

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



Prepared by: Kanwal Rafique


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.


 Kanwal Rafique 


temp-5-yt 





Kanwal Rafique


temp-5-yt


PLAN	STATUS	PROJECT ID
Growth Trial	Active	0zj3464r 

 Getting started

 Overview

 Members

 Studios


 Datasets

Next steps

Initialize your project with the CLI

Run this command in your Terminal to continue setting up your project.

```
npm create sanity@latest -- --project 0zj3464r --dataset produ
```

 Copy

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
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 0zj3464r --dataset production --template clean
Need to install the following packages:
create-sanity@3.70.0
```

sanity.io/manage/personal/project/0zj3464r/api

S Kanwal Rafique

temp-5-yt

30 days left in trial

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
> uiux-hackathon@0.1.0 npx
> create-sanity --project 0zj3464r --dataset production --template clean

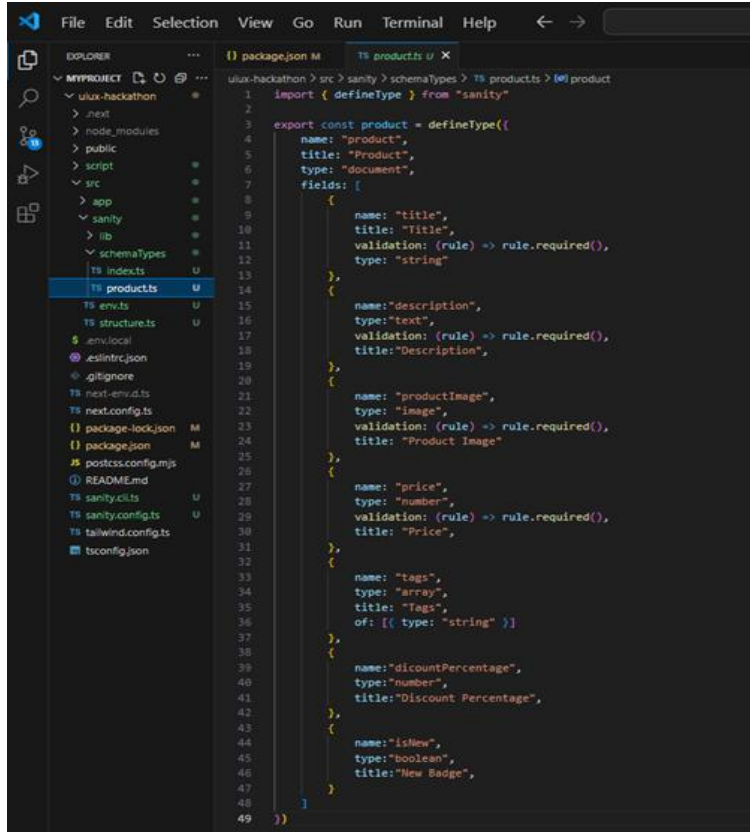
✓ You are logged in as kanwalrafiq786@gmail.com using GitHub
✓ Fetching existing projects

? 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.local file? Yes
Added http://localhost:3000 to CORS origins
Running 'npm install --legacy-peer-deps --save @sanity/vision@3 sanity@3 @sanity/image-url@1 styled-components@6'
```

