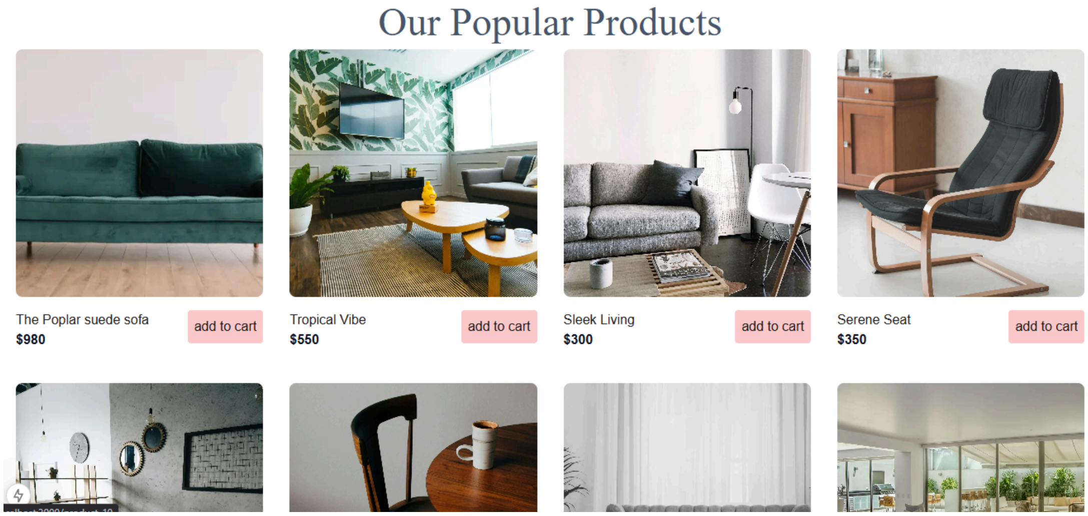
MARKET PLACE HACKATHON

DAY 03

Presented By:Darakhshan Jan 18, 2025



Our Popular Products

The Poplar suede sofa
$980

Tropical Vibe
$550

Sleek Living
$300

Serene Seat
$350

This documentation will guide you through the setup and implementation of a Sanity schema, a migration script for transferring data to another Sanity account, and overall project setup

1 : Install Nextjs Project :-

Command: npx create-next-app@14.2.20 .

2 : Sanity :-

Create a new project of sanity in sanity dashboard Then run the project installation command in your next js project.

