

Ammar Shaikh

12014

Tuesday 2-5 (evening)

API Integration Report - [HEKTO WEBSITE]

Reviewed API Documentation:

- I carefully read the provided API documentation for my assigned template to understand the available endpoint (/products).
- I identified the structure of the data returned by the API, including field names and data types.

Set Up API Calls:

- I used Thunder client to test the API endpoint and ensure the data was being returned correctly. I created utility functions in my Next.js project to fetch data from the API.
- I used fetch to make GET requests to the API endpoints and stored the responses in variables. I logged the API responses in the console to verify the data structure.

Compared API Data with Sanity Schema:

- I reviewed the API data structure and compared it with the existing schema in Sanity CMS. I identified mismatches in field names and data types.
- I updated the Sanity schema to match the API data structure. For example:
 - └ API Field: product_title → Sanity Field: name
 - └ API Field: price → Sanity Field: price (with proper data type)

I added new fields in Sanity CMS to accommodate additional data from the API , because the API is not enough to complete my website products and their details .

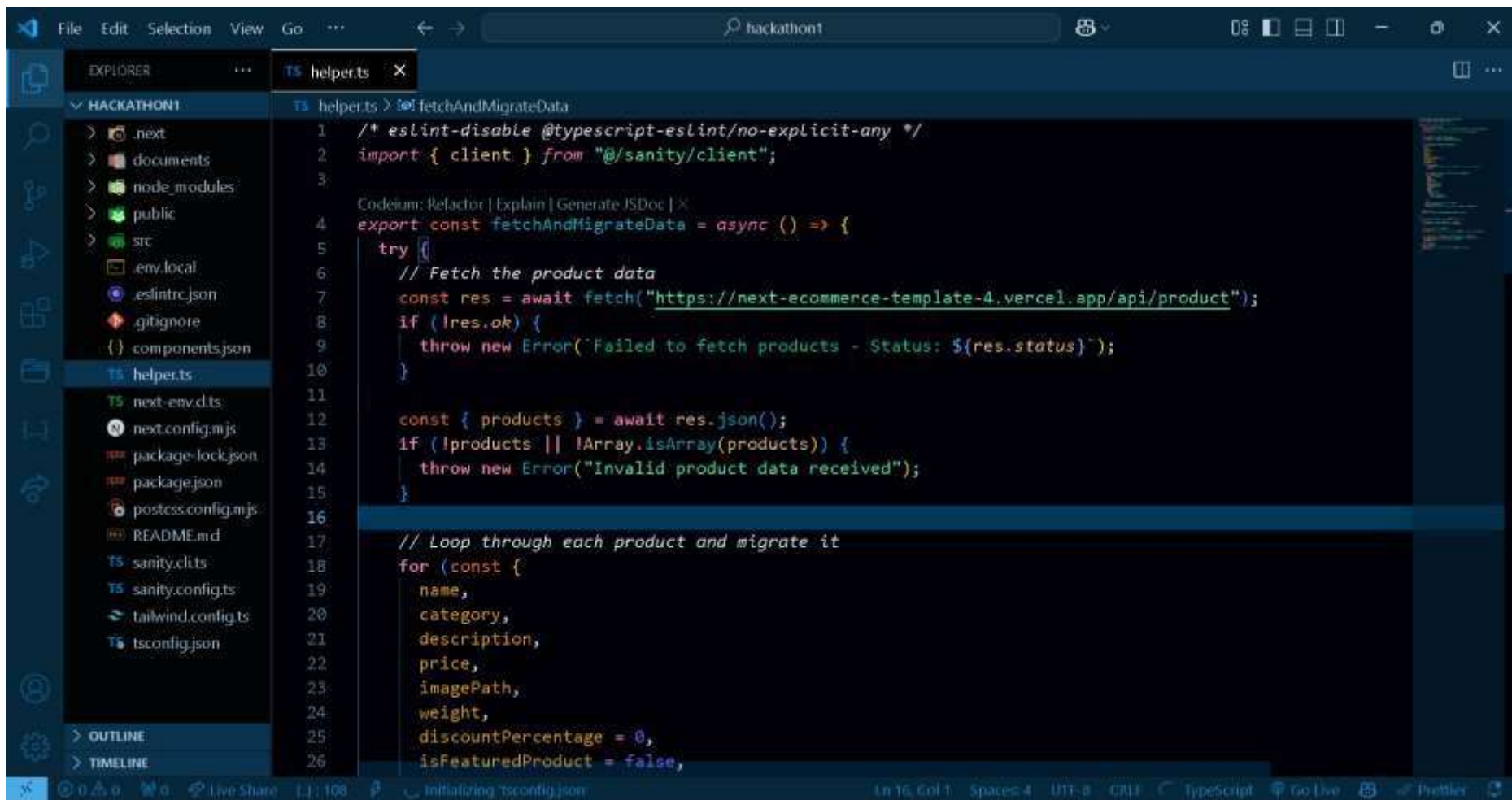


To migrate data from the API to Sanity CMS, I followed these steps:

- I decided to use the provided API to fetch data and write a script to import it into Sanity CMS.
- I created a script Folder and then i created a custom migration (.ts) file to fetch data from the API and transform it into the format required by Sanity CMS.
