

## DAY 3 REPORT – API INTEGRATION AND DATA MIGRATION

### 1. API Integration Process

This section explains how we connected the provided API to our project.

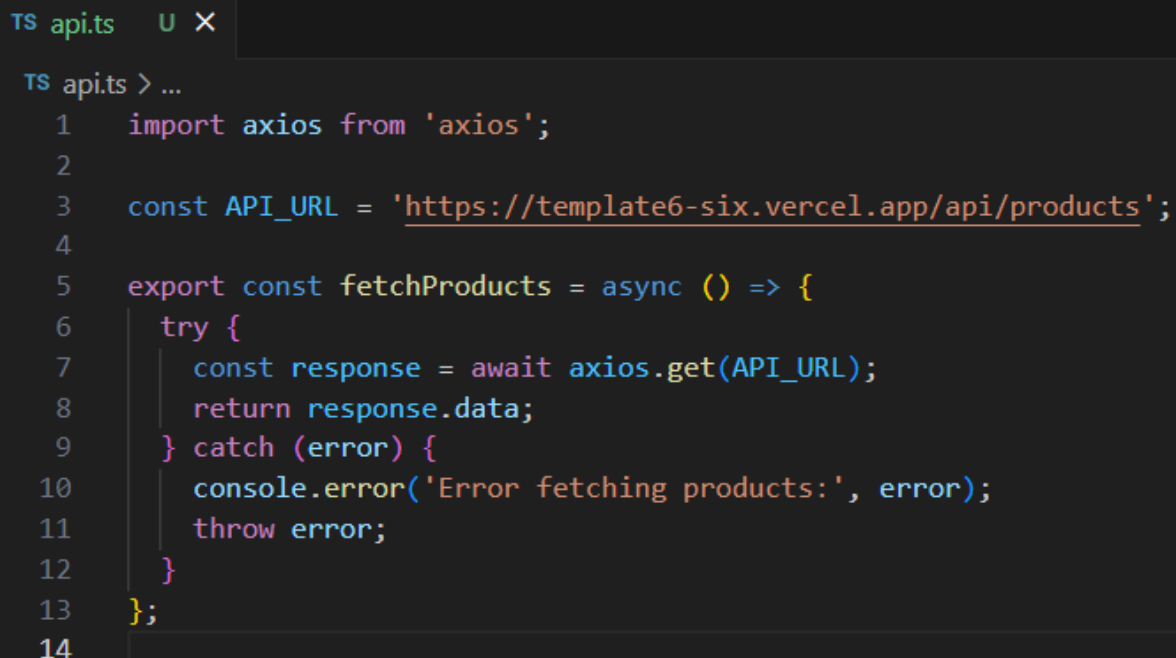
#### Steps:

##### 1. Understand the API

- ✓ Reviewed the API URL: <https://template6-six.vercel.app/api/products>.
- ✓ Used Postman to test API responses.
  - Example: Made a GET request to /products to retrieve a list of items.
  - Verified the response structure (i.e, product title, price, description).

##### 2. Set Up API in Next.js

- ✓ Installed Axios for making API calls: `npm install axios`.
- ✓ Created a utility file **api.ts**:



```
TS api.ts  U X
TS api.ts > ...
1  import axios from 'axios';
2
3  const API_URL = 'https://template6-six.vercel.app/api/products';
4
5  export const fetchProducts = async () => {
6    try {
7      const response = await axios.get(API_URL);
8      return response.data;
9    } catch (error) {
10     console.error('Error fetching products:', error);
11     throw error;
12   }
13 };
14
```

##### 3. Display Data on Frontend

- ✓ Used the **fetchProducts** function in **pages/index.ts** to display products:
- ✓ Tested by running `npm run dev` and viewing the data at <http://localhost:3000>.
- ✓ API request in Postman showing product data.
- ✓ Products displayed on the frontend.

## 2. Adjustments Made to Schemas

The provided schema needed changes to match our frontend design.

