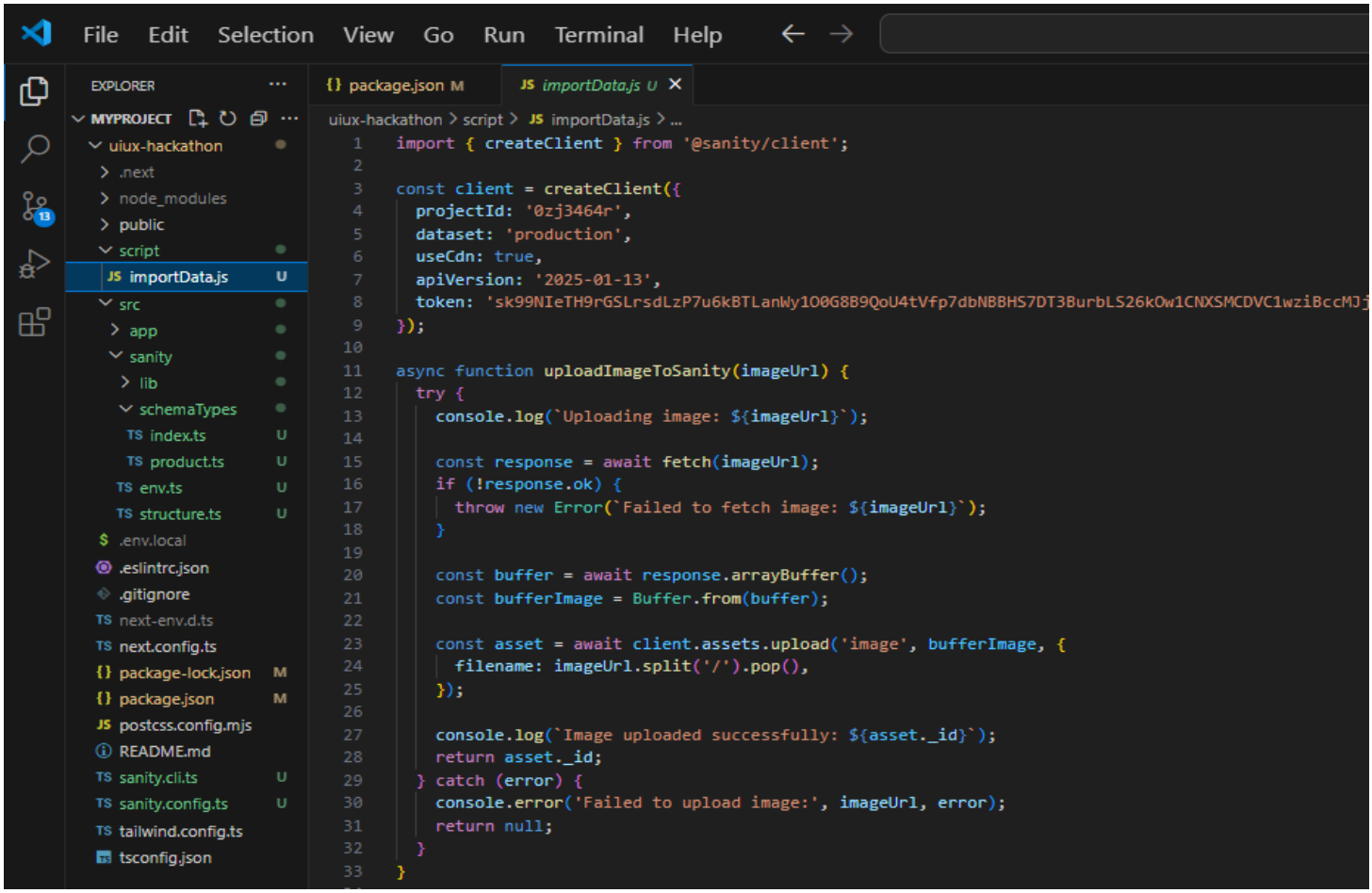


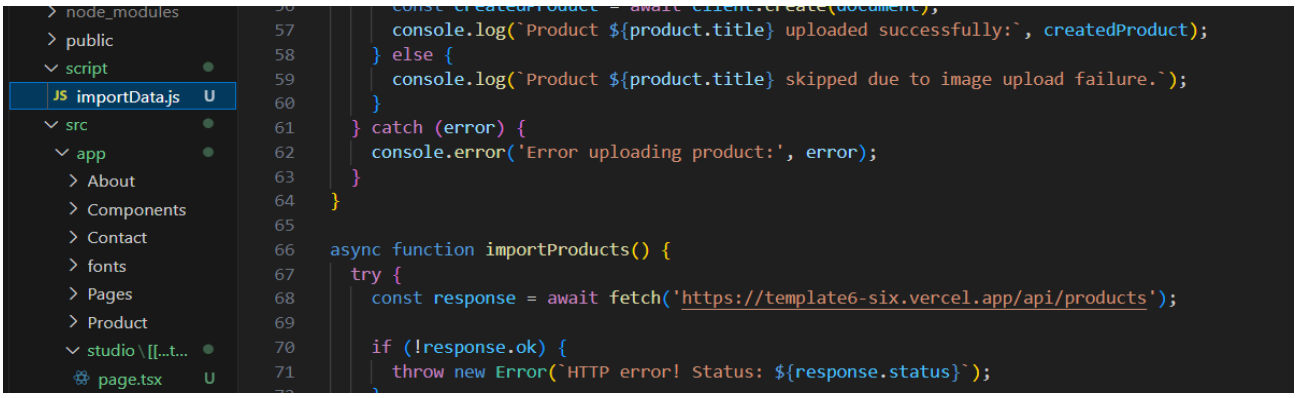


4. Data Migration

As part of the migration process, the product data fetched from the API is added into the Sanity CMS backend to ensure