Day 3 – API Integration Report for Furniro

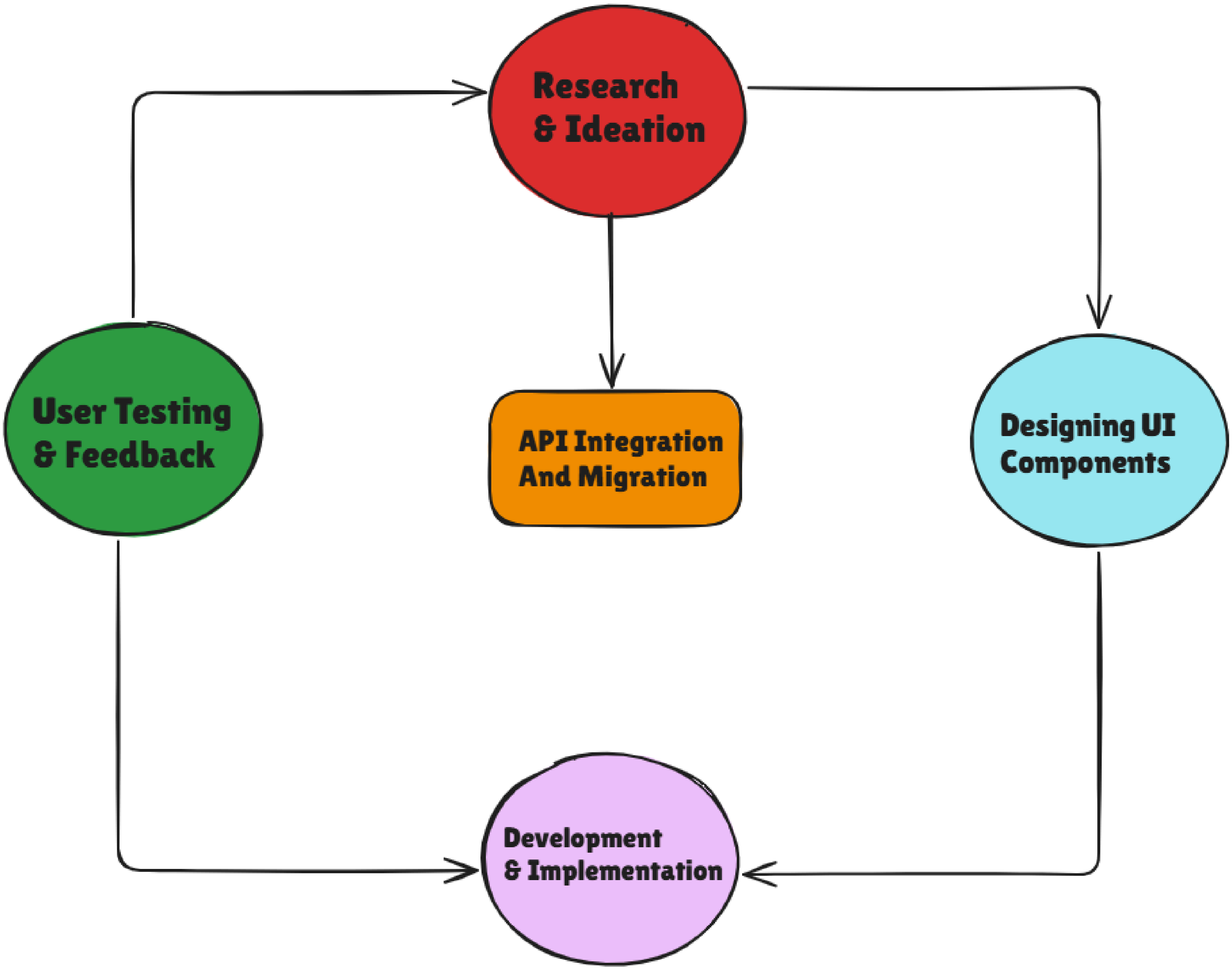
1. Introduction

This report outlines the integration process of product and category data from an external API into the backend system of Furniro, an e-commerce platform specializing in furniture. The integration leverages Sanity CMS for content management and Next.js for frontend rendering.