3. API Integration Process

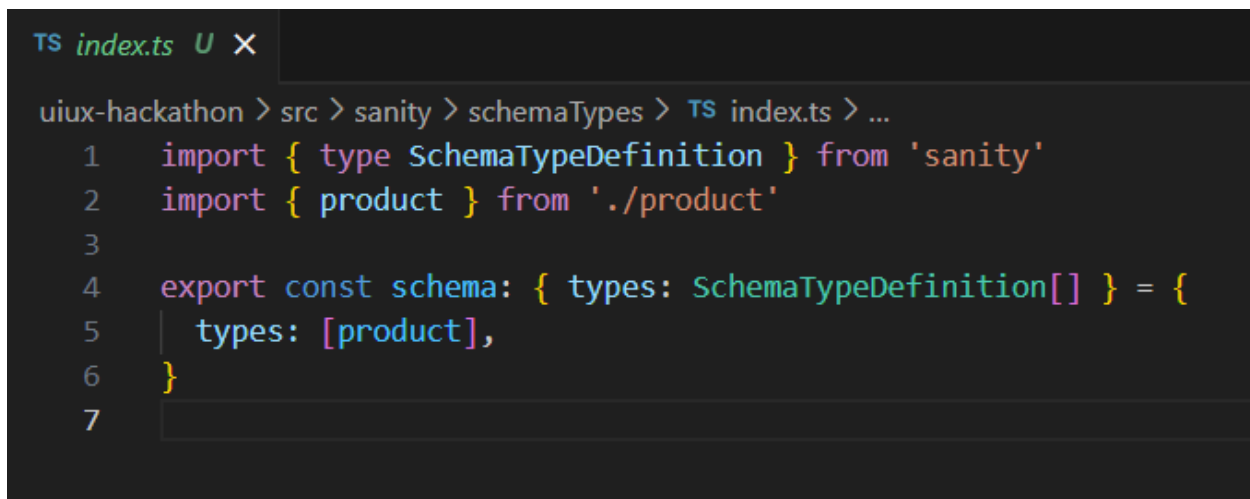
The API integrated during Day 3 is hosted at the following endpoint:

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



The screenshot shows a VS Code editor with the Explorer sidebar on the left. The Explorer shows a project structure with folders like 'src', 'sanity', and 'schemaTypes'. The file 'TS products.ts' is selected. The main editor displays the content of 'TS products.ts', which defines a Sanity schema for a product. The code includes imports for 'defineType' from 'sanity' and 'product' from './product'. It then defines a 'product' schema with fields: 'title' (string, required), 'description' (text, required), 'productImage' (image, required), 'price' (number, required), 'tags' (array of strings), 'discountPercentage' (number), and 'isNew' (boolean).

```
1 import { defineType } from "sanity"
2
3 export const product = defineType({
4   name: "product",
5   title: "Product",
6   type: "document",
7   fields: [
8     {
9       name: "title",
10      title: "Title",
11      validation: (rule) => rule.required(),
12      type: "string"
13    },
14    {
15      name: "description",
16      type: "text",
17      validation: (rule) => rule.required(),
18      title: "Description",
19    },
20    {
21      name: "productImage",
22      type: "image",
23      validation: (rule) => rule.required(),
24      title: "Product Image"
25    },
26    {
27      name: "price",
28      type: "number",
29      validation: (rule) => rule.required(),
30      title: "Price",
31    },
32    {
33      name: "tags",
34      type: "array",
35      title: "Tags",
36      of: [{ type: "string" }]
37    },
38    {
39      name: "discountPercentage",
40      type: "number",
41      title: "Discount Percentage",
42    },
43    {
44      name: "isNew",
45      type: "boolean",
46      title: "New Badge",
47    }
48  ]
49 })
```