- I used the Sanity client library to upload the data to the CMS. I ran the migration script to import product data, categories, and other relevant information into Sanity CMS.
- I verified the imported data by checking the Sanity dashboard and ensuring all fields were correctly populated.

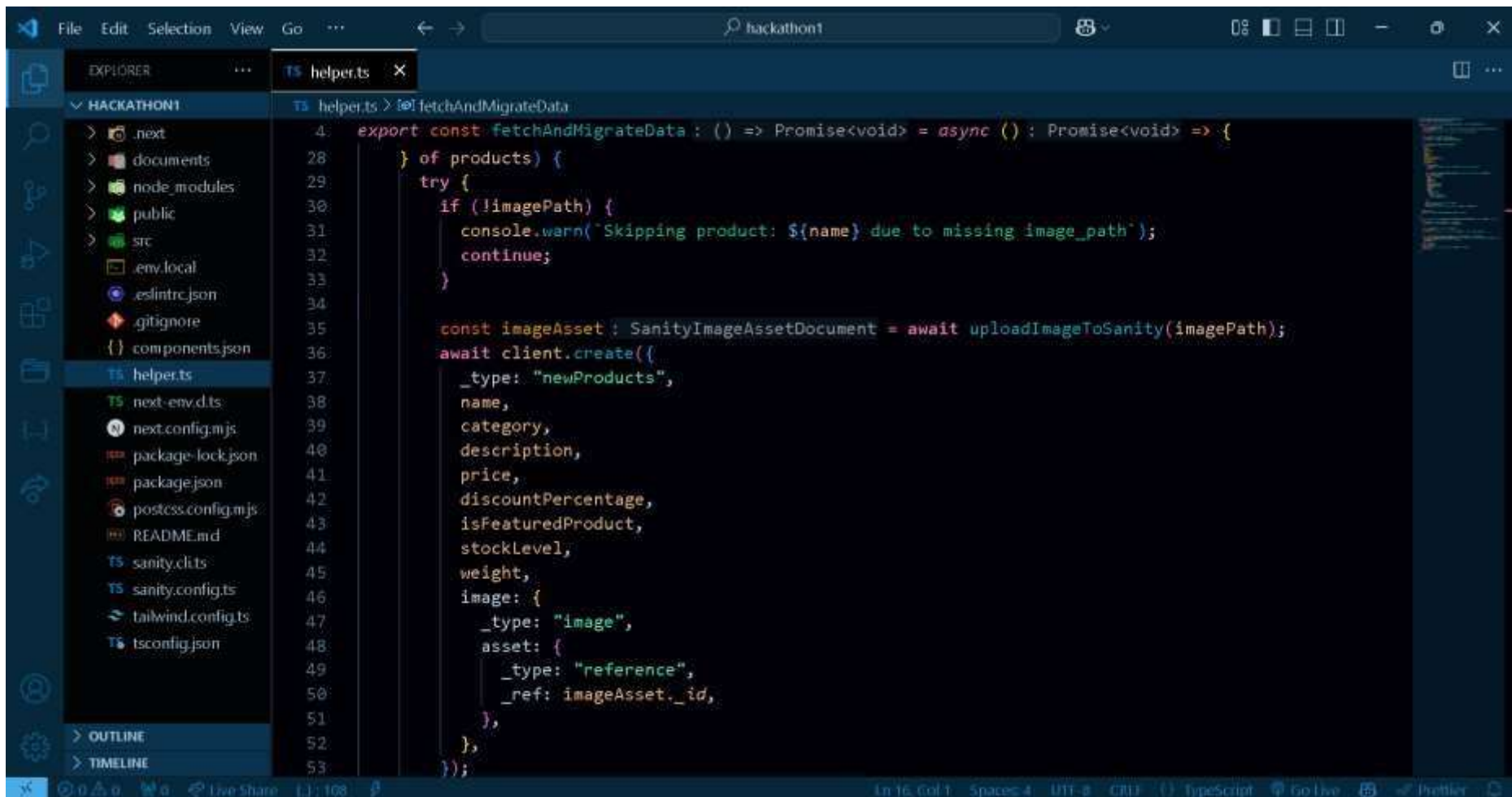
In this project, I successfully integrated the provided API into my Next.js frontend and migrated data into Sanity CMS. I adjusted the schema to match the API data structure and ensured the data was accurately displayed in the frontend. This exercise helped me gain practical experience in API integration, data migration, and schema validation, which are essential skills for building scalable marketplaces.