Changes:

### 1. Added Fields

A screenshot of a code editor window titled 'TS product.ts M X'. The editor shows a TypeScript file with a schema definition for a 'product' type using the 'defineType' function from 'sanity'. The schema includes fields for 'title', 'description', 'productImage', 'price', 'tags', 'dicountPercentage' (sic), and 'isNew'. Each field is defined with its name, title, type, and validation rules. The code is as follows:

```
template6 > src > sanity > schemaTypes > TS product.ts > ...
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      type: "string",
12      validation: (rule) => rule.required(),
13    },
14    {
15      name: "description",
16      type: "text",
17      title: "Description",
18      validation: (rule) => rule.required(),
19    },
20    {
21      name: "productImage",
22      type: "image",
23      title: "Product Image",
24      validation: (rule) => rule.required(),
25    },
26    {
27      name: "price",
28      type: "number",
29      title: "Price",
30      validation: (rule) => rule.required(),
31    },
32    {
33      name: "tags",
34      type: "array",
35      title: "Tags",
36      of: [{ type: "string" }],
37    },
38    {
39      name: "dicountPercentage",
40      type: "number",
41      title: "Discount Percentage",
42    },
43    {
44      name: "isNew",
45      type: "boolean",
46      title: "New Badge",
47    },
48  ],
49 });
```

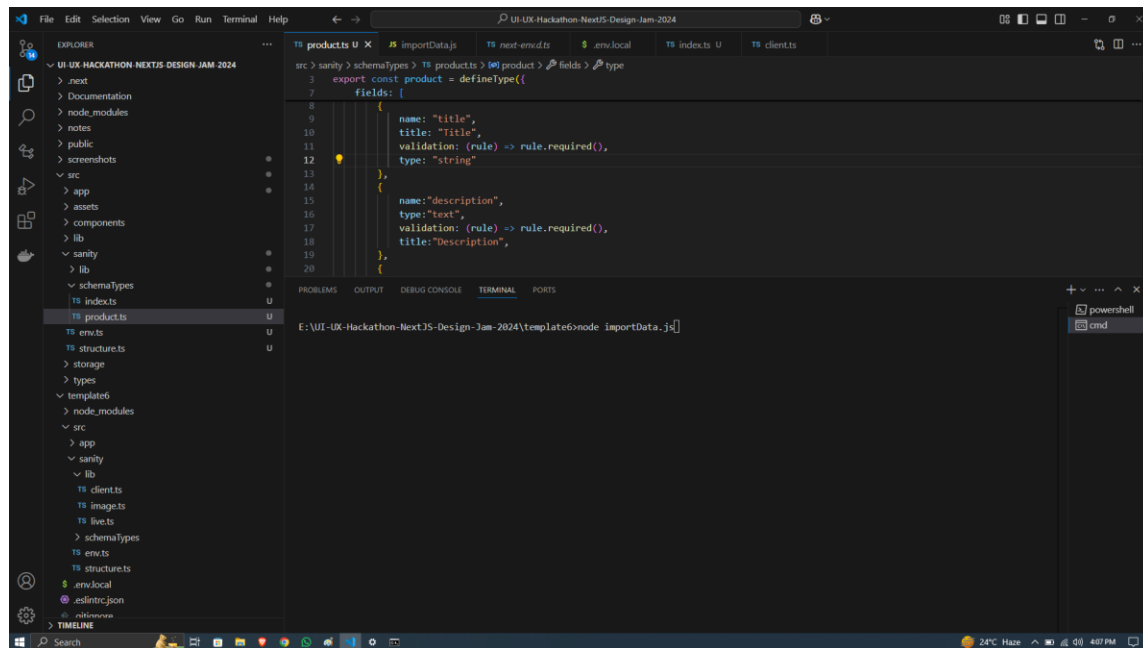
- ✓ Modified Field Validations
- ✓ Adjusted schema in Sanity Studio.