Objectives

The main objectives of this integration were:

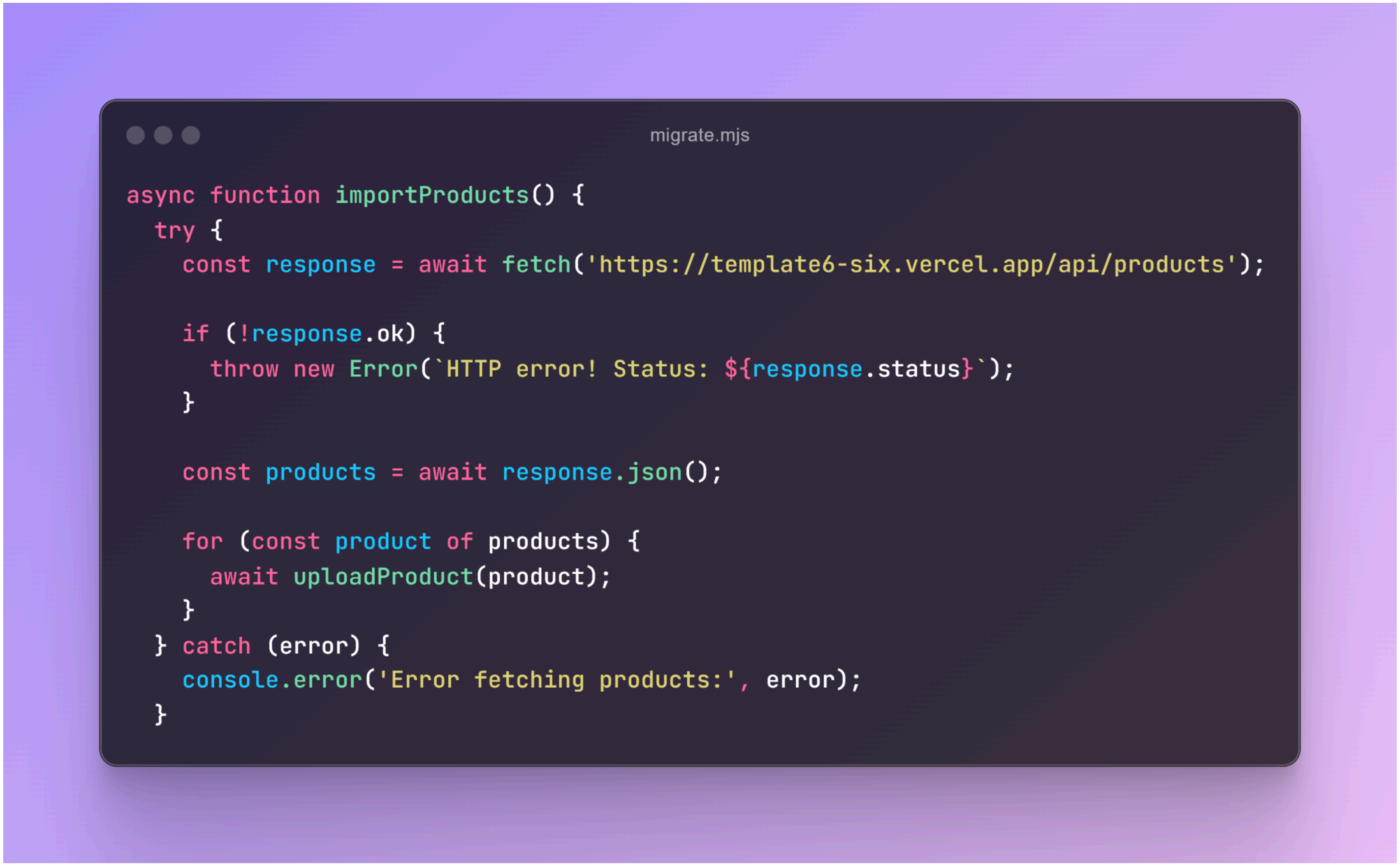
1. Fetching product and category data from an external API.
2. Storing and managing the data in Sanity CMS.
3. Dynamically displaying the fetched data on the frontend, enhancing user experience.



Step 1: Fetching Data from API

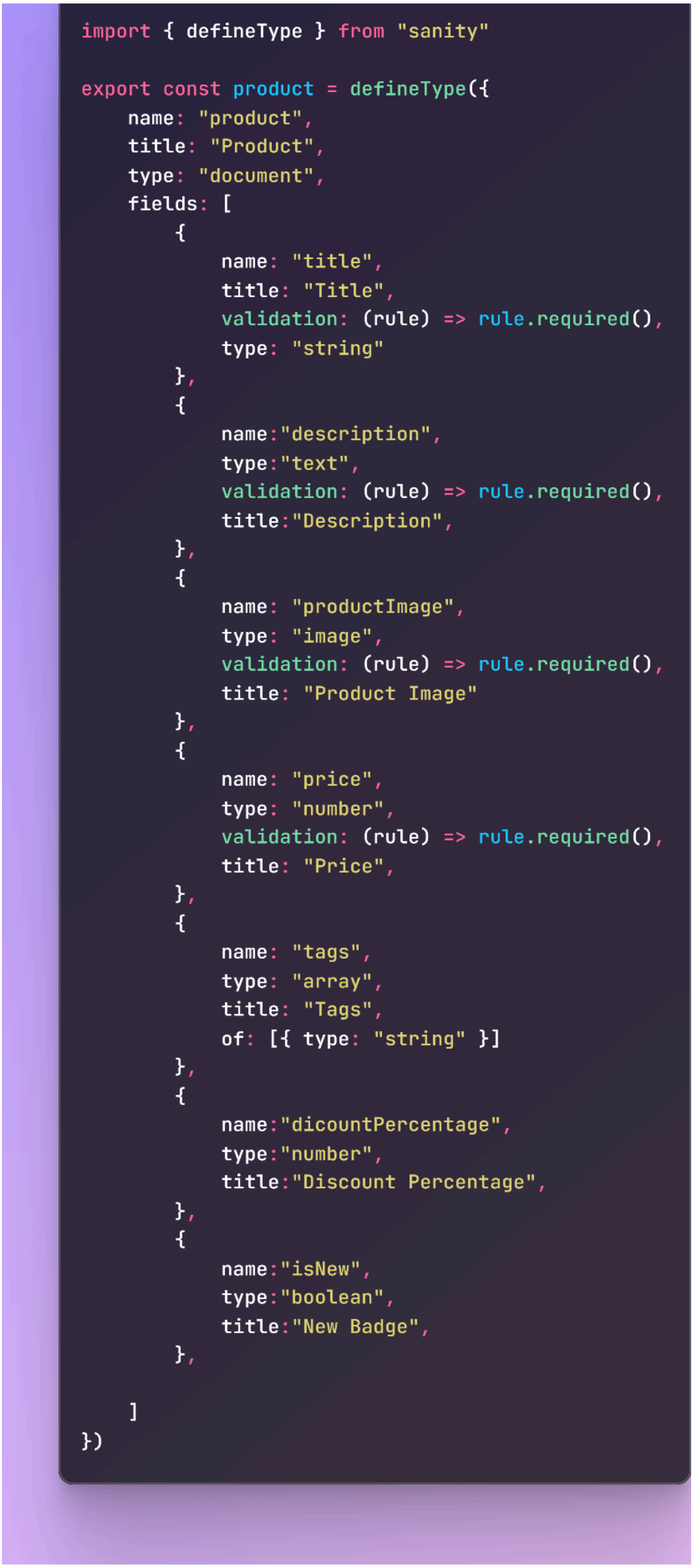
The API provided the following endpoints:

- **Products Endpoint**: Includes details such as titles, prices, descriptions, categories, inventory, and images.
    - **Base API URL:** https://template6-six.vercel.app/api/products



```javascript
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);
  }
```

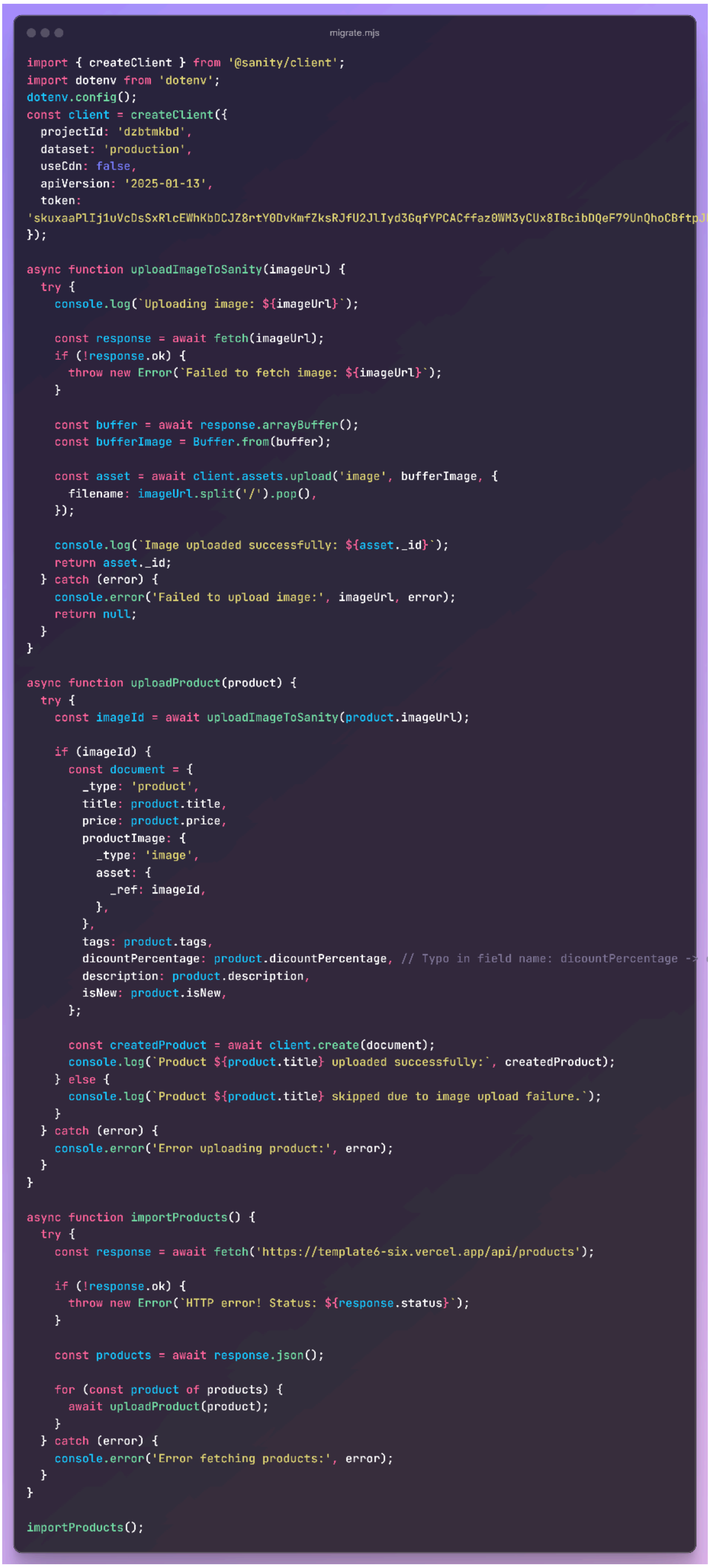
- Category schema: Structured for easy classification and filtering.



```
import { defineType } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        {
            name: "title",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        },
        {
            name:"description",
            type:"text",
            validation: (rule) => rule.required(),
            title:"Description",
        },
        {
            name: "productImage",
            type: "image",
            validation: (rule) => rule.required(),
            title: "Product Image"
        },
        {
            name: "price",
            type: "number",
            validation: (rule) => rule.required(),
            title: "Price",
        },
        {
            name: "tags",
            type: "array",
            title: "Tags",
            of: [{ type: "string" }]
        },
        {
            name:"dicountPercentage",
            type:"number",
            title:"Discount Percentage",
        },
        {
            name:"isNew",
            type:"boolean",
            title:"New Badge",
        },

    ]
})
```

Migration Script

A migration script was developed to automate data insertion into Sanity CMS, ensuring smooth integration.

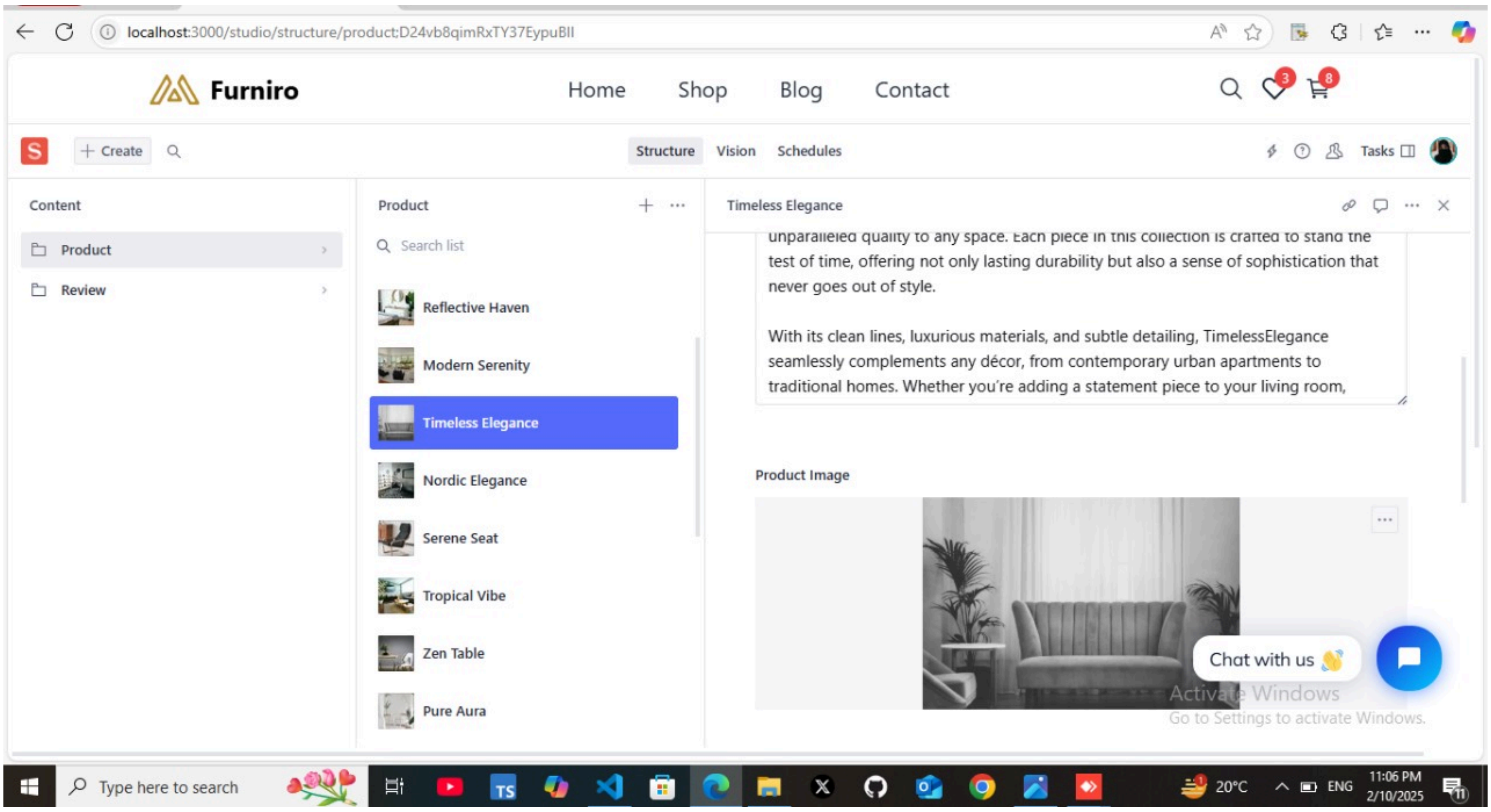


migrate.mjs

```
import { createClient } from '@sanity/client';
import dotenv from 'dotenv';
dotenv.config();
const client = createClient({
  projectId: 'dzbtmkbd',
  dataset: 'production',
  useCdn: false,
  apiVersion: '2025-01-13',
  token:
'skuxaaPlIj1uVcDsSxRlcEWhKbDCJZ8rtY0DvKmfZksRJfU2JlIyd3GqfYPCACffaz0WM3yCUx8IBcibDQeF79UnQhoCBftpJ
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    const response = await fetch(imageUrl);
    if (!response.ok) {
      throw new Error(`Failed to fetch image: ${imageUrl}`);
    }

    const buffer = await response.arrayBuffer();
    const bufferImage = Buffer.from(buffer);

    const asset = await client.assets.upload('image', bufferImage, {
      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 uploadProduct(product) {
  try {
    const imageId = await uploadImageToSanity(product.imageUrl);

    if (imageId) {
      const document = {
        _type: 'product',
        title: product.title,
        price: product.price,
        productImage: {
          _type: 'image',
          asset: {
            _ref: imageId,
          },
        },
        tags: product.tags,
        dicountPercentage: product.dicountPercentage, // Typo in field name: dicountPercentage ->
        description: product.description,
        isNew: product.isNew,
      };

      const createdProduct = await client.create(document);
      console.log(`Product ${product.title} uploaded successfully:`, createdProduct);
    } else {
      console.log(`Product ${product.title} skipped due to image upload failure.`);
    }
  } catch (error) {
    console.error('Error uploading product:', error);
  }
}

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);
  }
}

importProducts();
```

Sanity Dashboard Screenshot

The updated product and category data is now visible in the Sanity dashboard, enabling real-time management.



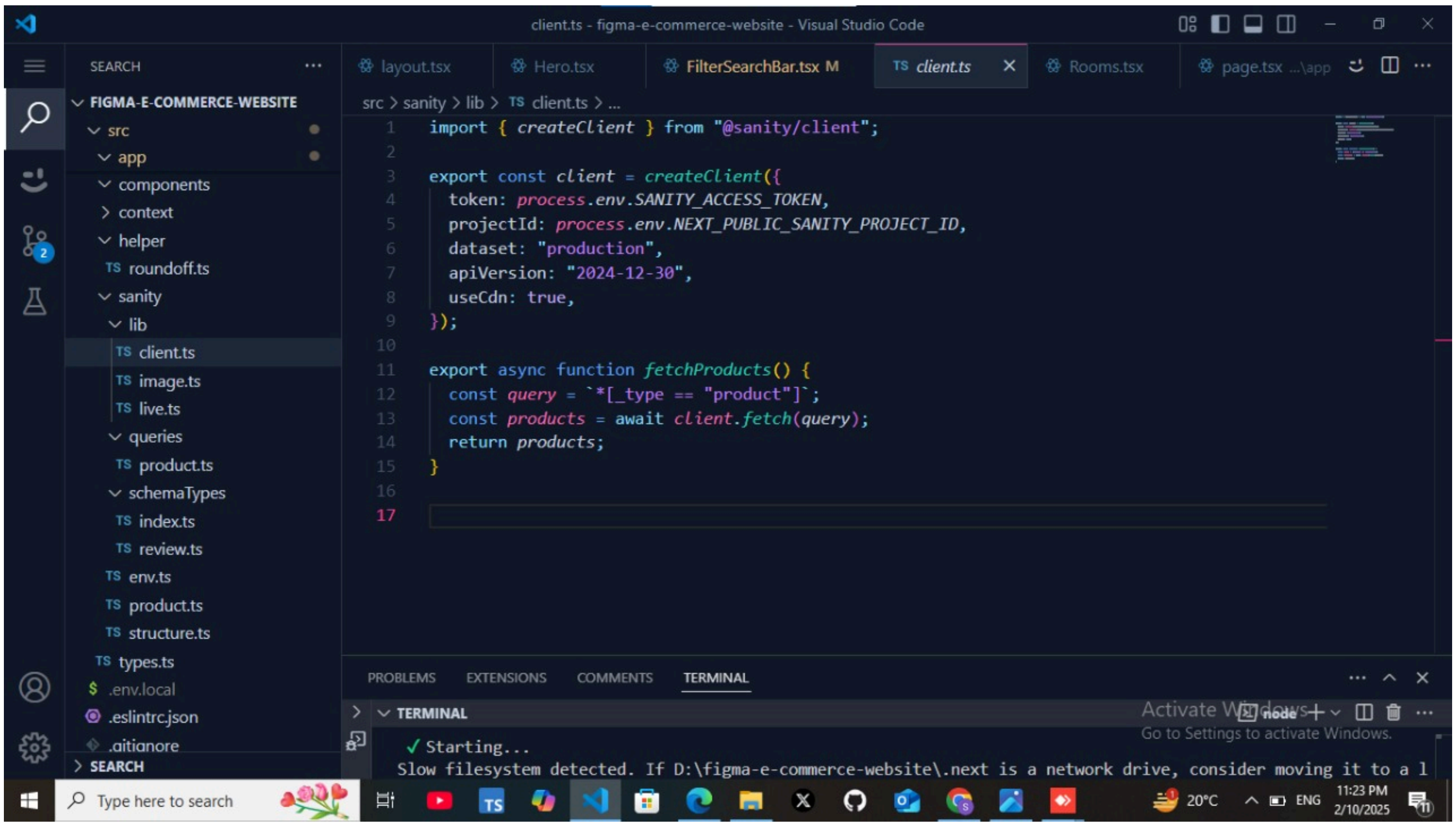
unparalleled quality to any space. Each piece in this collection is crafted to stand the test of time, offering not only lasting durability but also a sense of sophistication that never goes out of style.

With its clean lines, luxurious materials, and subtle detailing, TimelessElegance seamlessly complements any décor, from contemporary urban apartments to traditional homes. Whether you're adding a statement piece to your living room,

Step 3: Rendering Data on Frontend

Using Next.js, the stored data from Sanity CMS was dynamically fetched and displayed on the Furniro frontend. This approach ensures seamless integration, improving the shopping experience.

Code Example for Fetching Data in Frontend:

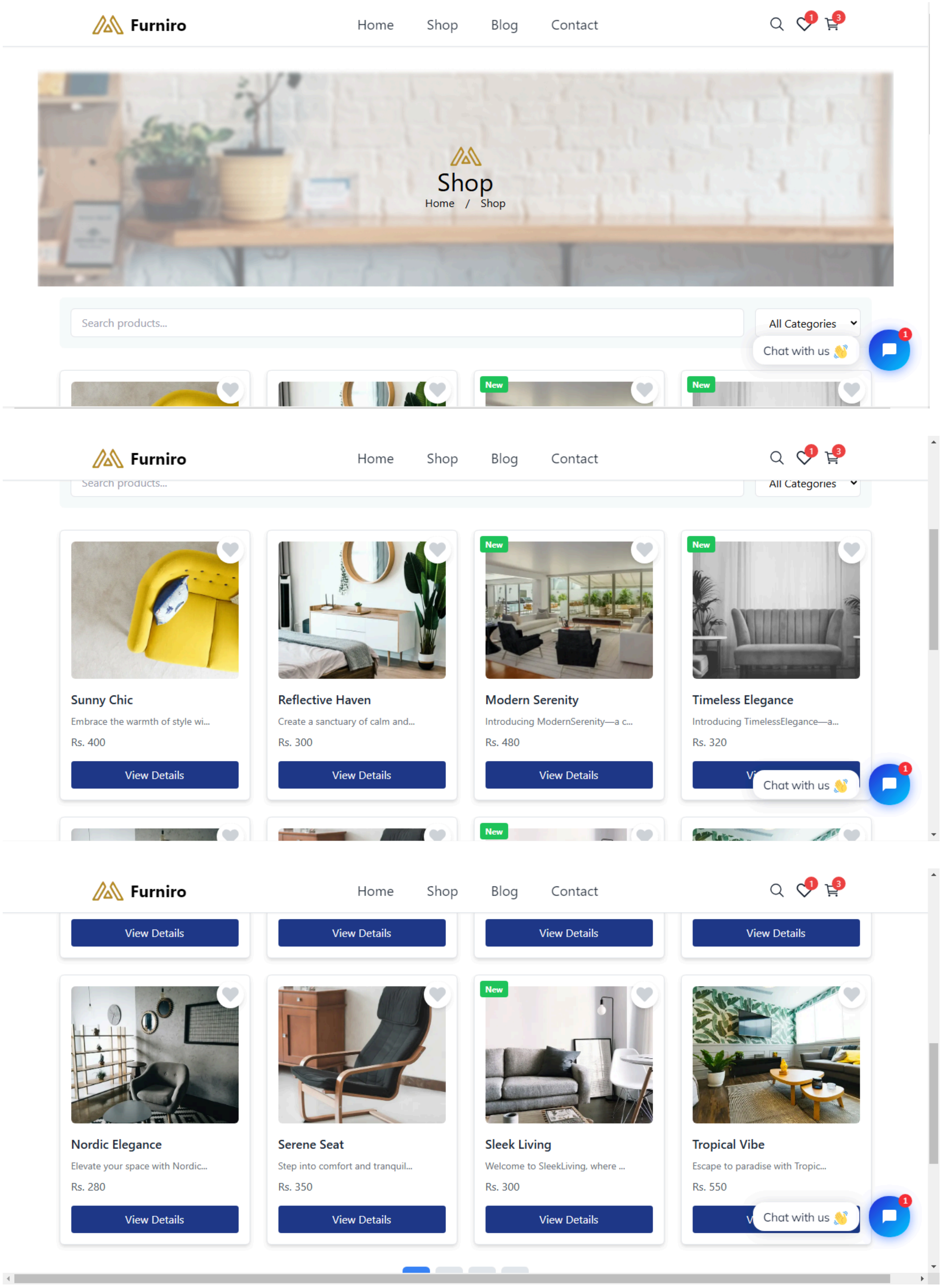


```ts
import { createClient } from "@sanity/client";

export const client = createClient({
  token: process.env.SANITY_ACCESS_TOKEN,
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: "production",
  apiVersion: "2024-12-30",
  useCdn: true,
});

export async function fetchProducts() {
  const query = `*[_type == "product"]`;
  const products = await client.fetch(query);
  return products;
}

```

```
 ✓ Starting...
Slow filesystem detected. If D:\figma-e-commerce-website\.next is a network drive, consider moving it to a l
```

Product Display

A screenshot of the dynamically rendered product listing page demonstrates how products are displayed on the Furniro website.



Sunny Chic
Embrace the warmth of style wi...
Rs. 400

Reflective Haven
Create a sanctuary of calm and...
Rs. 300

Modern Serenity
Introducing ModernSerenity—a c...
Rs. 480

Timeless Elegance
Introducing TimelessElegance—a...
Rs. 320

Nordic Elegance
Elevate your space with Nordic...
Rs. 280

Serene Seat
Step into comfort and tranquil...
Rs. 350

Sleek Living
Welcome to SleekLiving, where ...
Rs. 300

Tropical Vibe
Escape to paradise with Tropic...
Rs. 550

Conclusion

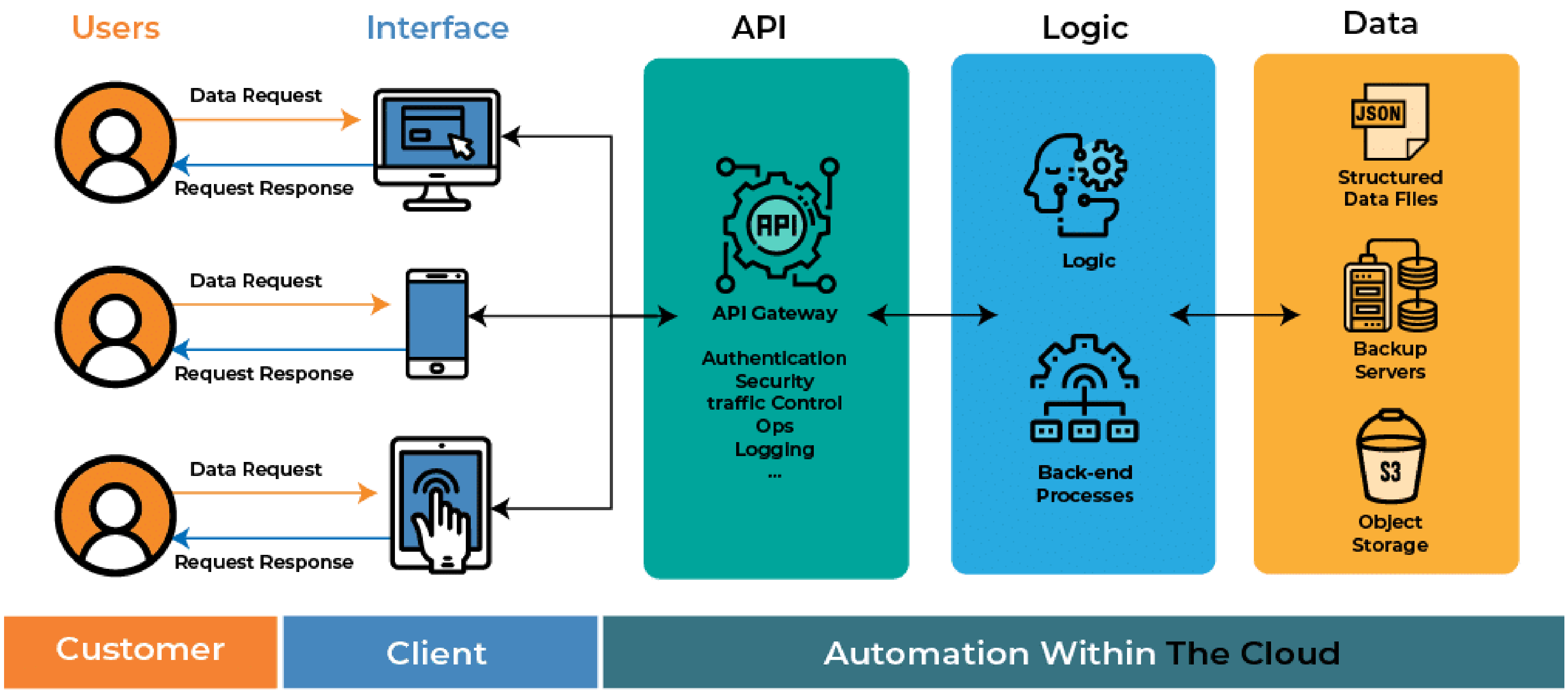
The API integration for Furniro has been successfully completed, ensuring seamless synchronization between Sanity CMS and the frontend. This integration plays a crucial role in enhancing the overall efficiency of Furniro's digital infrastructure. By establishing a robust connection between the backend and the frontend, the data flow remains uninterrupted, ensuring that all product listings and category updates reflect in real time.

The process involved multiple key steps, starting with fetching data from the external API, processing and structuring it for optimized storage in Sanity CMS, and dynamically rendering the content on the frontend using Next.js. Each of these steps was executed with precision to ensure maximum efficiency, accuracy, and performance for the platform.

With this integration, Furniro's data management capabilities have significantly improved, allowing for better organization and accessibility of product information. Real-time updates ensure that administrators can seamlessly modify, add, or remove products and categories without facing synchronization issues. Furthermore, users can experience an enhanced shopping journey, where the latest products, prices, and details are instantly available, contributing to a smooth and engaging user experience.

The use of Next.js for frontend rendering optimizes performance, making pages load faster and improving the website's responsiveness. By leveraging Sanity CMS, Furniro now has a flexible and scalable system that allows for easy content management while maintaining the highest level of data integrity.

This successful integration marks a major milestone in Furniro's development, ensuring that it remains competitive in the e-commerce space by offering a modern, efficient, and user-friendly interface that is constantly updated with relevant information.



Documentation Author: Sana Ishaq
Task Given By: Sir Ameen Alam
Class Teacher: Sir Hamza Syed

Slot: (Friday 9am to 12pm)