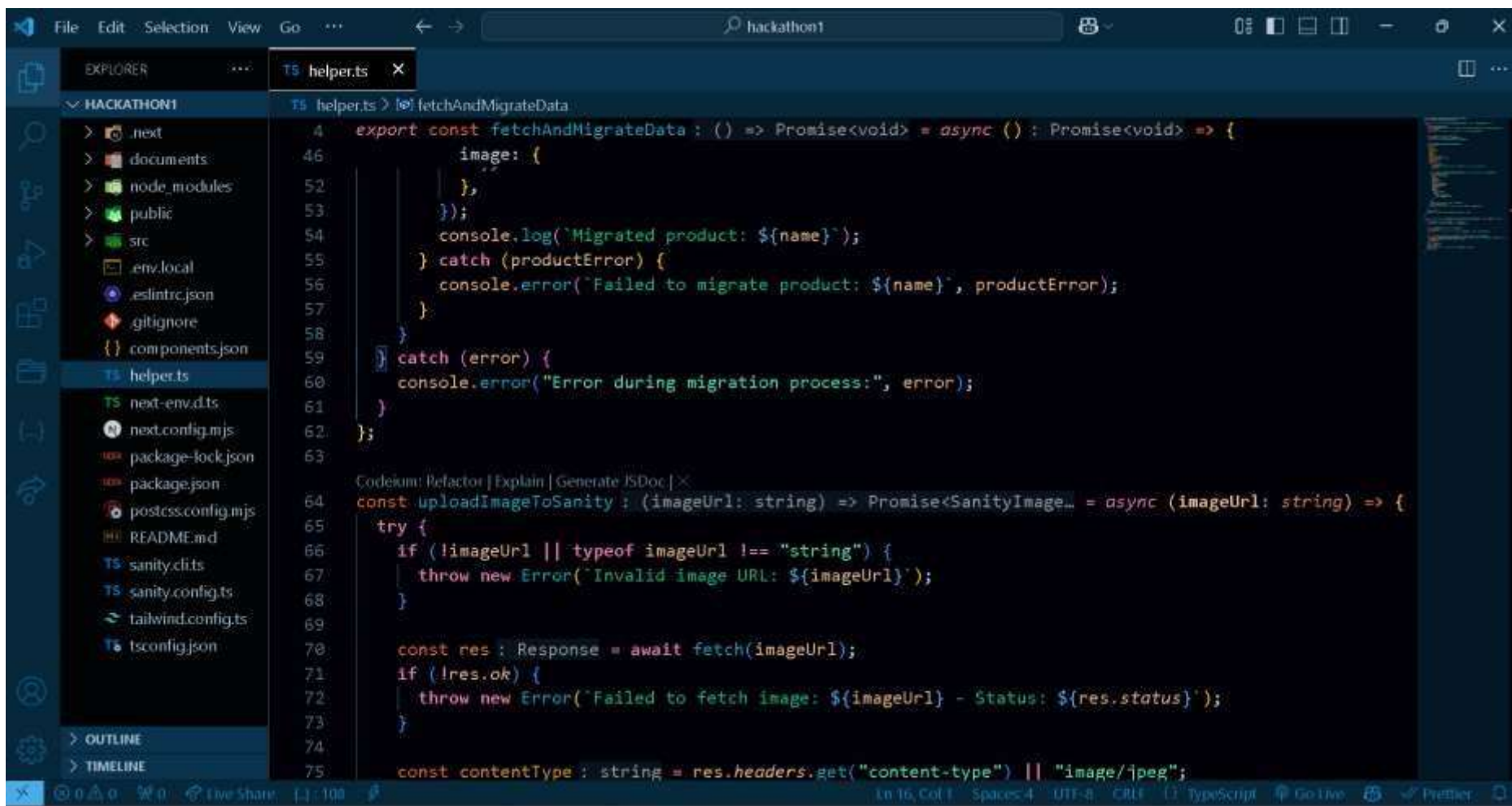
Task	Status
API Understanding	✓
Schema Validation	✓
Data Migration	✓
API Integration in Next.js	✓
Submission Preparation	✓