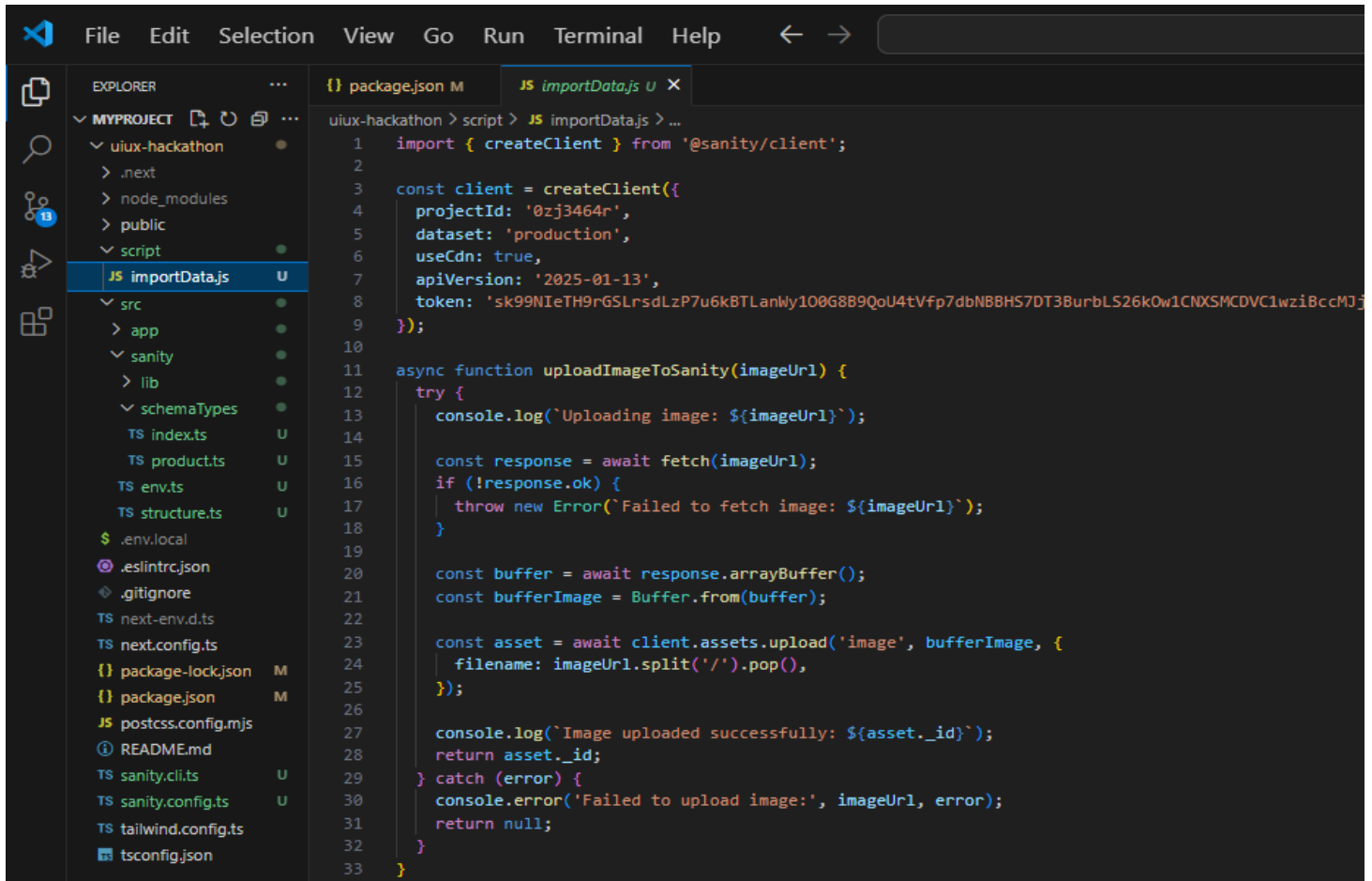
The screenshot shows a VS Code editor with the Explorer sidebar on the left. The file 'TS index.ts' is selected. The main editor displays the content of 'TS index.ts', which imports 'SchemaTypeDefinition' from 'sanity' and 'product' from './product'. It then exports a 'schema' object with a 'types' array containing the 'product' schema.

```
1 import { type SchemaTypeDefinition } from 'sanity'
2 import { product } from './product'
3
4 export const schema: { types: SchemaTypeDefinition[] } = {
5   types: [product],
6 }
7
```

