

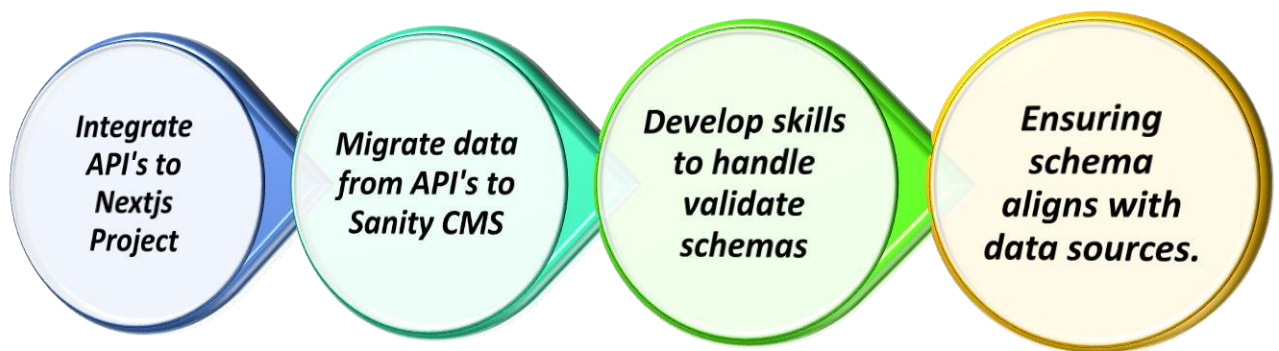
HACKATHON 3

DAY 3 - API INTEGRATION AND DATA MIGRATION

OBJECTIVE:

The focus for Day 3 is integrating APIs and migrating data into Sanity CMS to build a functional marketplace backend. This approach replicates real-world practices and prepares to handle diverse client requirements, including integrating headless APIs or migrating existing data from popular E-Commerce platforms.

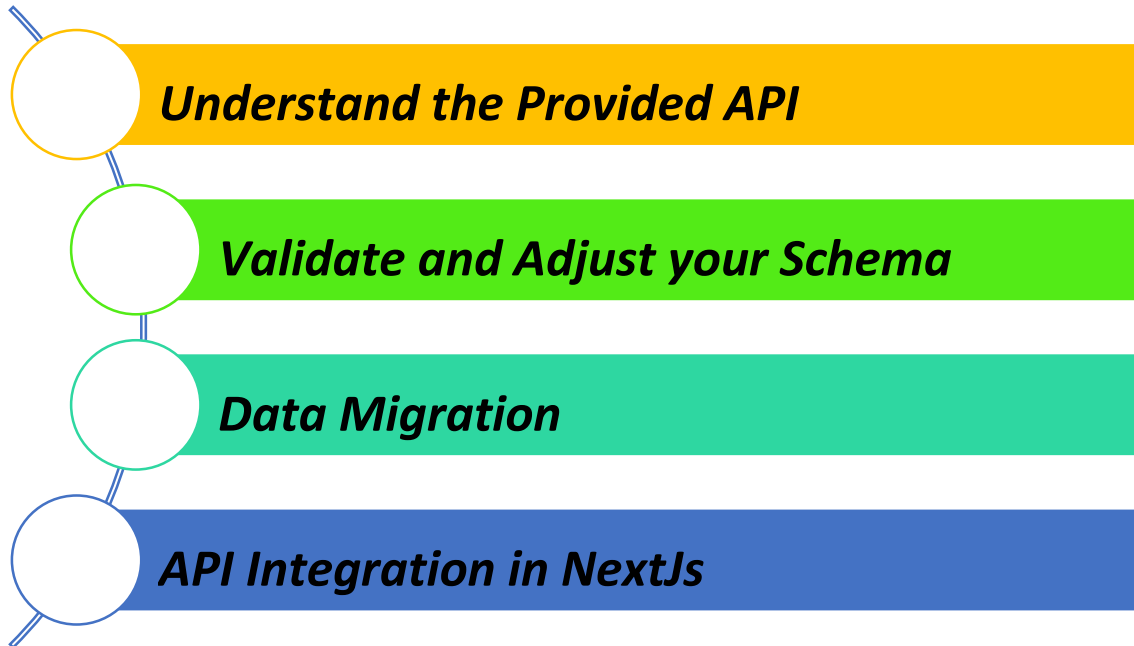
Key Learning Outcomes:



API Overview:

API URL: <https://next-ecommerce-template-4.vercel.app/api/product>

STEPS FOR DAY 3:



CODE SNIPPETS

1. Understand the Provided API

```
src > scripts > JS import-data.mjs > ...
1 import { createClient } from '@sanity/client':

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ...

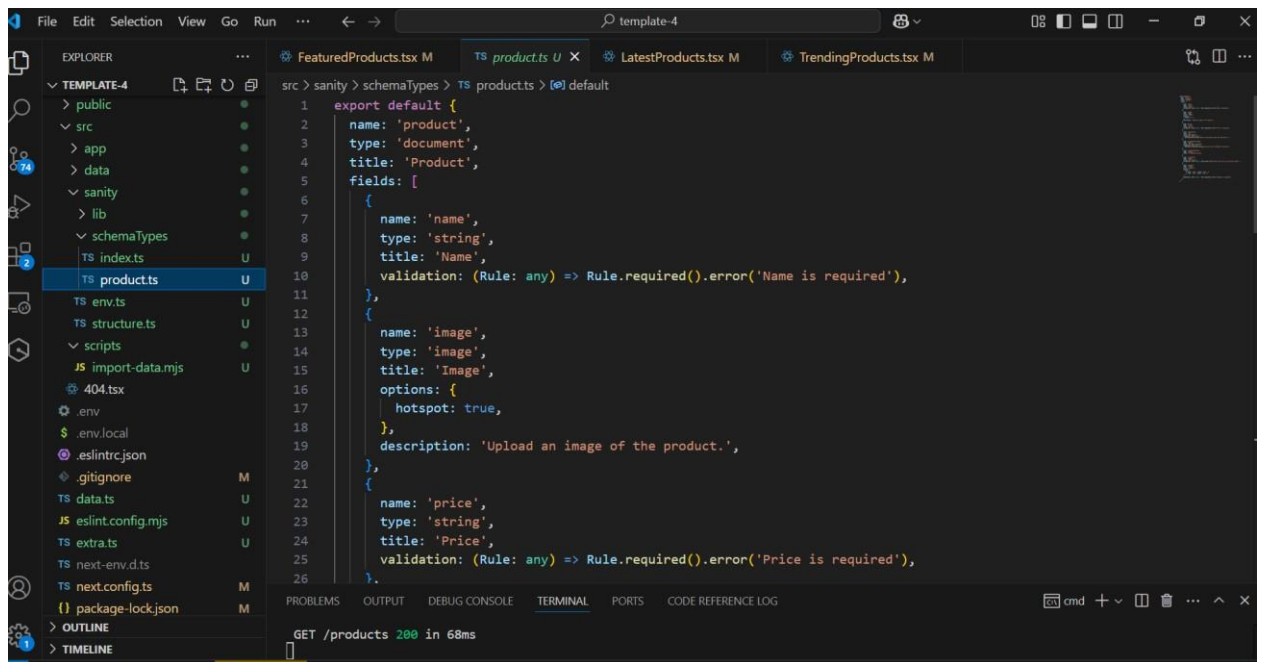
Uploading Image : https://next-e-commerce-template-4.vercel.app/product/Chair (22).png
Image Uploaded Successfully : image-c919c7d6e7b1d3ece287cde7c3fff40c97216ac3-374x374-png
Uploading Chair - Nordic Net Red Chair to Sanity !
Uploaded Successfully: AJVJVObFUPWewZSbi4F1EF
-----

Processing Item: Cantilever Chair
Uploading Image : https://next-e-commerce-template-4.vercel.app/product/Chair (23).png
Image Uploaded Successfully : image-90a8d692777c7c2b80c7e49916bdce829702c35d-342x342-png
Uploading Chair - Cantilever Chair to Sanity !
Uploaded Successfully: AJVJVObFUPWewZSbi4F1IW
-----

Processing Item: Luxury Flower Shell Sofa Chair
Uploading Image : https://next-e-commerce-template-4.vercel.app/product/Chair (34).png
Image Uploaded Successfully : image-b36c424e4368829cc69cefdb42ba7b9d213e965e-1258x1258-png
Uploading Sofa - Luxury Flower Shell Sofa Chair to Sanity !
Uploaded Successfully: jGHnK2T7WRTGSZ6CrmmWRqc
-----

Data Import Completed Successfully !
C:\Users\hp\Desktop\template-4>
```