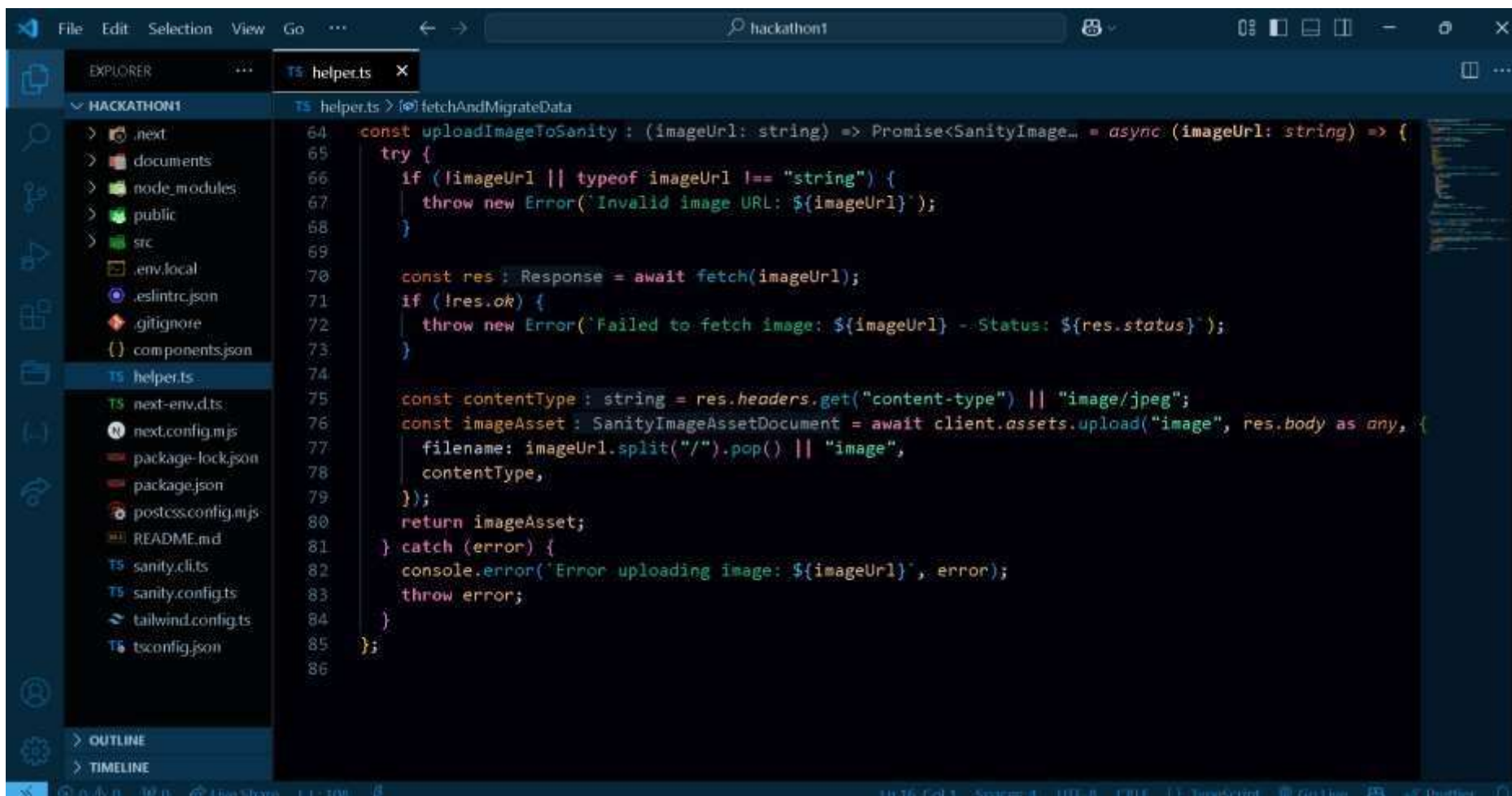


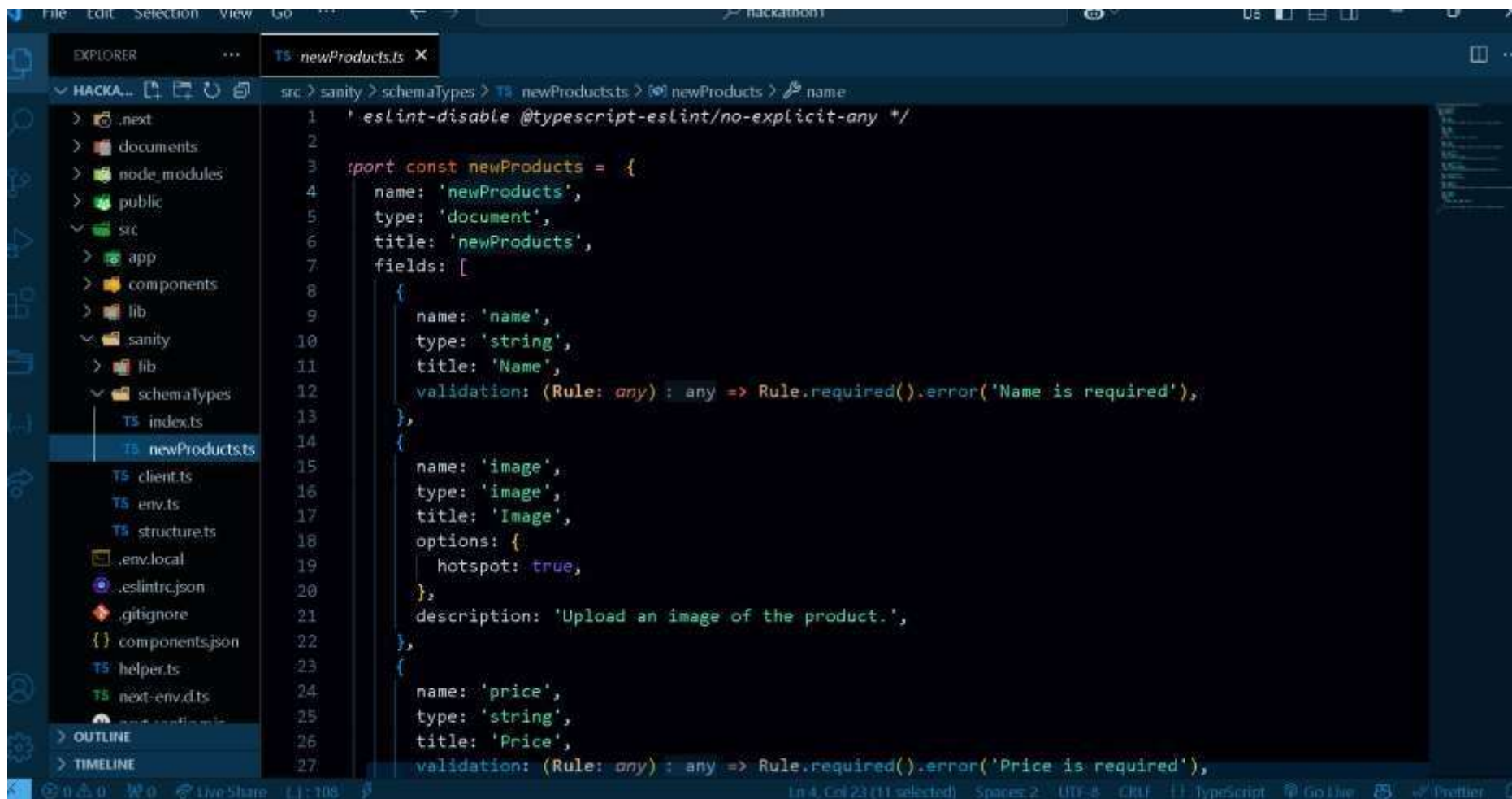
The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left displays the file structure of a project named 'HACKATHON1'. The main editor area shows the contents of 'helper.ts', which is a TypeScript file. The code defines a function 'fetchAndMigrateData' that uses 'async/await' to fetch product data from an API and then migrates it. The status bar at the bottom indicates the current file is 'helper.ts' and shows various settings like 'Ln 16, Col 1' and 'Spaces: 4'.

```
1  /* eslint-disable @typescript-eslint/no-explicit-any */
2  import { client } from "@sanity/client";
3
4  Codeium: Refactor | Explain | Generate JSDoc | X
5  export const fetchAndMigrateData = async () => {
6    try {
7      // Fetch the product data
8      const res = await fetch("https://next-ecommerce-template-4.vercel.app/api/product");
9      if (!res.ok) {
10        throw new Error(`Failed to fetch products - Status: ${res.status}`);
11      }
12
13      const { products } = await res.json();
14      if (!products || !Array.isArray(products)) {
15        throw new Error("Invalid product data received");
16      }
17
18      // Loop through each product and migrate it
19      for (const {
20        name,
21        category,
22        description,
23        price,
24        imagePath,
25        weight,
26        discountPercentage = 0,
27        isFeaturedProduct = false,
```