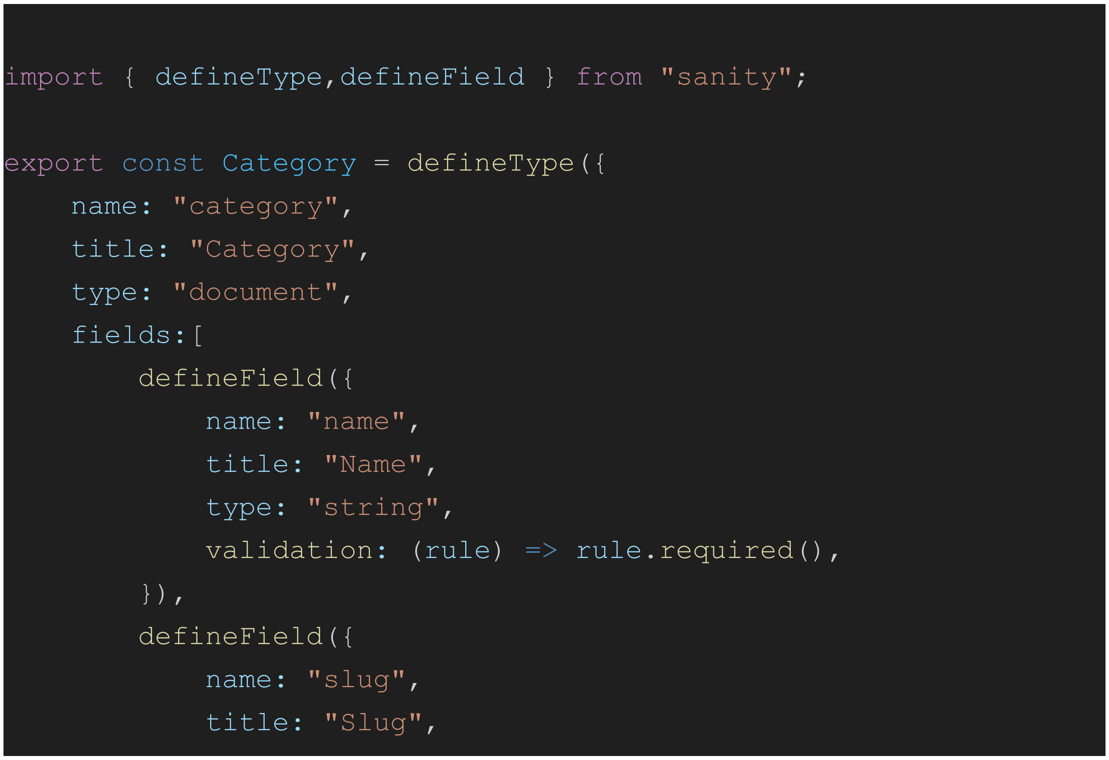
3: Sanity Schema :-

Create a new project of sanity in sanity dashboard

1: After the Installation navigate to schema folder: /src/sanity/schemaTypes.

4 : Our Schema

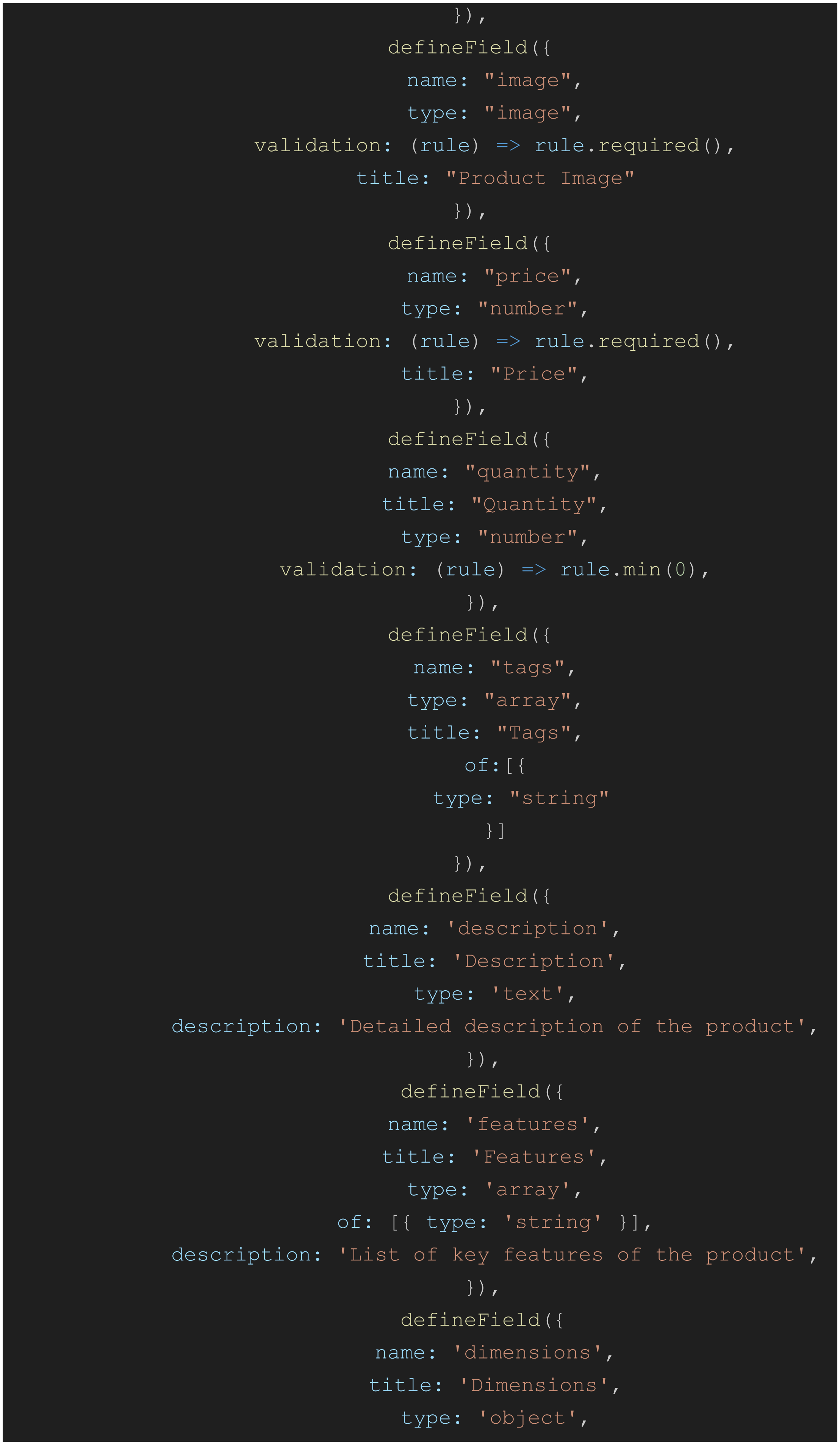
1: Create two new files and add the following code: products.ts



```
import { defineType,defineField } from "sanity";

export const Category = defineType({
    name: "category",
    title: "Category",
    type: "document",
    fields:[
        defineField({
            name: "name",
            title: "Name",
            type: "string",
            validation: (rule) => rule.required(),
        }),
        defineField({
            name: "slug",
            title: "Slug",
            type: "slug",
            validation: (rule) => rule.required(),
            options: {
                source: "name",
            }
        })
    ]
})
```