2. Validate and Adjust Your Schema:

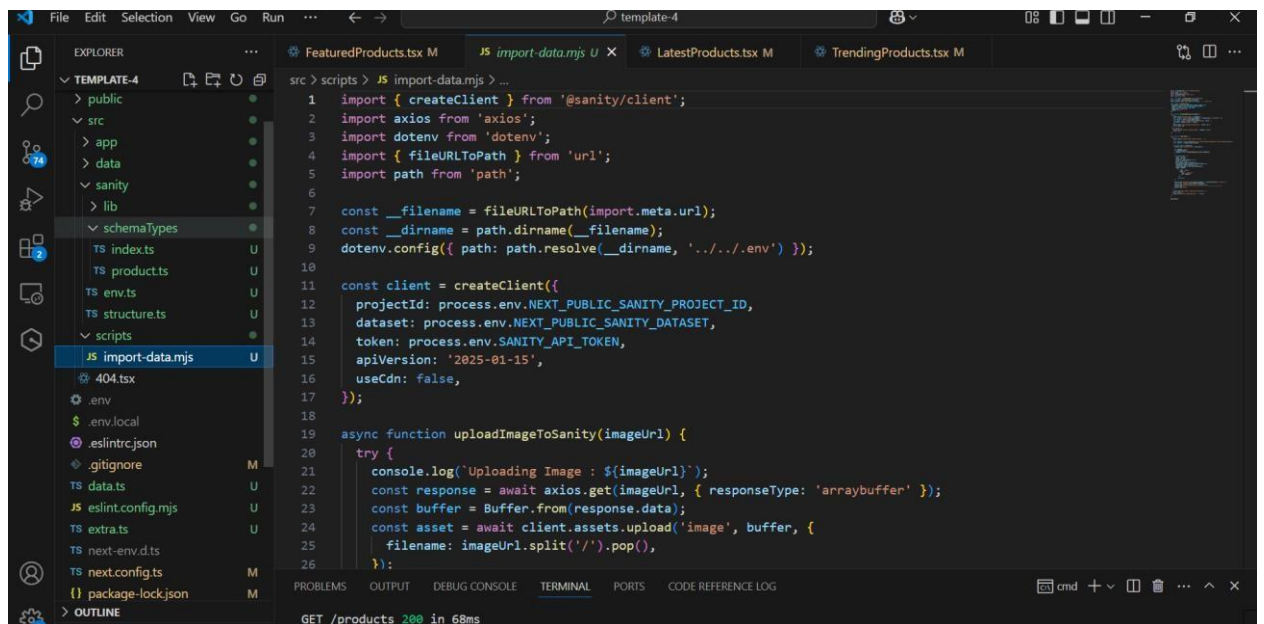


The screenshot shows the VS Code editor with the 'product.ts' file open. The file defines a schema type for 'product' with fields 'name', 'image', and 'price'. The 'name' field is a string with a required validation rule. The 'image' field is an image type with a required validation rule and a description. The 'price' field is a string with a required validation rule. The terminal shows a GET request to '/products' returning 200 in 68ms.

```
src > sanity > schemaTypes > TS products > default
1 export default {
2   name: 'product',
3   type: 'document',
4   title: 'Product',
5   fields: [
6     {
7       name: 'name',
8       type: 'string',
9       title: 'Name',
10      validation: (Rule: any) => Rule.required().error('Name is required'),
11    },
12    {
13      name: 'image',
14      type: 'image',
15      title: 'Image',
16      options: {
17        hotspot: true,
18      },
19      description: 'Upload an image of the product.',
20    },
21    {
22      name: 'price',
23      type: 'string',
24      title: 'Price',
25      validation: (Rule: any) => Rule.required().error('Price is required'),
26    }
27  ]
28 }
```

