

Chairs E-Commerce Website

Day 3: API Integration and Data Migration

Objective:

The primary goal of Day 3 is to integrate APIs and migrate data into Sanity CMS to establish a functional marketplace backend. This process involves understanding API documentation, validating schemas, and importing or migrating data. The end result is a seamless integration of backend data into the frontend application using Next.js.

API Integration and Migration Process

Step 1: Understand the Provided API

- **Reviewed API Documentation:**
 - Accessed the assigned API documentation from the provided template.
 - Identified key endpoints like:
 - /products: For fetching product listings.
 - /categories: For fetching product categories.
 - Tested the API using Postman to ensure it returns valid and structured data.

Step 2: Validate and Adjust Sanity CMS Schema

- Compared the provided API fields with the existing Sanity CMS schema created in Day 2.
- Adjusted the schema fields to match the API data structure. For example:
 - **API Field:** product_title was mapped to **Schema Field:** name.
 - Modified field types where necessary to ensure compatibility (e.g., string, number, array).
- Added relationships in the schema to support data linking (e.g., linking products to categories).

Step 3: Data Migration

- **Migration Methods Used:**
 1. **Script-Based Migration:**
 - Wrote scripts in JavaScript to fetch data from the API.
 - Transformed the data structure to match the Sanity CMS schema.

- Uploaded the data using Sanity's API tools.

2. Manual Import:

- Exported API data in JSON format.
- Used Sanity's import tools to manually upload smaller datasets.
- **Validation:** Ensured all data was properly imported by verifying in the Sanity CMS dashboard.

Step 4: API Integration in Next.js

- **Utility Functions:**
 - Created reusable utility functions to fetch data from the API.
 - Implemented error handling for edge cases (e.g., empty or invalid responses).
- **Frontend Integration:**
 - Rendered fetched data in the following components:
 - **Product List:** Displayed all products with details such as name, price, and description.
 - **Categories:** Created a dropdown menu for filtering products by category.
- **Testing:**
 - Used browser developer tools and Postman to test API responses.
 - Logged responses in the console to verify data consistency.

Schema Adjustments

Field Name	API Field	Adjusted Field	Data Type
product_title	name	title	string
product_price	price	cost	number
category_id	categories	linkedCategory	array
image_url	image	url	string

Screenshots



Ubaid Raza

beginners



4 days left in trial



Getting started

Overview

Members

Studios

Datasets

Access

Activity

Usage

Plan

API

Settings

Webhooks

CORS origins

Tokens

Name

Examples: "Employee import", "Website preview" or "PDF generator".

day3

Permissions

Choose the access privileges for the token.

Contributor

☐ Read and write access to draft content within all datasets, with no access to project settings. (Tokens: read+write drafts)

Deploy Studio (Token only)

☐ Access to deploy Sanity Studio and GraphQL APIs to our hosted service.

Developer

☒ Read and write access to all datasets, with access to project settings for developers. (Tokens: read+write)

Editor

☐ Read and write access to all datasets, with limited access to project settings. (Tokens: read+write)

Viewer

☐ Read access to all datasets, with limited access to project settings. (Tokens: read-only)

Save

Cancel

What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases

Activate Windows

Go to Settings to activate Windows.

package.json > {} dependencies

```
1 {
2   "name": "practice-app",
3   "version": "0.1.0",
4   "private": true,
```

PROBLEMS OUTPUT PORTS TERMINAL DEBUG CONSOLE

Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SIDDIQUI TAJ 2024\Desktop\practice-app>npm install @sanity/client axios dotenv

added 12 packages, removed 7 packages, changed 6 packages, and audited 1334 packages in 5m

245 packages are looking for funding
run `npm fund` for details

3 vulnerabilities (2 moderate, 1 high)