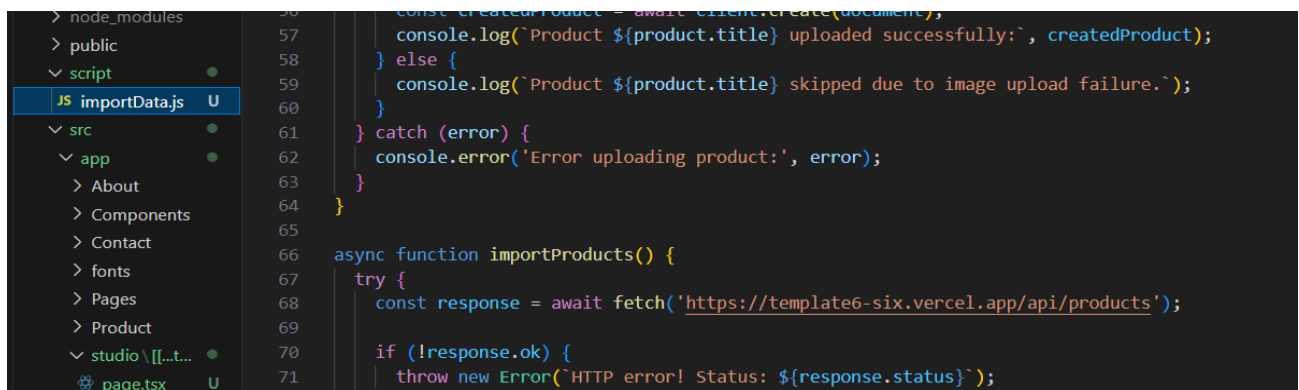
4. Data Migration

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](#)