GET /products 200 in 68ms

3. Data Migration

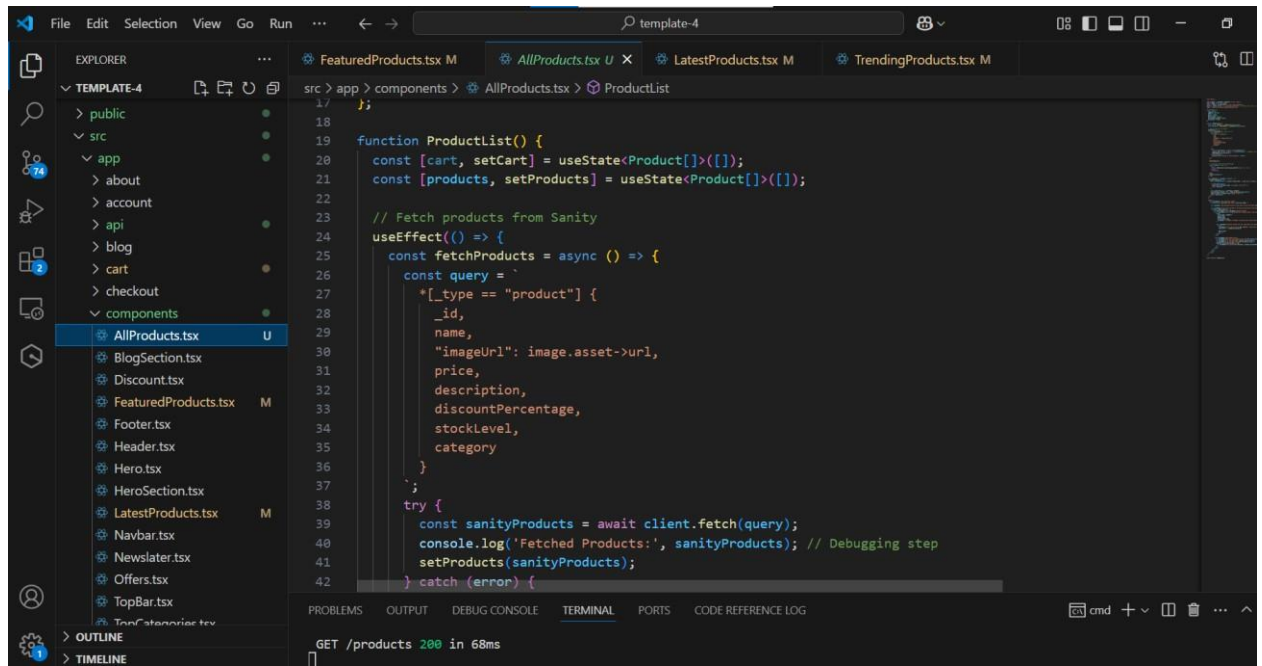


The screenshot shows the VS Code editor with the 'import-data.mjs' file open. The script imports necessary modules and defines a function to upload images to Sanity. The terminal shows a GET request to '/products' returning 200 in 68ms.

```
src > scripts > JS import-data.mjs > ...
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import dotenv from 'dotenv';
4 import { fileURLToPath } from 'url';
5 import path from 'path';
6
7 const __filename = fileURLToPath(import.meta.url);
8 const __dirname = path.dirname(__filename);
9 dotenv.config({ path: path.resolve(__dirname, '../.env') });
10
11 const client = createClient({
12   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
13   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
14   token: process.env.SANITY_API_TOKEN,
15   apiVersion: '2025-01-15',
16   useCdn: false,
17 });
18
19 async function uploadImageToSanity(imageUrl) {
20   try {
21     console.log('Uploading Image : ${imageUrl}');
22     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
23     const buffer = Buffer.from(response.data);
24     const asset = await client.assets.upload('image', buffer, {
25       filename: imageUrl.split('/').pop(),
26     });
27   } catch (error) {
28     console.error('Error uploading image:', error);
29   }
30 }
```