### 3. Migration Steps and Tools Used

Steps:

1. Used the Provided Migration Script
2. Imported Data into Sanity

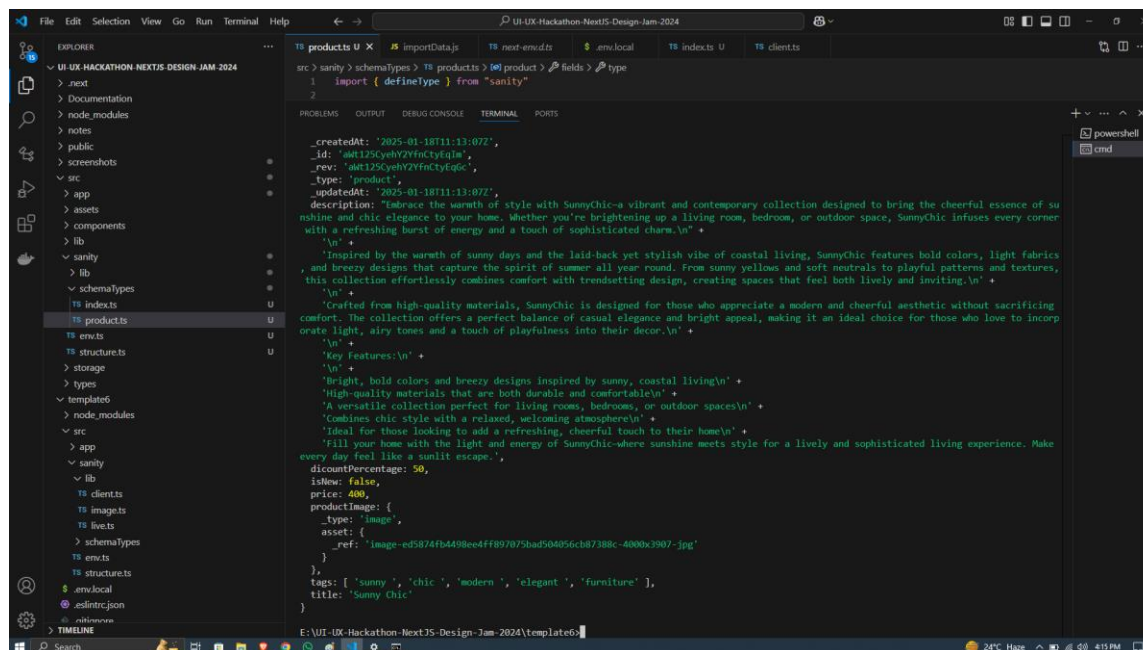
CMD Before Compilation:



```
src > sanity > schemaTypes > TS products > [0] product > [0] fields > [0] type
3 export const product = defineType({
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 });
```

```
E:\UI-UX-Hackathon-NextJS-Design-Jam-2024\template6>node importData.js
```

CMD After Compilation:



```
src > sanity > schemaTypes > TS products > [0] product > [0] fields > [0] type
1 import { defineType } from "sanity"
2
```