smooth data management. The migration script used in this process is from the repository provided.

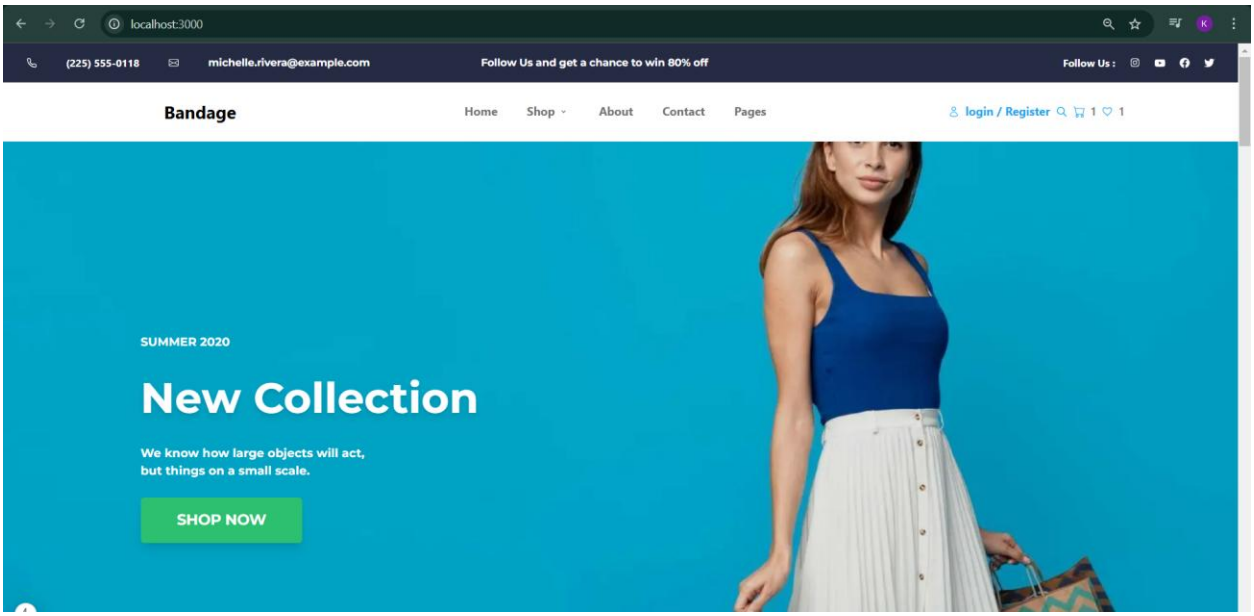
- **Migration Script:** importData.js
- **API Route for Products:** route.ts