To address all issues, run:
npm audit fix

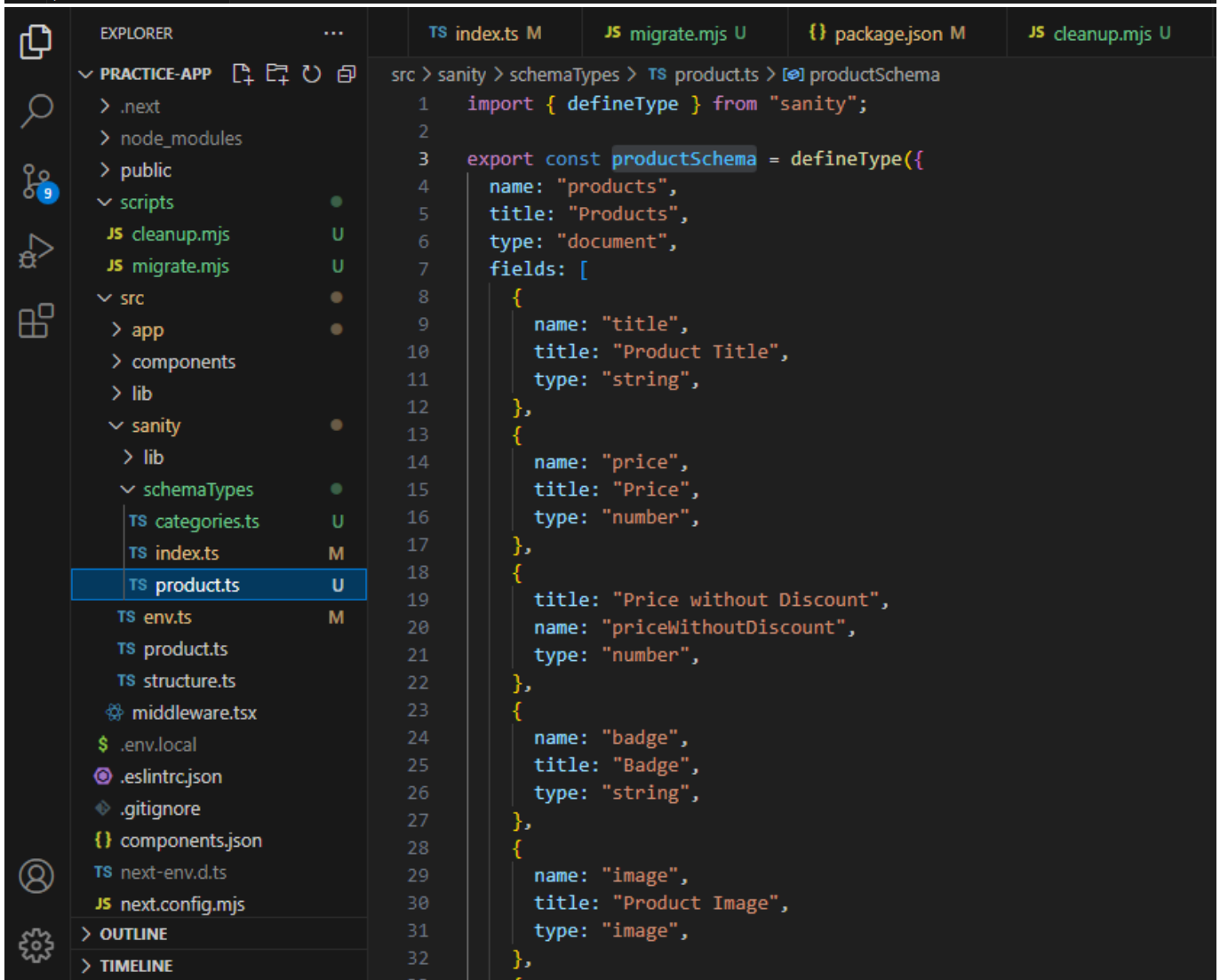