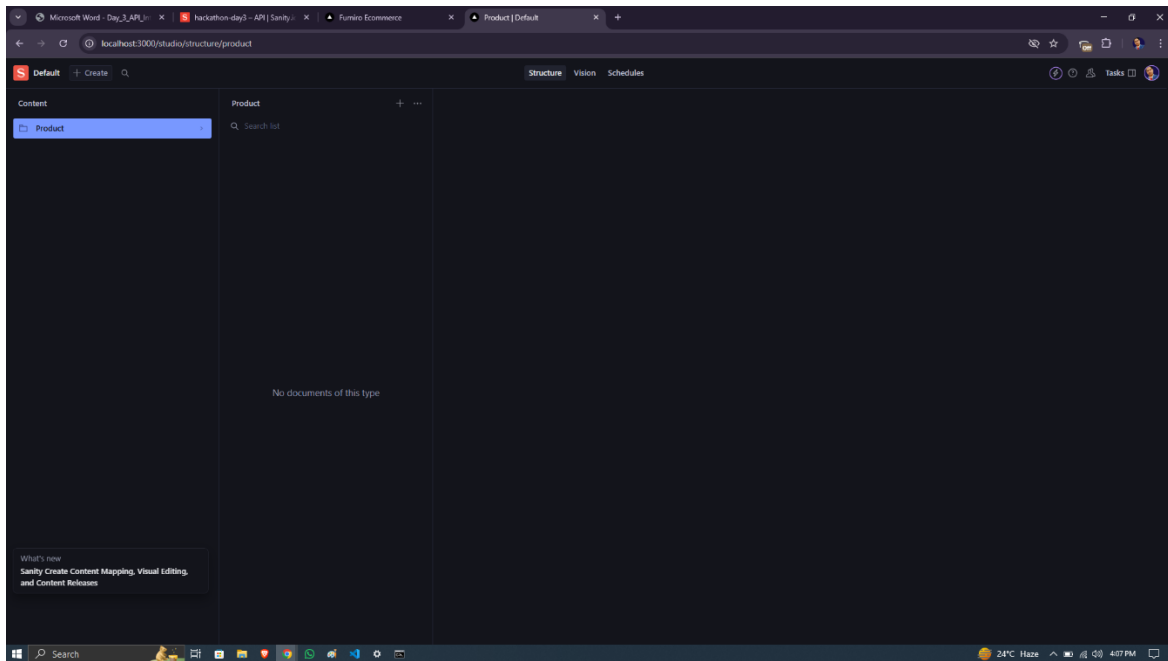
```
{
  "_createdAt": "2025-01-18T11:13:07Z",
  "_id": "skt12ScyehY2FnCtyfale",
  "_rev": "skt12ScyehY2FnCtyfale",
  "_type": "product",
  "_updatedAt": "2025-01-18T11:13:07Z",
  "description": "Embrace the warmth of style with SunnyChic—a vibrant and contemporary collection designed to bring the cheerful essence of sunshine and chic elegance to your home. Whether you're brightening up a living room, bedroom, or outdoor space, SunnyChic infuses every corner with a refreshing burst of energy and a touch of sophisticated charm.\n",
  "keyFeatures": "\n",
  "inspiredBy": "Inspired by the warmth of sunny days and the laid-back yet stylish vibe of coastal living, SunnyChic features bold colors, light fabrics, and breezy designs that capture the spirit of summer all year round. From sunny yellow and soft neutrals to playful patterns and textures, this collection effortlessly combines comfort with trendsetting design, creating spaces that feel both lively and inviting.\n",
  "materials": "\n",
  "craftedFrom": "Crafted from high-quality materials, SunnyChic is designed for those who appreciate a modern and cheerful aesthetic without sacrificing comfort. The collection offers a perfect balance of casual elegance and bright appeal, making it an ideal choice for those who love to incorporate light, airy tones and a touch of playfulness into their decor.\n",
  "keyFeatures": "\n",
  "inspiredBy": "\n",
  "materials": "Bright, bold colors and breezy designs inspired by sunny, coastal living\n",
  "craftedFrom": "High-quality materials that are both durable and comfortable\n",
  "inspiredBy": "A versatile collection perfect for living rooms, bedrooms, or outdoor spaces\n",
  "craftedFrom": "Combines chic style with a relaxed, welcoming atmosphere\n",
  "inspiredBy": "Ideal for those looking to add a refreshing, cheerful touch to their home\n",
  "craftedFrom": "Fill your home with the light and energy of SunnyChic—where sunshine meets style for a lively and sophisticated living experience. Make every day feel like a sunlit escape.",
  "discountPercentage": 50,
  "isNew": false,
  "price": 400,
  "productImage": {
    "type": "image",
    "asset": {
      "_ref": "image-ed5874fb448ee4ff897075bad504056cb87388c-400bc3907-jpg"
    }
  },
  "tags": [ "sunny", "chic", "modern", "elegant", "furniture" ],
  "title": "Sunny Chic"
}
```

```
E:\UI-UX-Hackathon-NextJS-Design-Jam-2024\template6>
```