File Edit Selection View Go ... hackathon1

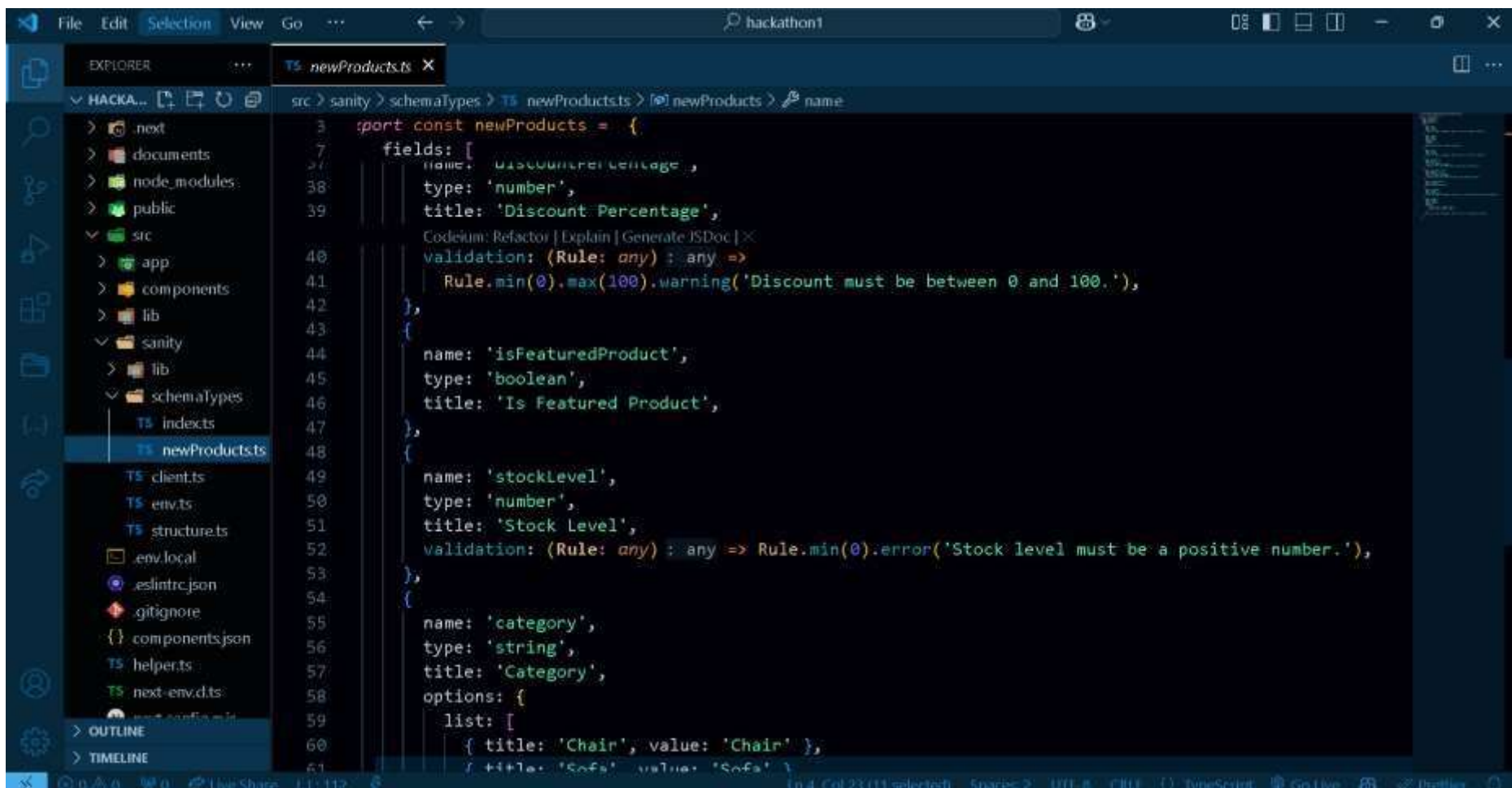
EXPLORER

HACKA... [src > sanity > schemaTypes > TS newProducts.ts > newProducts > name]

```
3  import const newProducts = {
7    fields: [
21      {
22        description: 'Upload an image of the product.',
23      },
24      {
25        name: 'price',
26        type: 'string',
27        title: 'Price',
28        validation: (Rule: any) : any => Rule.required().error('Price is required'),
29      },
30      {
31        name: 'description',
32        type: 'text',
33        title: 'Description',
34        validation: (Rule: any) : any =>
35          Rule.max(150).warning('Keep the description under 150 characters.'),
36      },
37      {
38        name: 'discountPercentage',
39        type: 'number',
40        title: 'Discount Percentage',
41        validation: (Rule: any) : any =>
42          Rule.min(0).max(100).warning('Discount must be between 0 and 100.'),
43      },
44    ],
45  }
```