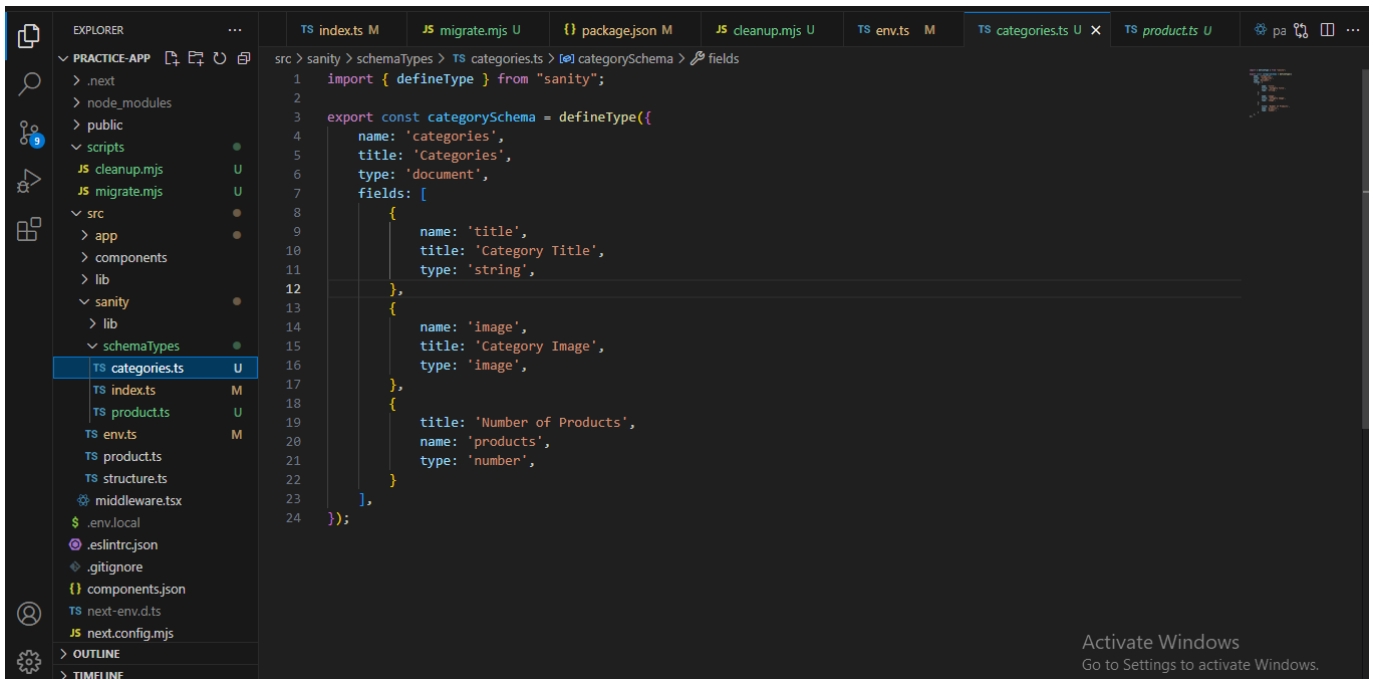
Run `npm audit` for details.