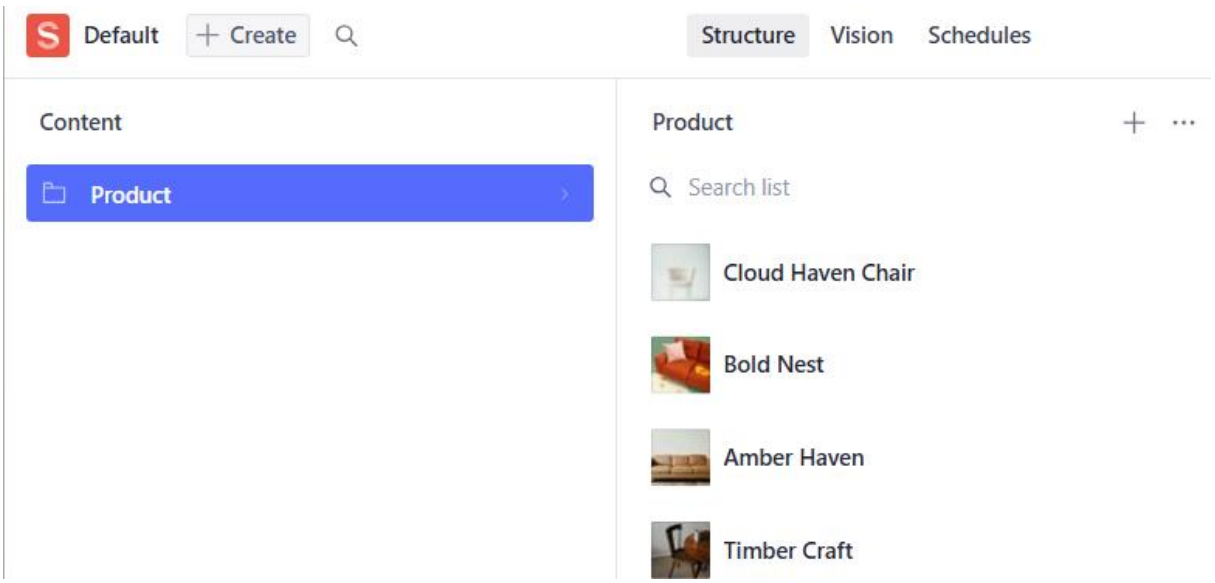


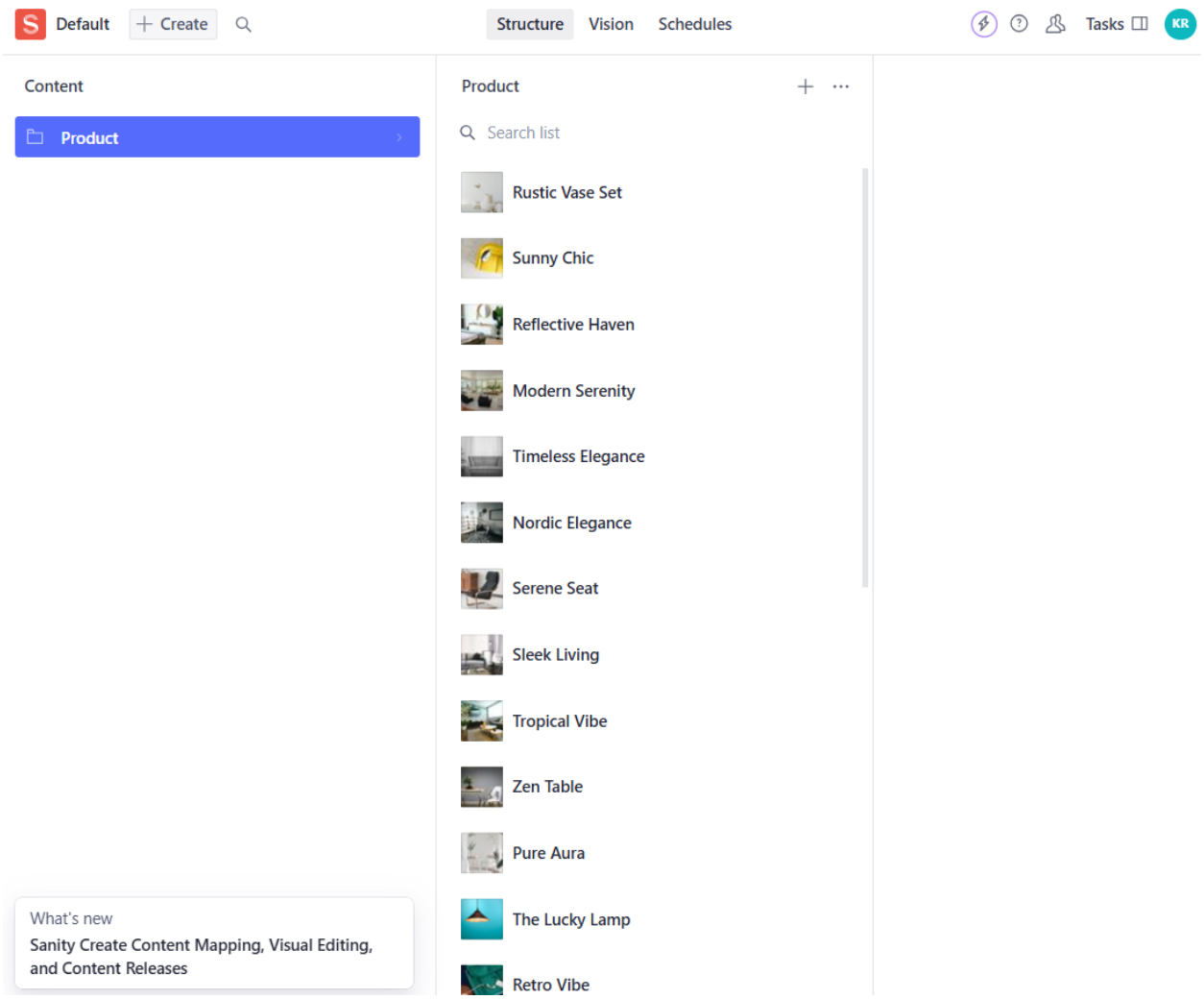


5. Conclusion

The integration of the API and the data migration into Sanity CMS was successfully completed. The product data is now available in the Sanity CMS and displayed dynamically in the frontend. This approach mimics real-world integration and data management practices, helping students understand how to work with headless APIs and migrate data to CMS platforms for E-Commerce websites.







---

6. References

1. API Endpoint: https://template6-six.vercel.app/api/products
2. Migration Script: importData.js
3. API Route File: route.ts

---

7. Code Snippets

1. **API Fetch Code:**

```
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);
  }
}
```

2. **Sanity CMS Migration Script:**

```
uiux-hackathon > {} package.json > {} scripts
{
  "name": "uiux-hackathon",
  "version": "0.1.0",
  "private": true,
  "type": "module",
  ▷ Debug
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "import-data": "node script/importData.js"
```