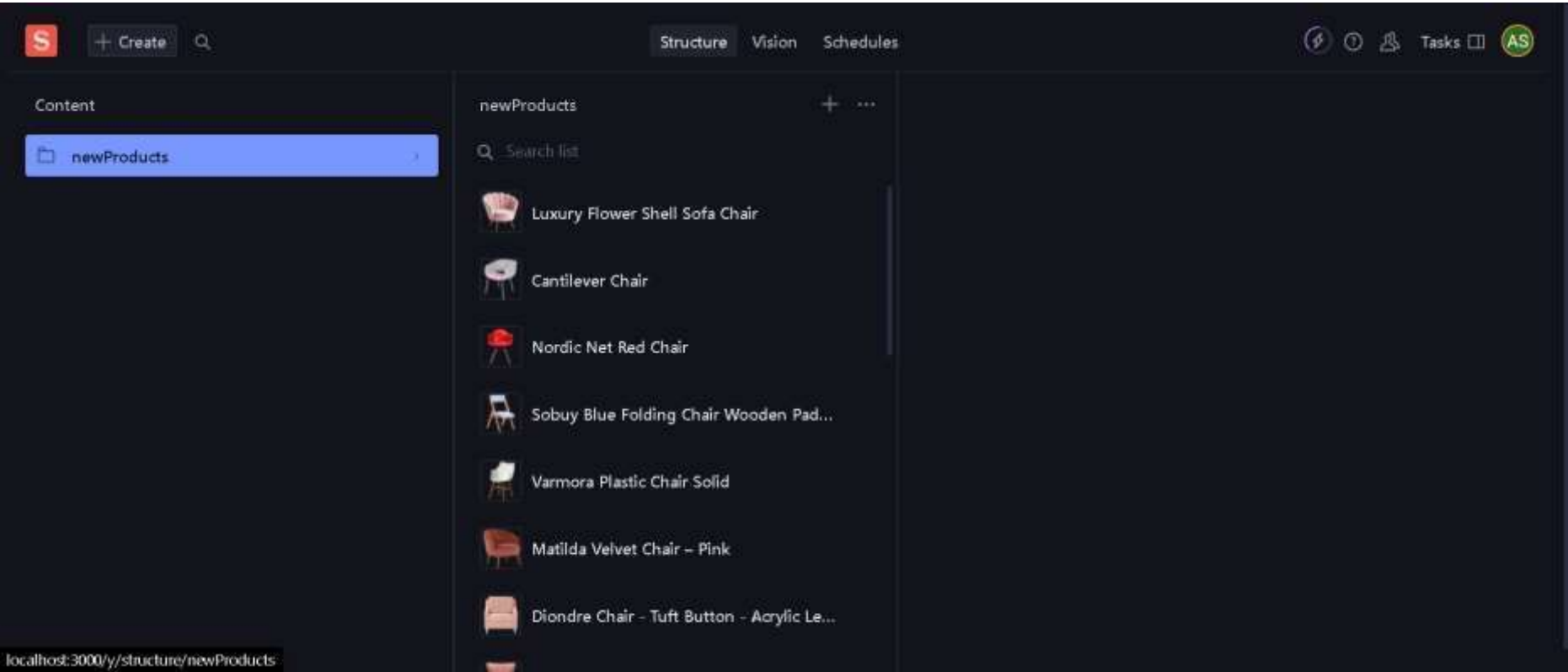
Codeium: Refactor | Explain | Generate JSDoc | X

Ln 4, Col 21 (11 selected) Spaces: 2 UTF-8 CRLF TypeScript Go Live Prettier



```
... TS newProducts.ts X
src > sanity > schemaTypes > TS newProducts.ts > [e] newProducts > name
3  import const newProducts = {
7    fields: [
49      name: 'stockLevel',
50      type: 'number',
51      title: 'Stock Level',
52      validation: (Rule: any) : any => Rule.min(0).error('Stock level must be a positive number.'),
53    ],
54    {
55      name: 'category',
56      type: 'string',
57      title: 'Category',
58      options: {
59        list: [
60          { title: 'Chair', value: 'Chair' },
61          { title: 'Sofa', value: 'Sofa' },
62        ],
63      },
64      validation: (Rule: any) : any => Rule.required().error('Category is required'),
65    ],
66  ],
67  };

```

Featured Products



Tribù Elio Chair

1200 \$



Armchair Tortuga

850 \$



Stylish Golden Metal Leg...

780 \$



Cozy Armchair

520 \$



Tribù Elio Chair
1200 \$



Armchair Tortuga
850 \$



Stylish Golden Metal Le...
780 \$



Cozy Armchair
520 \$





Stylish Golden Metal Legs Mint Blue Fabric Velvet Sofa Leisure Armchair



\$780 \$

Color

A luxurious armchair with velvet fabric and golden metal legs.

Add To Cart

Categories: Sofa

Weight:

Share