C:\Users\SIDDIQUI TAJ 2024\Desktop\practice-app>

node
cmd

Activate Windows
Go to Settings to activate Windows.

Ln 18, Col 33 Spaces: 2 UTF-8 LF JSON Prettier



1. API Calls:

The screenshot shows a VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'PRACTICE-APP' with various files and folders. The terminal is running a command to execute a migration script, and the output shows the migration of categories and products.

```
$.env.local
1 NEXT_PUBLIC_SANITY_PROJECT_ID="1r2n996y"
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 NEXT_PUBLIC_SANITY_AUTH_TOKEN="skYU5ALX920FTa40C7s02qDtxWkiaiLsKMJ1He5zZZbPtOqSuApMKTEvfopSA0CZda8f1X3ZshyPW9D4nWgFZzc
4 BASE_URL="https://giaic-hackathon-template-08.vercel.app"
5 NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY=pk_test_Z29sZGVuLXBpcGVmaXNoLTczLmNsZXJrLmFjY291bnRzLmRldiQ
6 CLERK_SECRET_KEY=sk_test_eVoDDKTA6mgrNQXhmdJoi15160EOsEppqBzoj8bVzCZ9
7
8
```

```
C:\Users\SIDDIQUI TAJ 2024\Desktop\practice-app>npm run migrate

> practice-app@0.1.0 migrate
> node scripts/migrate.mjs

Starting data migration...
Migrating category: Wing Chair
Migrated category: Wing Chair (ID: 26fd7176-3c4d-40fc-a73a-3b85a9b5e15f)
Migrating category: Wooden Chair
Migrated category: Wooden Chair (ID: 407a8583-6203-4f61-becf-8e8b4c5461b6)
Migrating category: Desk Chair
Migrated category: Desk Chair (ID: b5710116-09af-4d0e-aa9a-dcd02fe919a9)
Migrating product: SleekSpin
Migrated product: SleekSpin (ID: Pneo15i0MtwqZxDLMtUV7c)
Migrating product: Citrus Edge
Migrated product: Citrus Edge (ID: b1ZJwZ0NDTDC1m2UymUAFR)
Migrating product: Rose Luxe Armchair
Migrated product: Rose Luxe Armchair (ID: gisOx3lh9g61b1H2CexRa5)
Migrating product: Library Stool Chair
Migrated product: Library Stool Chair (ID: c0mh6EChCrz9DSr1SLtLl0)
Migrating product: Modern Cozy
Migrated product: Modern Cozy (ID: 2CsdtQm68165aFe5M9Hg1d)
Migrating product: Scandi Dip Set
```

Activate Windows
Go to Settings to activate Windows.

2. Frontend Display:



Library Stool Chair

\$20.00 USD

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tincidunt erat enim. Lorem ipsum dolor sit amet, consectetur adipiscing

 Add To Cart

FEATURED PRODUCTS

[View all](#)



Library Stool Chair

\$99



Library Stool Chair

\$99



Library Stool Chair

\$99



Library Stool Chair

\$99



Library Stool Chair

\$99

3. Sanity CMS:

- Successfully migrated data displayed in Sanity CMS dashboard.

S

+ Create

Q

StructureVisionSchedules

Content

ProductsCategories

Categories

Search list

Desk Chair

Wooden Chair

Wing Chair


Desk Chair

Categories

Category Title

Desk Chair

Category Image



S

+ Create

Q

StructureVisionSchedules

Content

ProductsCategories

Products

Search list

Ivory Charm

Gray Elegance

Nordic Spin

Scandi Dip Set

Modern Cozy

Library Stool Chair

SleekSpin

Rose Luxe Armchair

Citrus Edge

Scandi Din Set

Modern Cozy

Category

Wooden Chair

Product Description

Published 4 min. ago

Code Snippets

1. Utility Function for Fetching API Data

```
export const fetchData = async (endpoint: string) => {
  try {
    const response = await fetch(endpoint);
    if (!response.ok) {
      throw new Error(`Failed to fetch data from ${endpoint}`);
    }
    const data = await response.json();
    return data;
  } catch (error) {
    console.error("Error fetching data:", error);
    return null;
  }
};
```

2. Migration Script

```
import sanityClient from "@sanity/client";
import fetch from "node-fetch";

const client = sanityClient({
  projectId: "your_project_id",
  dataset: "production",
  useCdn: false,
  token: "your_sanity_token",
});

const migrateData = async () => {
  const apiEndpoint = "https://your-api-endpoint/products";
  const response = await fetch(apiEndpoint);
  const products = await response.json();

  for (const product of products) {
    await client.create({
      _type: "product",
      name: product.name,
      price: product.price,
      image: product.image_url,
      category: product.category_id,
    });
  }

  console.log("Data migration complete.");
};

migrateData();
```

Activate W
Go to Settings

By following the outlined steps, API integration and data migration were successfully completed. The Sanity CMS is now populated with relevant data, and the frontend reflects accurate product and category information. This exercise provided practical experience with API integration, schema validation, and data migration techniques essential for real-world marketplace development.

(Ensure all screenshots and additional details are added before final submission.)