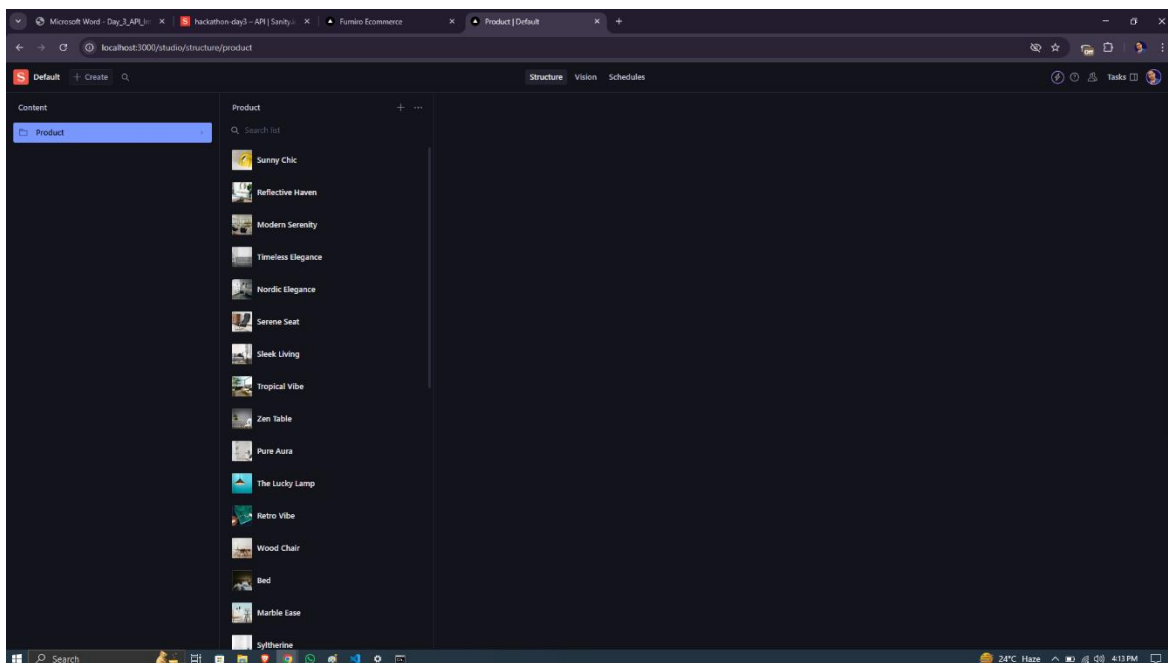
## Tools Used:

- ✓ Postman (for API testing).
- ✓ Node.js (for running migration scripts).
- ✓ Sanity Studio (for schema and data management).
- ✓ Populated fields in Sanity Studio.

## Sanity Dashboard Before Migration:



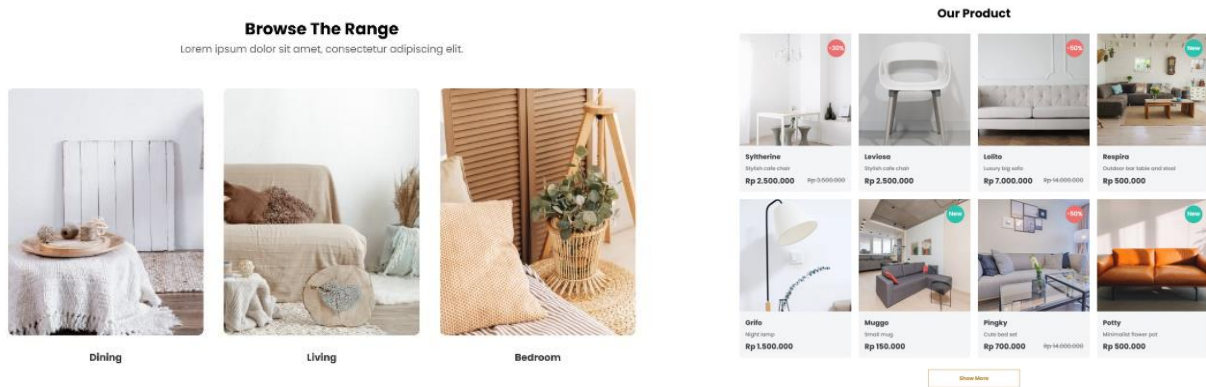
## Sanity Dashboard After Migration:



## 4. Final Results

### Data Displayed on Frontend

- ✓ Successfully fetched data from Sanity and displayed it using Next.js components.
- ✓ Added styles to match the frontend design.



### Conclusion

- ✓ **API Integration:** Successfully connected and fetched data from the provided API.
- ✓ **Schema Adjustments:** Customized to include category and stock fields.
- ✓ **Migration:** Imported product data into Sanity Studio using a modified script.