2 : Create two new files and add the following code: category.ts

```
         import { defineType, defineField } from "sanity"

               export const product = defineType({
                           name: "product",
                           title: "Product",
                           type: "document",
                               fields: [
                               defineField({
                               name:"category",
                               title:"Category",
                               type:"reference",
                                     to:[{
                                  type:"category"
                                       }]
                                     }
                                    ),
                               defineField({
                                 name: "name",
                                title: "Title",
                    validation: (rule) => rule.required(),
                                type: "string"
                                    }),
                               defineField({
                                 name: "slug",
                                title: "Slug",
                    validation: (rule) => rule.required(),
                                 type: "slug"
```



```
}),
defineField({
name: "image",
type: "image",
validation: (rule) => rule.required(),
title: "Product Image"
}),
defineField({
name: "price",
type: "number",
validation: (rule) => rule.required(),
title: "Price",
}),
defineField({
name: "quantity",
title: "Quantity",
type: "number",
validation: (rule) => rule.min(0),
}),
defineField({
name: "tags",
type: "array",
title: "Tags",
of:[{
type: "string"
}]
}),
defineField({
name: 'description',
title: 'Description',
type: 'text',
description: 'Detailed description of the product',
}),
defineField({
name: 'features',
title: 'Features',
type: 'array',
of: [{ type: 'string' }],
description: 'List of key features of the product',
}),
defineField({
name: 'dimensions',
title: 'Dimensions',
type: 'object',
fields: [
{ name: 'height', title: 'Height', type: 'string'
},
{ name: 'width', title: 'Width', type: 'string' },
{ name: 'depth', title: 'Depth', type: 'string' },
],
description: 'Dimensions of the product',
}),
]
})
```

2. Importing Schemas in index.ts
○ src/sanity/schemaTypes/index.ts

```
import { type SchemaTypeDefinition } from 'sanity'
import { product } from "./product";
import {Category} from "./category"
export const schema: { types: SchemaTypeDefinition[] } = {
  types: [product , Category ],
}
```

5: Environment Variables:

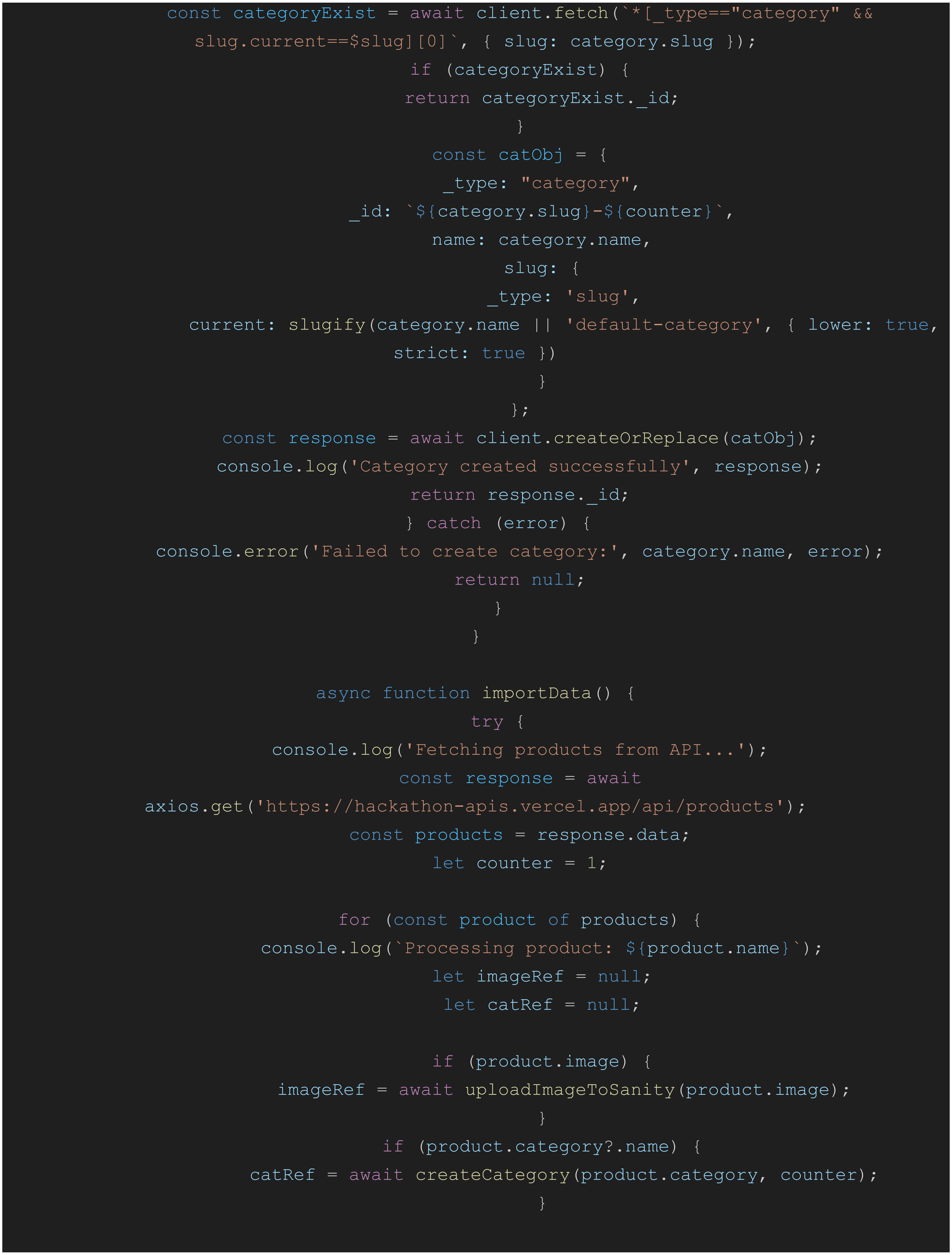
Create .env file and add the following variables :

```
SANITY_PROJECT_ID=YOUT-PROJECT-ID
SANITY_DATASET=production
API_TOKEN=your-token
```

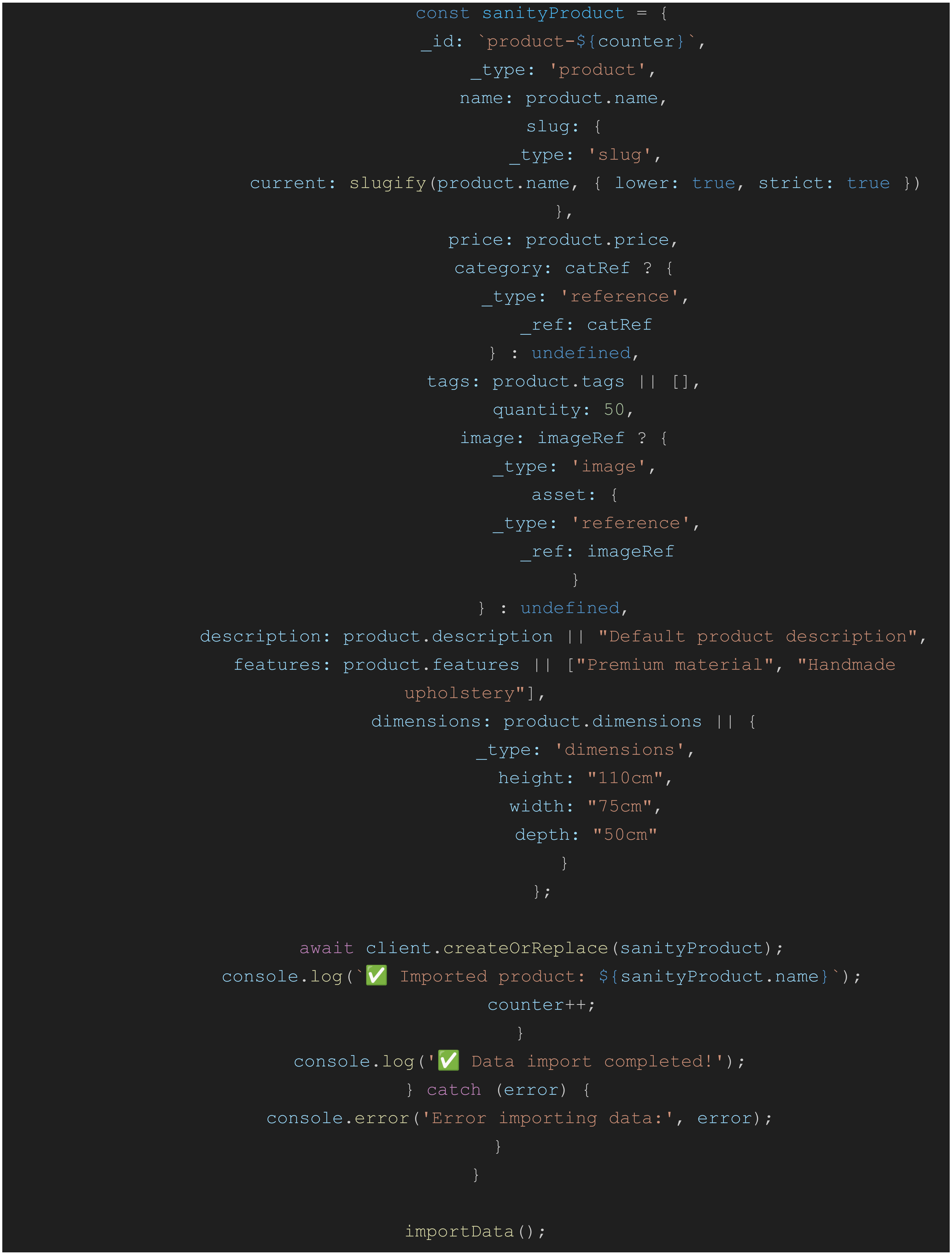
6: Data Migration Script:

When you pull these changes, ensuring consistency between your Sanity dataset and the API's data structure.data from an external API, the structure of the API data may change over time, or you may need to adapt your local schema to accommodate new fields, relationships, or types. Migration files allow you to systematically manage.

```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';
import slugify from 'slugify';

const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

const client = createClient({
projectId:"sh222unl",
dataset:"production" ,
useCdn: false,

token:"skI4iLVpxIaTJ6trkxft63IZKM6hnJ39ZtUhZCeNRfPHQk3LkKWSqXlCucICZHBMN0kUCzDIsevmYc
o1SLplDSrVBH3C48QI1SSBDWDQmdOAIYET6EPx4wE0J6MH4icM3GaiJbwJhr1WqJhX93VlQ8vAWH9yRW0cVRH
H0zYLi8QhpjKNK4EJ",
apiVersion: '2021-08-31'
});

async function uploadImageToSanity(imageUrl) {
try {
console.log(`Uploading image: ${imageUrl}`);
const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
const buffer = Buffer.from(response.data);
const asset = await client.assets.upload('image', buffer, {
filename: imageUrl.split('/').pop()
});
console.log(`Image uploaded successfully: ${asset._id}`);
return asset._id;
} catch (error) {
console.error('Failed to upload image:', imageUrl, error);
return null;
}
}

async function createCategory(category, counter) {
try {
```



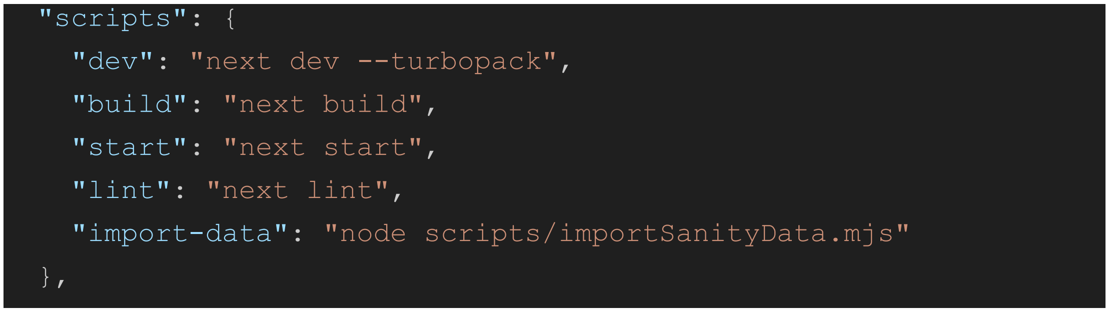
```
  const categoryExist = await client.fetch(`*[_type=="category" &&
     slug.current==$slug][0]`, { slug: category.slug });
                        if (categoryExist) {
                       return categoryExist._id;
                                  }
                          const catObj = {
                           _type: "category",
                  _id: `${category.slug}-${counter}`,
                          name: category.name,
                                 slug: {
                               _type: 'slug',
    current: slugify(category.name || 'default-category', { lower: true,
                       strict: true })
                                    }
                                 };
       const response = await client.createOrReplace(catObj);
       console.log('Category created successfully', response);
                        return response._id;
                        } catch (error) {
 console.error('Failed to create category:', category.name, error);
                            return null;
                                }
                              }

                async function importData() {
                              try {
            console.log('Fetching products from API...');
                       const response = await
axios.get('https://hackathon-apis.vercel.app/api/products');
                   const products = response.data;
                          let counter = 1;

                  for (const product of products) {
           console.log(`Processing product: ${product.name}`);
                          let imageRef = null;
                           let catRef = null;

                          if (product.image) {
            imageRef = await uploadImageToSanity(product.image);
                                    }
                      if (product.category?.name) {
         catRef = await createCategory(product.category, counter);
                                    }
```



```
const sanityProduct = {
_id: `product-${counter}`,
_type: 'product',
name: product.name,
slug: {
_type: 'slug',
current: slugify(product.name, { lower: true, strict: true })
},
price: product.price,
category: catRef ? {
_type: 'reference',
_ref: catRef
} : undefined,
tags: product.tags || [],
quantity: 50,
image: imageRef ? {
_type: 'image',
asset: {
_type: 'reference',
_ref: imageRef
}
} : undefined,
description: product.description || "Default product description",
features: product.features || ["Premium material", "Handmade upholstery"],
dimensions: product.dimensions || {
_type: 'dimensions',
height: "110cm",
width: "75cm",
depth: "50cm"
}
};

await client.createOrReplace(sanityProduct);
console.log(`✅ Imported product: ${sanityProduct.name}`);
counter++;
}
console.log('✅ Data import completed!');
} catch (error) {
console.error('Error importing data:', error);
}
}

importData();
```

2: Open`package.json` file :

add the following code inside of scripts:



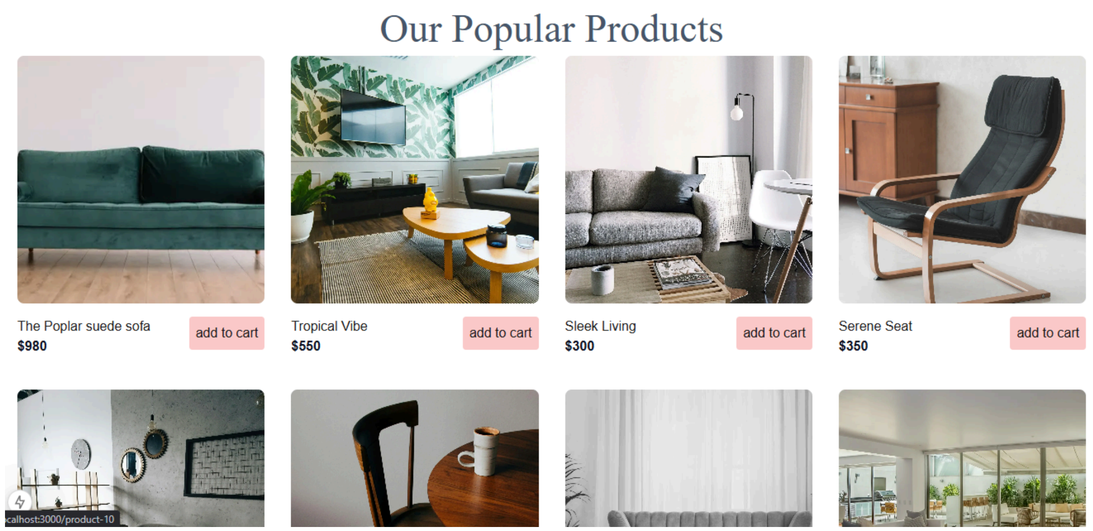
```
"scripts": {
    "dev": "next dev --turbopack",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "import-data": "node scripts/importSanityData.mjs"
  },
```

3: Install the following package before running the script : npm install dotenv

4: Now run the command npm run migrate

Command for import data : Npm run import-data

Write Query for fetch data from sanity to ui



Our Popular Products

The Poplar suede sofa
$980
add to cart

Tropical Vibe
$550
add to cart

Sleek Living
$300
add to cart

Serene Seat
$350
add to cart