GET /products 200 in 68ms

4. API Integration in Next.js

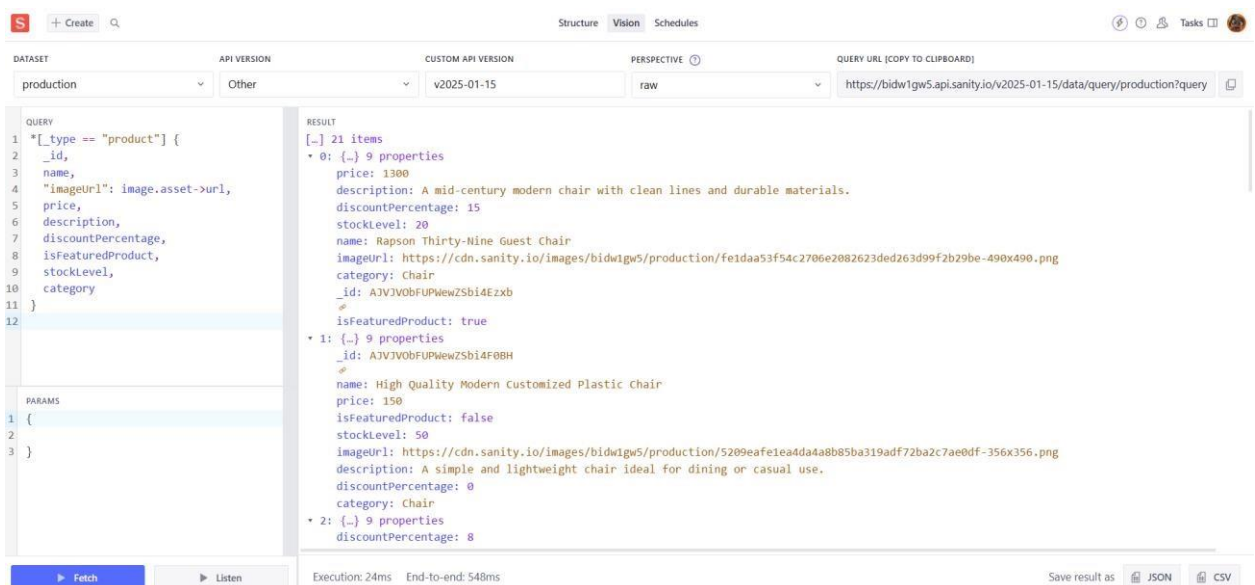


```
17 //
18
19 function ProductList() {
20   const [cart, setCart] = useState<Product[]>([]);
21   const [products, setProducts] = useState<Product[]>([]);
22
23   // Fetch products from Sanity
24   useEffect(() => {
25     const fetchProducts = async () => {
26       const query = `
27         *[_type == "product"] {
28           _id,
29           name,
30           "imageUrl": image.asset->url,
31           price,
32           description,
33           discountPercentage,
34           stockLevel,
35           category
36         }
37       `;
38       try {
39         const sanityProducts = await client.fetch(query);
40         console.log('Fetched Products:', sanityProducts); // Debugging step
41         setProducts(sanityProducts);
42       } catch (error) {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

GET /products 200 in 68ms

GROQ QUERY 1: ALL PRODUCTS



Sanity Studio interface showing a GROQ query for all products.

QUERY

```
1 *[_type == "product"] {
2   _id,
3   name,
4   "imageUrl": image.asset->url,
5   price,
6   description,
7   discountPercentage,
8   isFeaturedProduct,
9   stockLevel,
10  category
11 }
12
```

PARAMS

```
1 {
2 }
3
```

RESULT

```
[...] 21 items
• 0: {} 9 properties
  price: 1300
  description: A mid-century modern chair with clean lines and durable materials.
  discountPercentage: 15
  stockLevel: 20
  name: Rapson Thirty-Nine Guest Chair
  imageUrl: https://cdn.sanity.io/images/bidw1gw5/production/fe1daa53f54c2706e208263ded263d99f2b29be-490x490.png
  category: Chair
  _id: AJVJV0bFUPWewZSbi4Ezxb
  isFeaturedProduct: true
• 1: {} 9 properties
  _id: AJVJV0bFUPWewZSbi4F0BH
  name: High Quality Modern Customized Plastic chair
  price: 150
  isFeaturedProduct: false
  stockLevel: 50
  imageUrl: https://cdn.sanity.io/images/bidw1gw5/production/5209eafe1ea4da4a8b85ba319adff72ba2c7ae0df-356x356.png
  description: A simple and lightweight chair ideal for dining or casual use.
  discountPercentage: 0
  category: Chair
• 2: {} 9 properties
  discountPercentage: 8
```

Execution: 24ms End-to-end: 548ms

Save result as JSON CSV

2: BY CATEGORY

The screenshot shows the Sanity Studio interface with the following configuration:

- DATASET:** production
- API VERSION:** Other
- CUSTOM API VERSION:** v2025-01-15
- PERSPECTIVE:** raw
- QUERY URL [COPY TO CLIPBOARD]:** <https://bidw1gw5.api.sanity.io/v2025-01-15/data/query/production?query>

QUERY:

```
1 *[_type == "product" && category == "Chair"] {
2   _id,
3   name,
4   "imageUrl": image.asset->url,
5   price,
6   description,
7   discountPercentage,
8   stockLevel,
9   isFeaturedProduct
10 }
```

PARAMS:

```
1 {
2 }
3 }
```

RESULT:

```
[_] 18 items
  0: { _id: "A3VJVObFUPWewZSbi4Ezxb", name: "Rapson Thirty-Nine Guest Chair", imageUrl: "https://cdn.sanity.io/images/bidw1gw5/production/fe1daa53f54c2706e2082623ded263d99f2b29be-490x490.png", price: 1300, description: "A mid-century modern chair with clean lines and durable materials.", discountPercentage: 15, stockLevel: 20, isFeaturedProduct: true }
  1: { _id: "A3VJVObFUPWewZSbi4F0BH", name: "High Quality Modern Customized Plastic Chair", imageUrl: "https://cdn.sanity.io/images/bidw1gw5/production/5209eafe1ea4da4a8b85ba319adf72ba2c7ae0df-356x356.png", price: 150, description: "A simple and lightweight chair ideal for dining or casual use.", discountPercentage: 0, stockLevel: 50 }
  2: { _id: "A3VJVObFUPWewZSbi4F1EF", name: "Compact armchair with a cozy design for small spaces.", imageUrl: "https://cdn.sanity.io/images/bidw1gw5/production/cc09d3ce34a323e7d60432a366145277d5bb3c4f-606x528.png", price: 2000, description: "A modern chair with a carbon fiber frame and bold red accents.", discountPercentage: 0, stockLevel: 8 }
```

Execution: 23ms End-to-end: 903ms

3: FEATURED PRODUCTS

The screenshot shows the Sanity Studio interface with the following configuration:

- DATASET:** production
- API VERSION:** Other
- CUSTOM API VERSION:** v2025-01-15
- PERSPECTIVE:** raw
- QUERY URL [COPY TO CLIPBOARD]:** <https://bidw1gw5.api.sanity.io/v2025-01-15/data/query/production?query>

QUERY:

```
1 *[_type == "product" && isFeaturedProduct == true] {
2   _id,
3   name,
4   "imageUrl": image.asset->url,
5   price,
6   description,
7   discountPercentage,
8   stockLevel,
9   category
10 }
```

PARAMS:

```
1 {
2 }
3 }
```

RESULT:

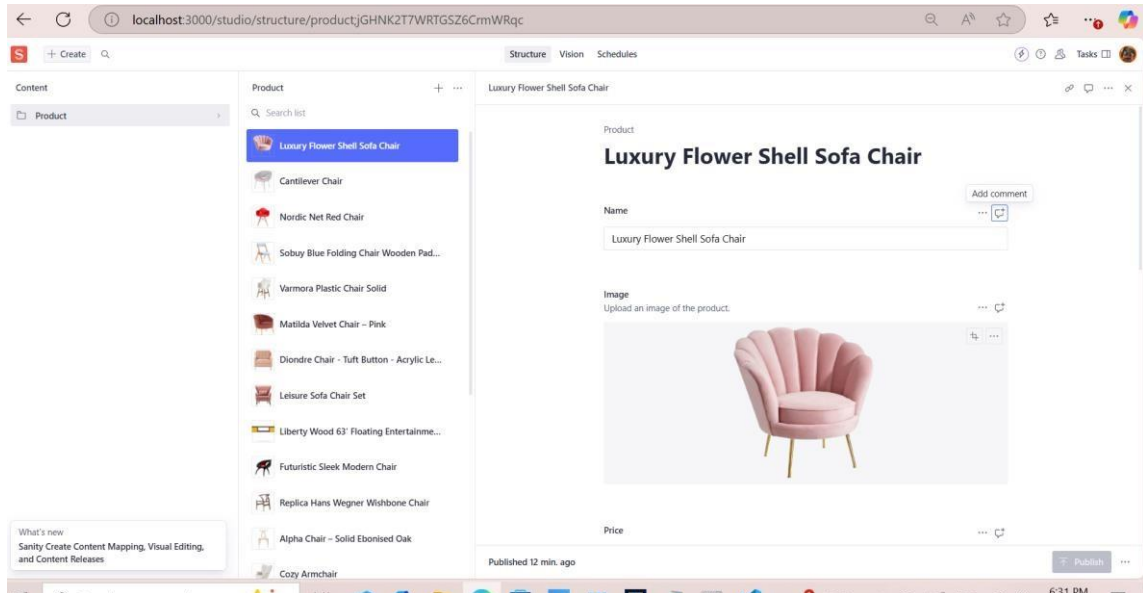
```
[_] 10 items
  0: { _id: "A3VJVObFUPWewZSbi4Ezxb", name: "Rapson Thirty-Nine Guest Chair", imageUrl: "https://cdn.sanity.io/images/bidw1gw5/production/fe1daa53f54c2706e2082623ded263d99f2b29be-490x490.png", price: 1300, description: "A mid-century modern chair with clean lines and durable materials.", discountPercentage: 15, stockLevel: 20, category: "Chair" }
  1: { _id: "A3VJVObFUPWewZSbi4F0FC", name: "Futuristic Sleek Modern Chair", imageUrl: "https://cdn.sanity.io/images/bidw1gw5/production/cc09d3ce34a323e7d60432a366145277d5bb3c4f-606x528.png", price: 2000, description: "A modern chair with a carbon fiber frame and bold red accents.", discountPercentage: 0, stockLevel: 3, category: "Chair" }
  2: { _id: "A3VJVObFUPWewZSbi4F1EF", name: "Compact armchair with a cozy design for small spaces.", imageUrl: "https://cdn.sanity.io/images/bidw1gw5/production/5209eafe1ea4da4a8b85ba319adf72ba2c7ae0df-356x356.png", price: 150, description: "A simple and lightweight chair ideal for dining or casual use.", discountPercentage: 0, stockLevel: 50, category: "Chair" }
```

Execution: 10ms End-to-end: 430ms

SHOWING POPULATED Data in Sanity:

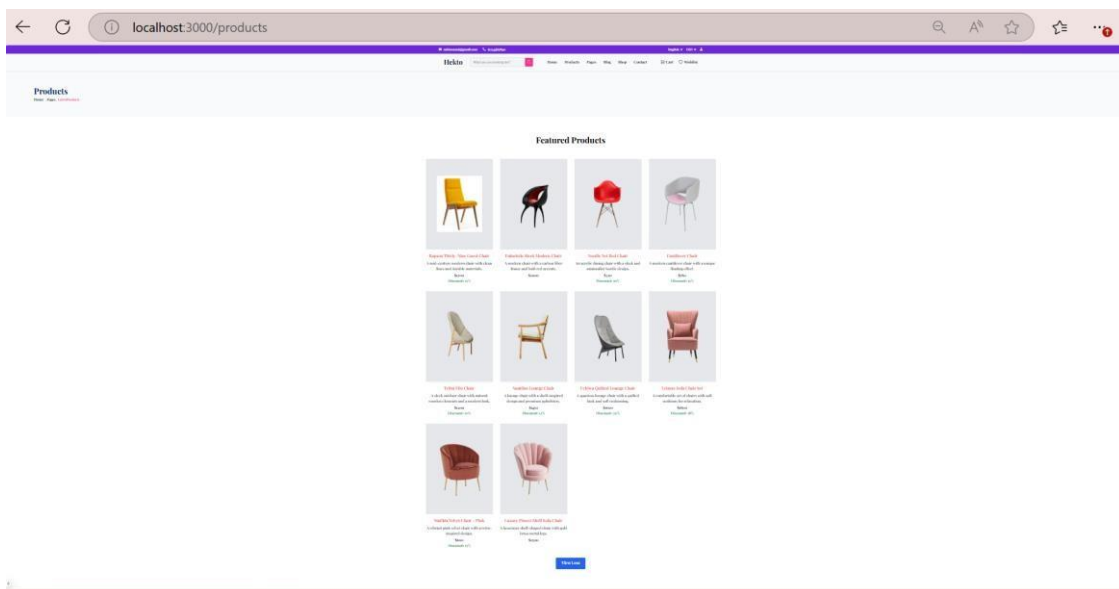
Sanity CMS populated with imported data from:

- *Provided API.*



Functional API integration in Next.js displaying:

- *Product listings.*
- *Categories.*
- *Other relevant data.*



SUMMARY OF THE WORK

Secure Sensitive Data

Defining Environmental Variables.

- Used .env files for storing API keys securely.

Data Validation During Migration

Applied schemas to check field types and constraints.

- Logged discrepancies for further analysis.

Thorough Documentation

Included screenshots, scripts, and testing notes.

- Maintained a changelog for schema adjustments.

Day 3 Checklist:

Self-Validation Checklist:

