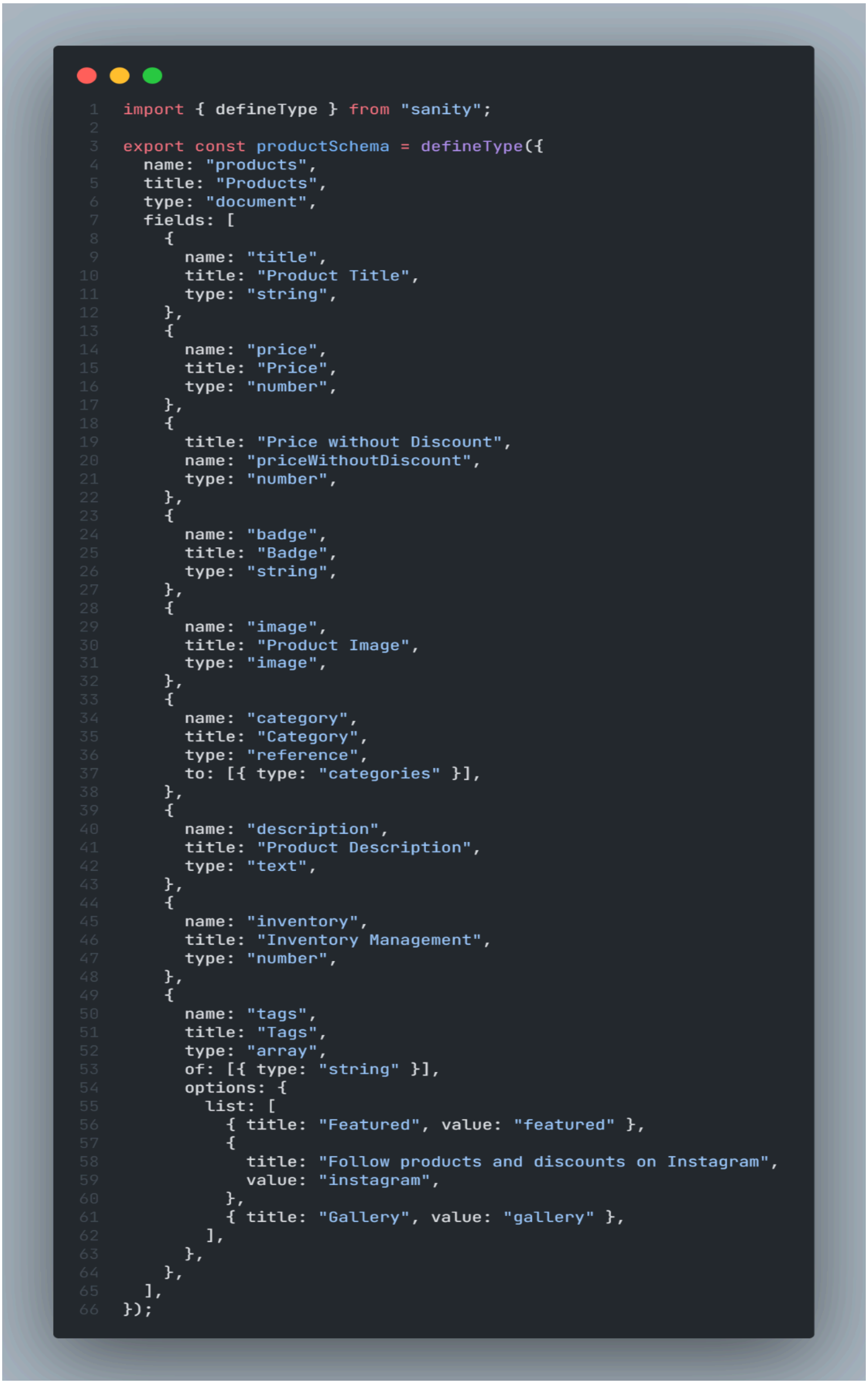
Day 3 - API Integration Report General E-commerce Marketplace

Our project integrates a fully functional e-commerce marketplace for furniture. This document outlines the API integration process, schema adjustments, migration steps, and frontend data display. It includes technical details, tools used, and relevant code snippets.

1: API Used: (e.g., RESTful API, Sanity CMS GraphQL)
2: Purpose: Explain what the API achieves (e.g., fetching product data).
Steps:

- Configure API base URL.
- Add authentication tokens or keys.

We adjusted the product schema in Sanity CMS to include additional fields like `badges`, `tags`, and `inventory management`. Below is the code snippet



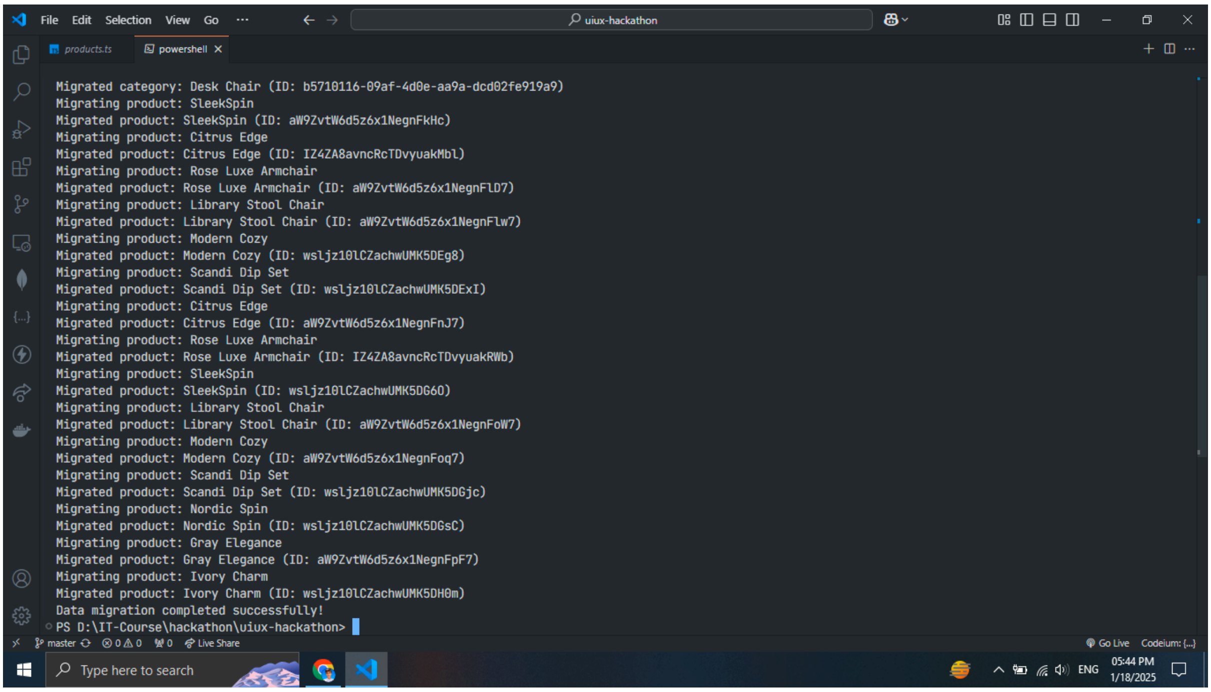
```
import { defineType } from "sanity";

export const productSchema = defineType({
  name: "products",
  title: "Products",
  type: "document",
  fields: [
    {
      name: "title",
      title: "Product Title",
      type: "string",
    },
    {
      name: "price",
      title: "Price",
      type: "number",
    },
    {
      title: "Price without Discount",
      name: "priceWithoutDiscount",
      type: "number",
    },
    {
      name: "badge",
      title: "Badge",
      type: "string",
    },
    {
      name: "image",
      title: "Product Image",
      type: "image",
    },
    {
      name: "category",
      title: "Category",
      type: "reference",
      to: [{ type: "categories" }],
    },
    {
      name: "description",
      title: "Product Description",
      type: "text",
    },
    {
      name: "inventory",
      title: "Inventory Management",
      type: "number",
    },
    {
      name: "tags",
      title: "Tags",
      type: "array",
      of: [{ type: "string" }],
      options: {
        list: [
          { title: "Featured", value: "featured" },
          {
            title: "Follow products and discounts on Instagram",
            value: "instagram",
          },
          { title: "Gallery", value: "gallery" },
        ],
      },
    },
  ],
});
```

Used Tools like Postman.

Steps to migrate data

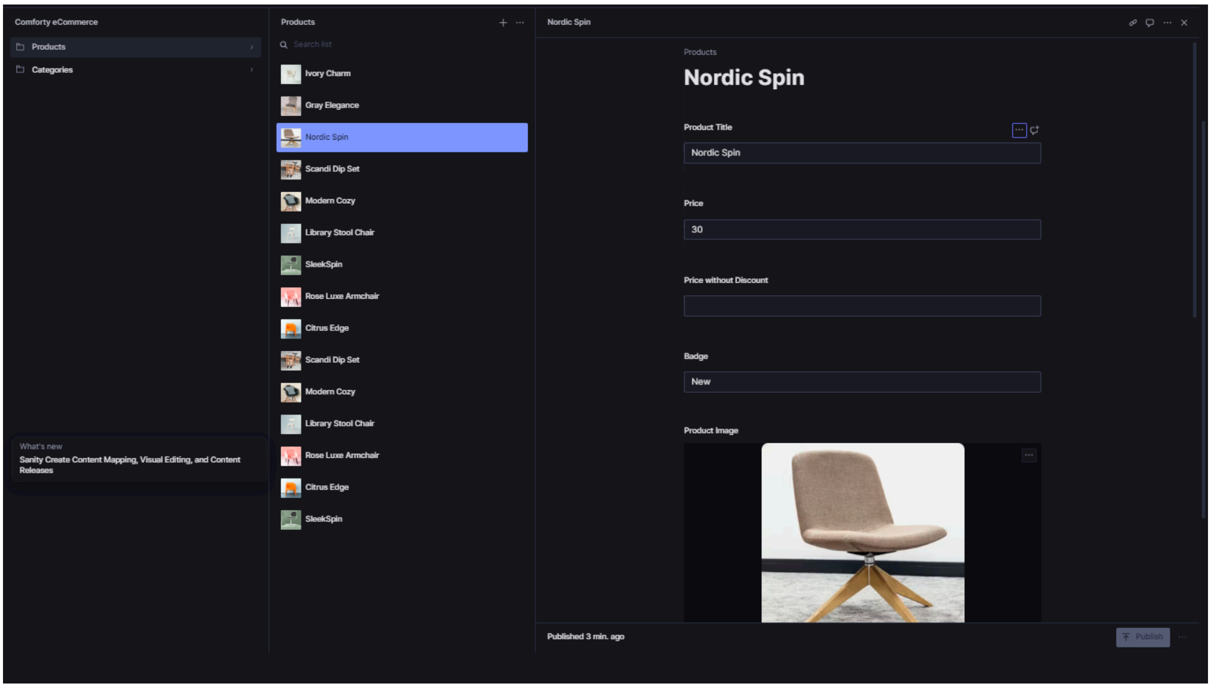
- Define BASE_URL=[your base url]
- Get the authentication token from sanity API token
- Create .env and past the key

API Calls



```
Migrated category: Desk Chair (ID: b5710116-09af-4d0e-aa9a-dcd02fe919a9)
Migrating product: SleekSpin
Migrated product: SleekSpin (ID: aW9ZvtW6d5z6x1NegnFkHc)
Migrating product: Citrus Edge
Migrated product: Citrus Edge (ID: IZ4ZA8avncRcTDvyuakMbl)
Migrating product: Rose Luxe Armchair
Migrated product: Rose Luxe Armchair (ID: aW9ZvtW6d5z6x1NegnFlD7)
Migrating product: Library Stool Chair
Migrated product: Library Stool Chair (ID: aW9ZvtW6d5z6x1NegnFlw7)
Migrating product: Modern Cozy
Migrated product: Modern Cozy (ID: wsljz10lCZachwUMK5DEg8)
Migrating product: Scandi Dip Set
Migrated product: Scandi Dip Set (ID: wsljz10lCZachwUMK5DExI)
Migrating product: Citrus Edge
Migrated product: Citrus Edge (ID: aW9ZvtW6d5z6x1NegnFnJ7)
Migrating product: Rose Luxe Armchair
Migrated product: Rose Luxe Armchair (ID: IZ4ZA8avncRcTDvyuakRWb)
Migrating product: SleekSpin
Migrated product: SleekSpin (ID: wsljz10lCZachwUMK5DG6O)
Migrating product: Library Stool Chair
Migrated product: Library Stool Chair (ID: aW9ZvtW6d5z6x1NegnFoW7)
Migrating product: Modern Cozy
Migrated product: Modern Cozy (ID: aW9ZvtW6d5z6x1NegnFoq7)
Migrating product: Scandi Dip Set
Migrated product: Scandi Dip Set (ID: wsljz10lCZachwUMK5DGjc)
Migrating product: Nordic Spin
Migrated product: Nordic Spin (ID: wsljz10lCZachwUMK5DGsC)
Migrating product: Gray Elegance
Migrated product: Gray Elegance (ID: aW9ZvtW6d5z6x1NegnFpF7)
Migrating product: Ivory Charm
Migrated product: Ivory Charm (ID: wsljz10lCZachwUMK5DH0m)
Data migration completed successfully!
PS D:\IT-Course\hackathon\uiux-hackathon>
```

Populated Sanity CMS fields.



What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases

Products

Nordic Spin

Product Title: Nordic Spin

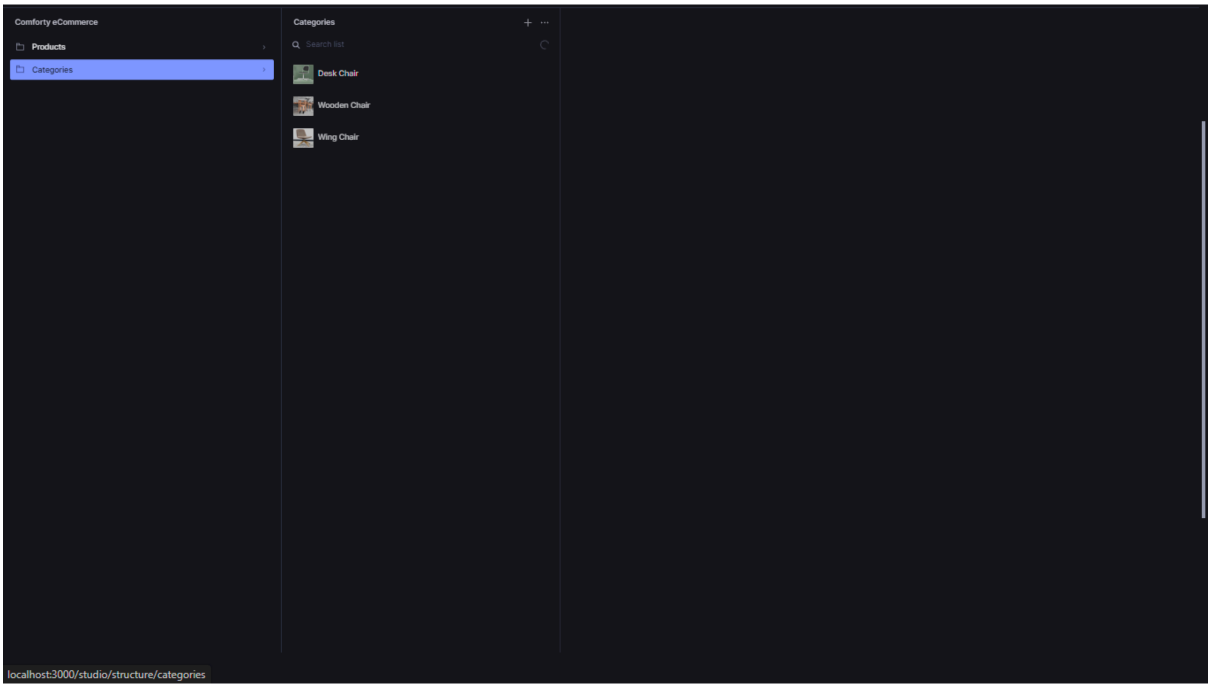
Price: 30

Price without Discount:

Badge: New

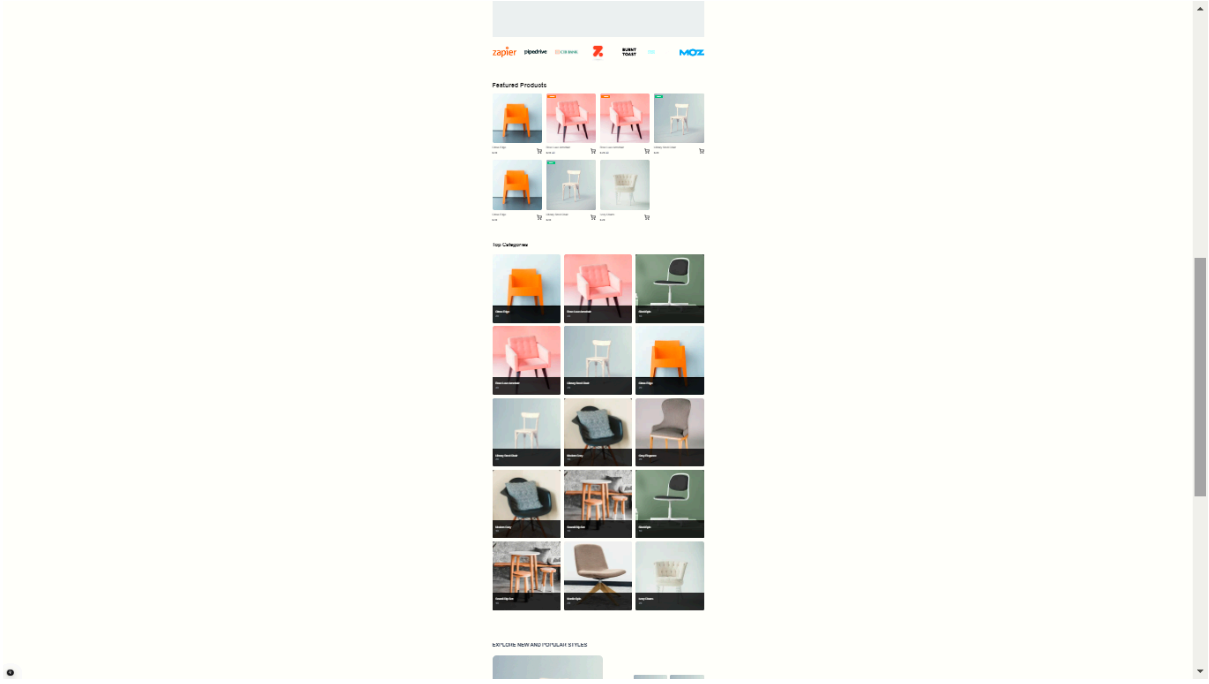
Product Image

Published 3 min. ago

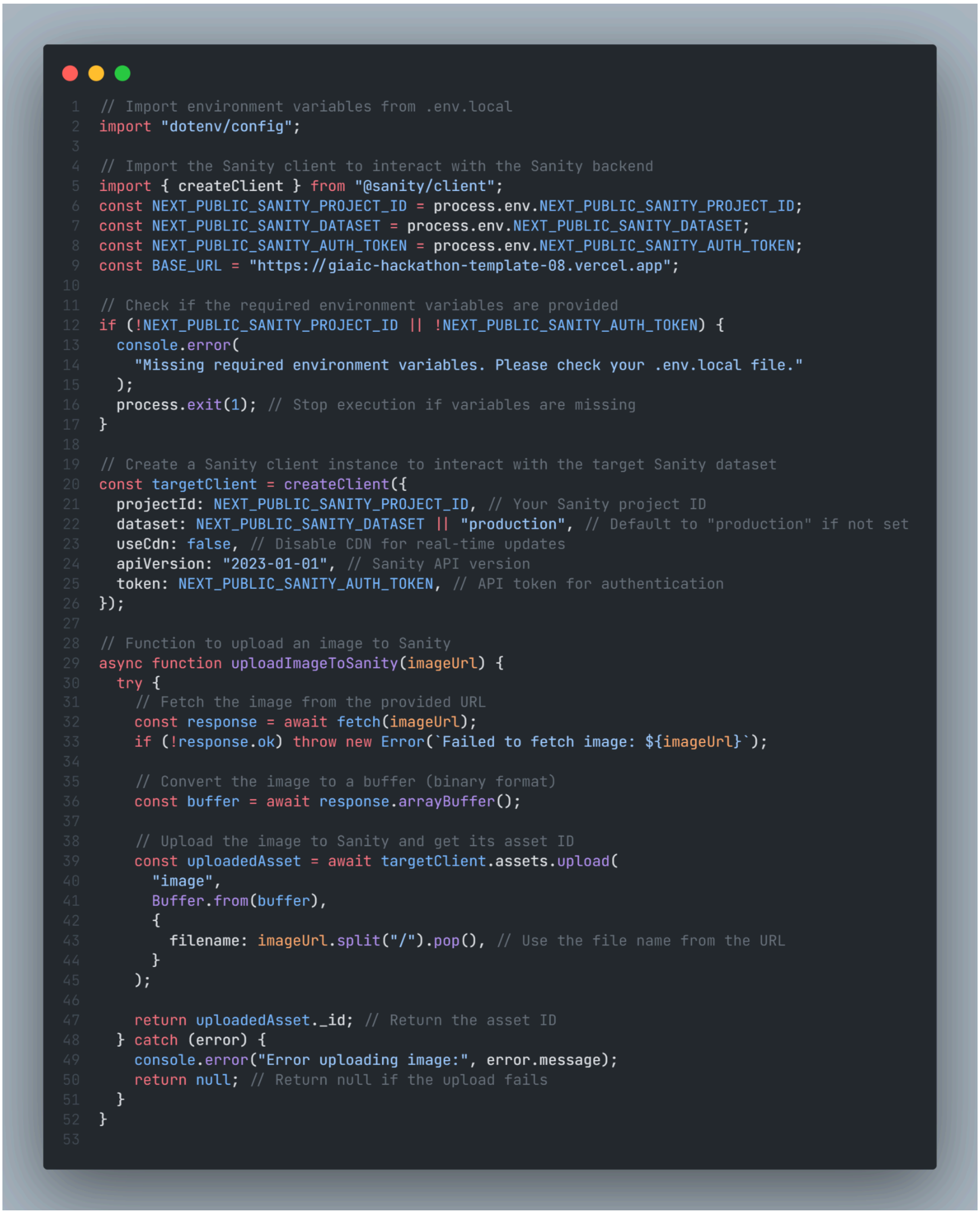


localhost:3000/studio/structure/categories

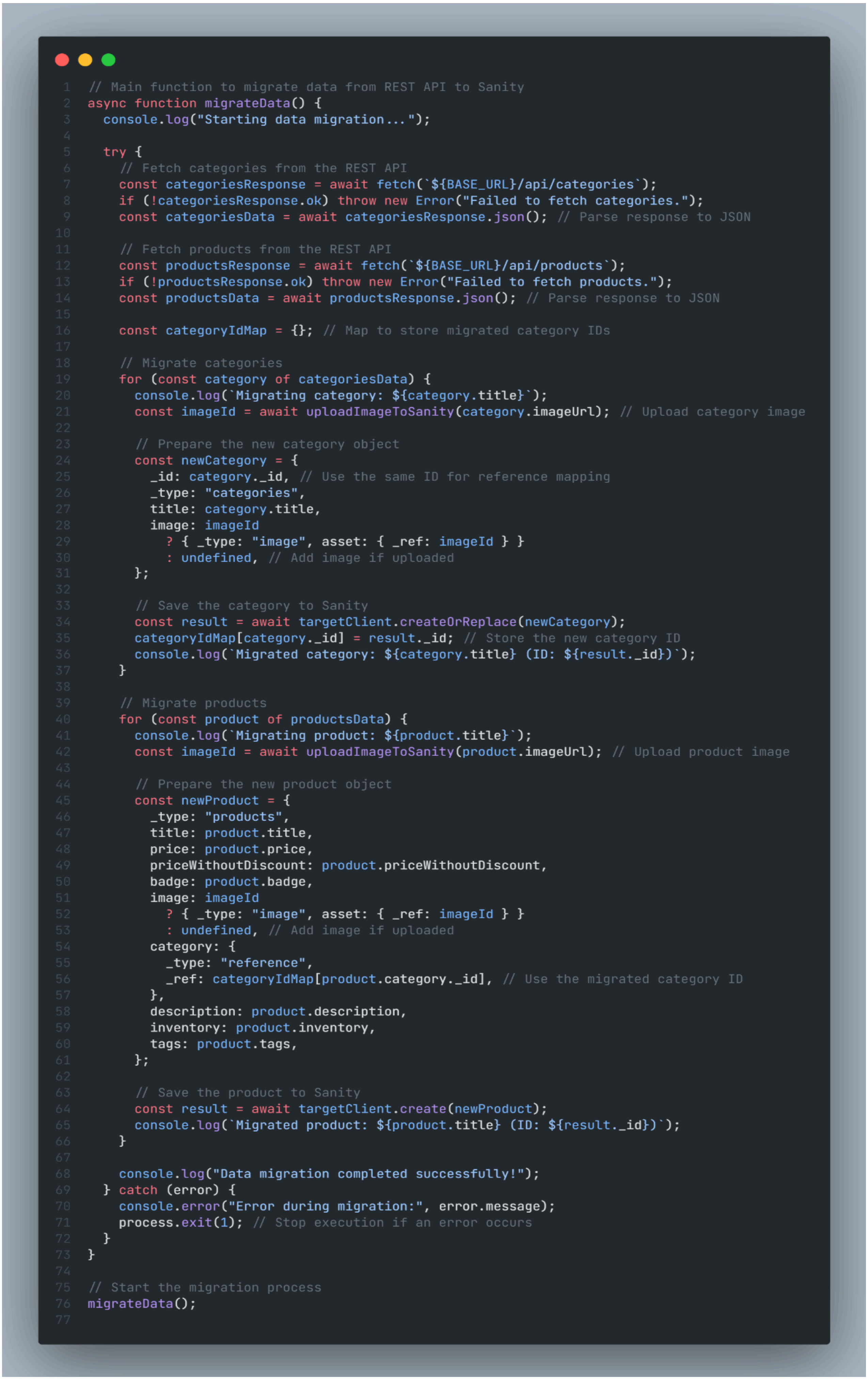
Data is successfully displayed in the frontend.



Code snippets for API integration and migration scripts.



```javascript
// Import environment variables from .env.local
import "dotenv/config";

// Import the Sanity client to interact with the Sanity backend
import { createClient } from "@sanity/client";
const NEXT_PUBLIC_SANITY_PROJECT_ID = process.env.NEXT_PUBLIC_SANITY_PROJECT_ID;
const NEXT_PUBLIC_SANITY_DATASET = process.env.NEXT_PUBLIC_SANITY_DATASET;
const NEXT_PUBLIC_SANITY_AUTH_TOKEN = process.env.NEXT_PUBLIC_SANITY_AUTH_TOKEN;
const BASE_URL = "https://giaic-hackathon-template-08.vercel.app";

// Check if the required environment variables are provided
if (!NEXT_PUBLIC_SANITY_PROJECT_ID || !NEXT_PUBLIC_SANITY_AUTH_TOKEN) {
  console.error(
    "Missing required environment variables. Please check your .env.local file."
  );
  process.exit(1); // Stop execution if variables are missing
}

// Create a Sanity client instance to interact with the target Sanity dataset
const targetClient = createClient({
  projectId: NEXT_PUBLIC_SANITY_PROJECT_ID, // Your Sanity project ID
  dataset: NEXT_PUBLIC_SANITY_DATASET || "production", // Default to "production" if not set
  useCdn: false, // Disable CDN for real-time updates
  apiVersion: "2023-01-01", // Sanity API version
  token: NEXT_PUBLIC_SANITY_AUTH_TOKEN, // API token for authentication
});

// Function to upload an image to Sanity
async function uploadImageToSanity(imageUrl) {
  try {
    // Fetch the image from the provided URL
    const response = await fetch(imageUrl);
    if (!response.ok) throw new Error(`Failed to fetch image: ${imageUrl}`);

    // Convert the image to a buffer (binary format)
    const buffer = await response.arrayBuffer();

    // Upload the image to Sanity and get its asset ID
    const uploadedAsset = await targetClient.assets.upload(
      "image",
      Buffer.from(buffer),
      {
        filename: imageUrl.split("/").pop(), // Use the file name from the URL
      }
    );

    return uploadedAsset._id; // Return the asset ID
  } catch (error) {
    console.error("Error uploading image:", error.message);
    return null; // Return null if the upload fails
  }
}
```



```javascript
// Main function to migrate data from REST API to Sanity
async function migrateData() {
  console.log("Starting data migration...");

  try {
    // Fetch categories from the REST API
    const categoriesResponse = await fetch(`${BASE_URL}/api/categories`);
    if (!categoriesResponse.ok) throw new Error("Failed to fetch categories.");
    const categoriesData = await categoriesResponse.json(); // Parse response to JSON

    // Fetch products from the REST API
    const productsResponse = await fetch(`${BASE_URL}/api/products`);
    if (!productsResponse.ok) throw new Error("Failed to fetch products.");
    const productsData = await productsResponse.json(); // Parse response to JSON

    const categoryIdMap = {}; // Map to store migrated category IDs

    // Migrate categories
    for (const category of categoriesData) {
      console.log(`Migrating category: ${category.title}`);
      const imageId = await uploadImageToSanity(category.imageUrl); // Upload category image

      // Prepare the new category object
      const newCategory = {
        _id: category._id, // Use the same ID for reference mapping
        _type: "categories",
        title: category.title,
        image: imageId
          ? { _type: "image", asset: { _ref: imageId } }
          : undefined, // Add image if uploaded
      };

      // Save the category to Sanity
      const result = await targetClient.createOrReplace(newCategory);
      categoryIdMap[category._id] = result._id; // Store the new category ID
      console.log(`Migrated category: ${category.title} (ID: ${result._id})`);
    }

    // Migrate products
    for (const product of productsData) {
      console.log(`Migrating product: ${product.title}`);
      const imageId = await uploadImageToSanity(product.imageUrl); // Upload product image

      // Prepare the new product object
      const newProduct = {
        _type: "products",
        title: product.title,
        price: product.price,
        priceWithoutDiscount: product.priceWithoutDiscount,
        badge: product.badge,
        image: imageId
          ? { _type: "image", asset: { _ref: imageId } }
          : undefined, // Add image if uploaded
        category: {
          _type: "reference",
          _ref: categoryIdMap[product.category._id], // Use the migrated category ID
        },
        description: product.description,
        inventory: product.inventory,
        tags: product.tags,
      };

      // Save the product to Sanity
      const result = await targetClient.create(newProduct);
      console.log(`Migrated product: ${product.title} (ID: ${result._id})`);
    }

    console.log("Data migration completed successfully!");
  } catch (error) {
    console.error("Error during migration:", error.message);
    process.exit(1); // Stop execution if an error occurs
  }
}

// Start the migration process
migrateData();
```