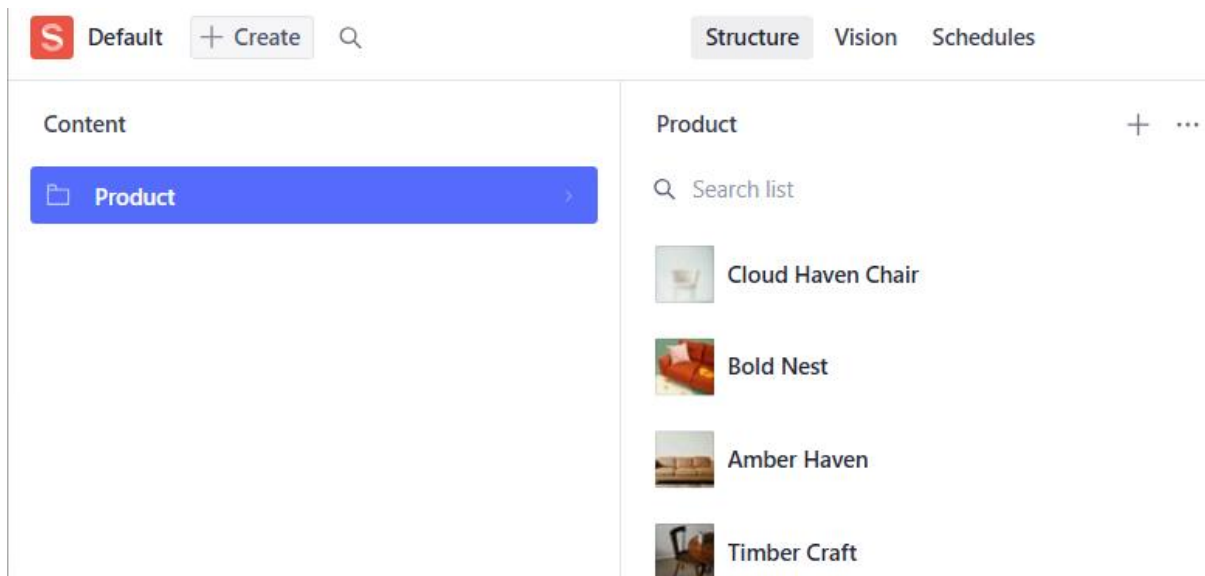
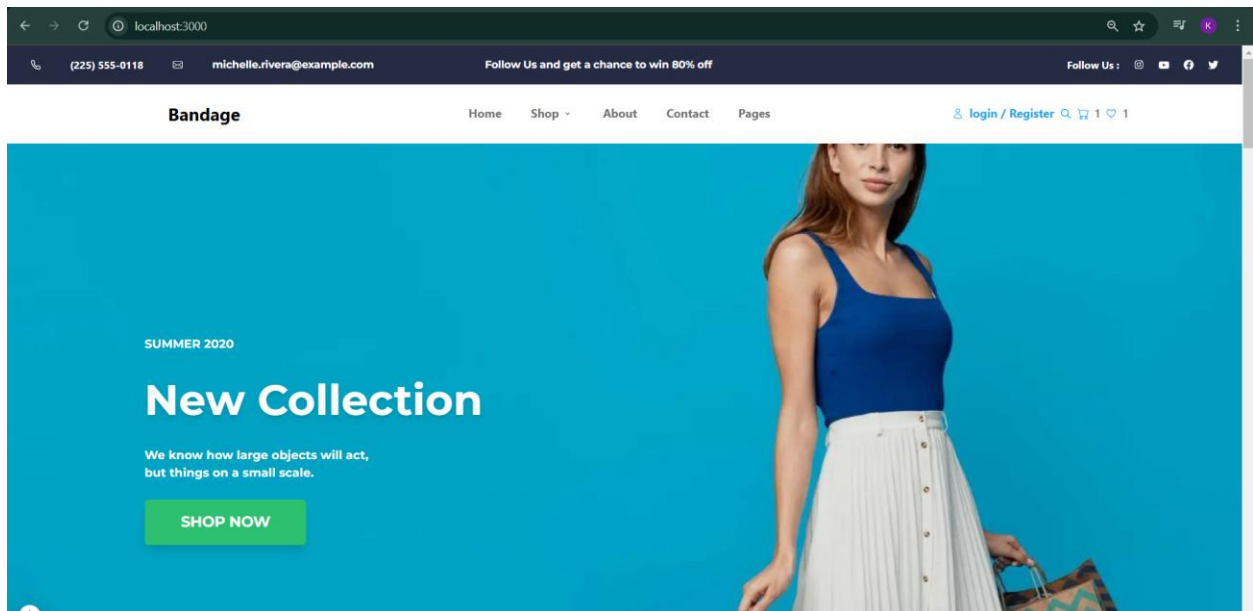
```
1 import { createClient } from '@sanity/client';
2
3 const client = createClient({
4   projectId: '0zj3464r',
5   dataset: 'production',
6   useCdn: true,
7   apiVersion: '2025-01-13',
8   token: 'sk99NIeTH9rGSLrSdLzP7u6kBTLanMy100G8B9QoU4tVfp7dbNB8H57DT3BurbLS26kOw1CNXSMCDVC1wziBccMJj'
9 });
10
11 async function uploadImageToSanity(imageUrl) {
12   try {
13     console.log(`Uploading image: ${imageUrl}`);
14
15     const response = await fetch(imageUrl);
16     if (!response.ok) {
17       throw new Error(`Failed to fetch image: ${imageUrl}`);
18     }
19
20     const buffer = await response.arrayBuffer();
21     const bufferImage = Buffer.from(buffer);
22
23     const asset = await client.assets.upload('image', bufferImage, {
24       filename: imageUrl.split('/').pop(),
25     });
26
27     console.log(`Image uploaded successfully: ${asset._id}`);
28     return asset._id;
29   } catch (error) {
30     console.error('Failed to upload image:', imageUrl, error);
31     return null;
32   }
33 }
```

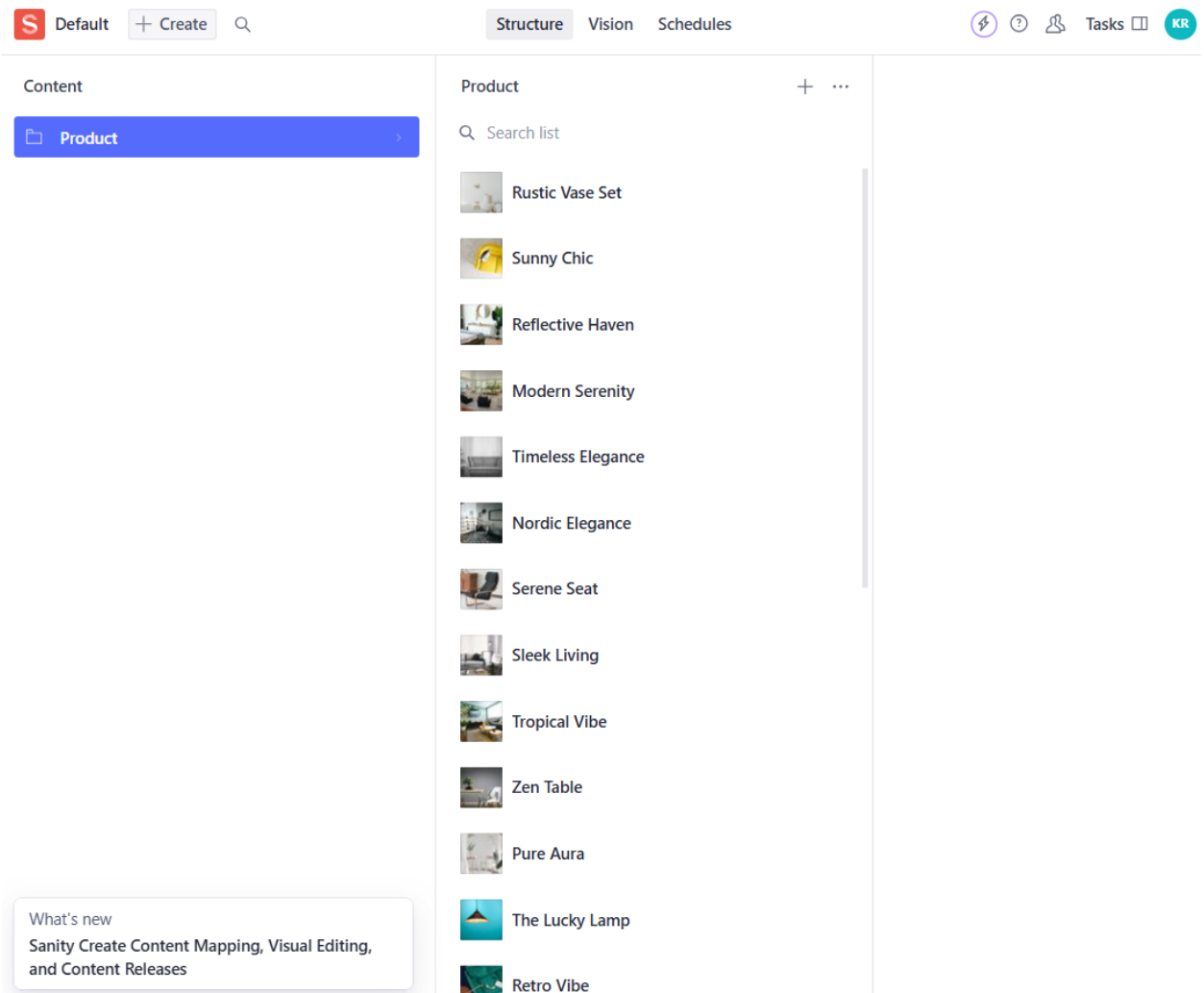


```
57 const createdProduct = await client.create(document);
58 console.log(`Product ${product.title} uploaded successfully:`, createdProduct);
59 } else {
60   console.log(`Product ${product.title} skipped due to image upload failure.`);
61 }
62 } catch (error) {
63   console.error('Error uploading product:', error);
64 }
65
66 async function importProducts() {
67   try {
68     const response = await fetch('https://template6-six.vercel.app/api/products');
69
70     if (!response.ok) {
71       throw new Error(`HTTP error! Status: ${response.status}`);
72     }
73   }
74 }
```

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
1   {
2     "name": "uiux-hackathon",
3     "version": "0.1.0",
4     "private": true,
5     "type": "module",
6     "scripts": {
7       "dev": "next dev",
8       "build": "next build",
9       "start": "next start",
10      "lint": "next lint",
11      "import-data": "node script/importData.js"
12    }
```