

Day 3 - API Integration Report – General – Ecommerce

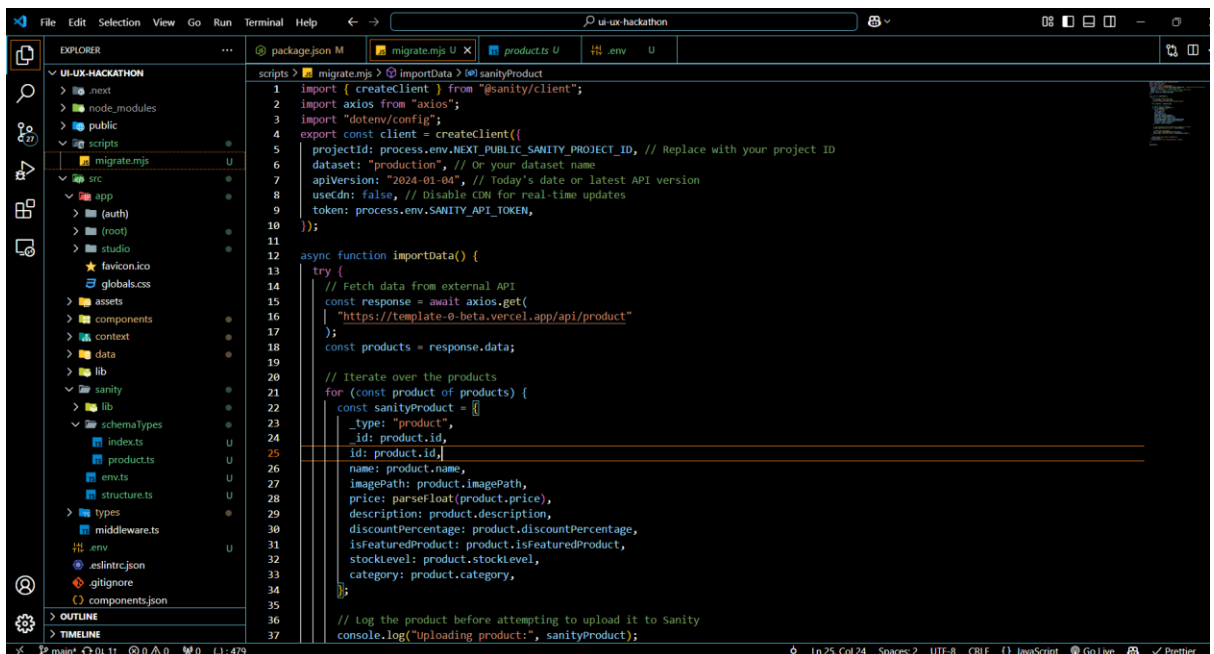
API Integration Process

1. Followed API Documentation

- Reviewed the provided API documentation of the template to understand its structure and endpoints.
- Identified required API endpoints for fetching and managing data.

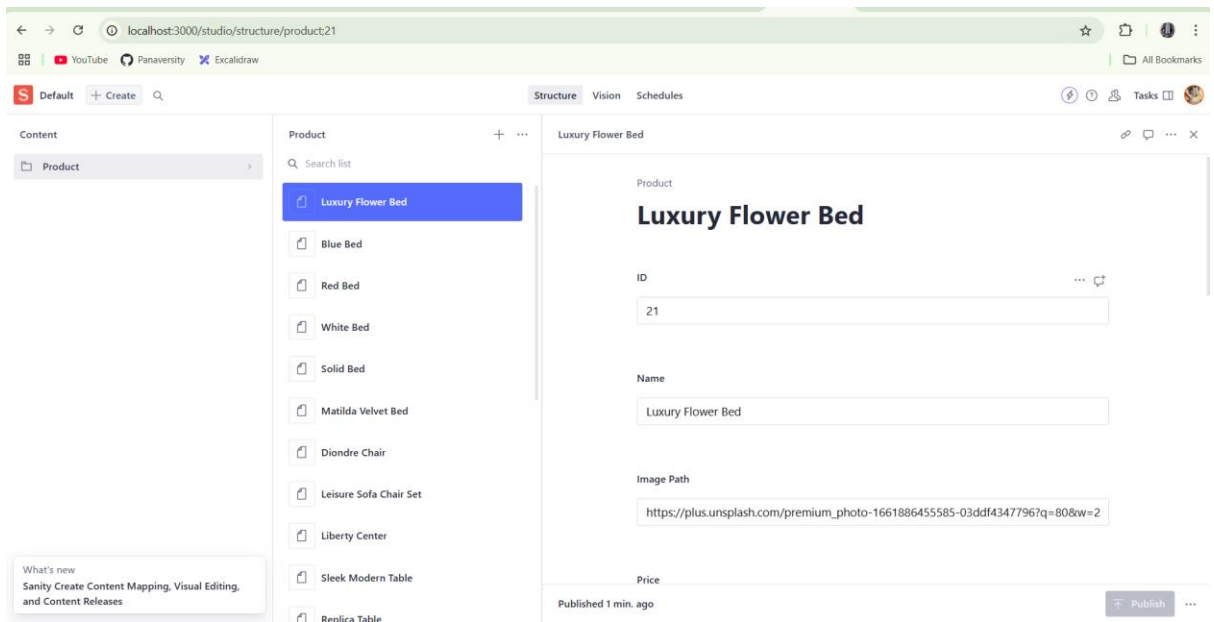
2. Data Migration Script Execution

- Executed a custom script designed to migrate API data into Sanity CMS.
- Verified successful migration by checking the populated Sanity CMS fields.



The screenshot shows a VS Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'UI-UX-HACKATHON' with various folders and files. The code editor displays a JavaScript file named 'migrate.mjs' with the following content:

```
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import 'dotenv/config';
4 export const client = createClient({
5   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID, // Replace with your project ID
6   dataset: 'production', // Or your dataset name
7   apiVersion: '2024-01-04', // Today's date or latest API version
8   useCdn: false, // Disable CDN for real-time updates
9   token: process.env.SANITY_API_TOKEN,
10 });
11
12
13 async function importData() {
14   try {
15     // Fetch data from external API
16     const response = await axios.get(
17       'https://template-0-beta.vercel.app/api/product'
18     );
19     const products = response.data;
20
21     // Iterate over the products
22     for (const product of products) {
23       const sanityProduct = {
24         _type: 'product',
25         id: product.id,
26         name: product.name,
27         imagePath: product.imagePath,
28         price: parseFloat(product.price),
29         description: product.description,
30         discountPercentage: product.discountPercentage,
31         isFeaturedProduct: product.isFeaturedProduct,
32         stockLevel: product.stockLevel,
33         category: product.category,
34       };
35
36       // Log the product before attempting to upload it to Sanity
37       console.log('Uploading product:', sanityProduct);
38     }
39   } catch (error) {
40     console.error('Error importing data:', error);
41   }
42 }
```



Adjustments Made in Product Schemas

- Modifications were made to the default schemas to accommodate the migrated data.
- Ensured compatibility with the structure of the API data.

```
const product = {
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    {
      name: "id",
      title: "ID",
      type: "string",
    },
    {
      name: "name",
      title: "Name",
      type: "string",
    },
    {
      name: "imagePath",
      title: "Image Path",
      type: "url",
    },
    {
      name: "price",
      title: "Price",
```

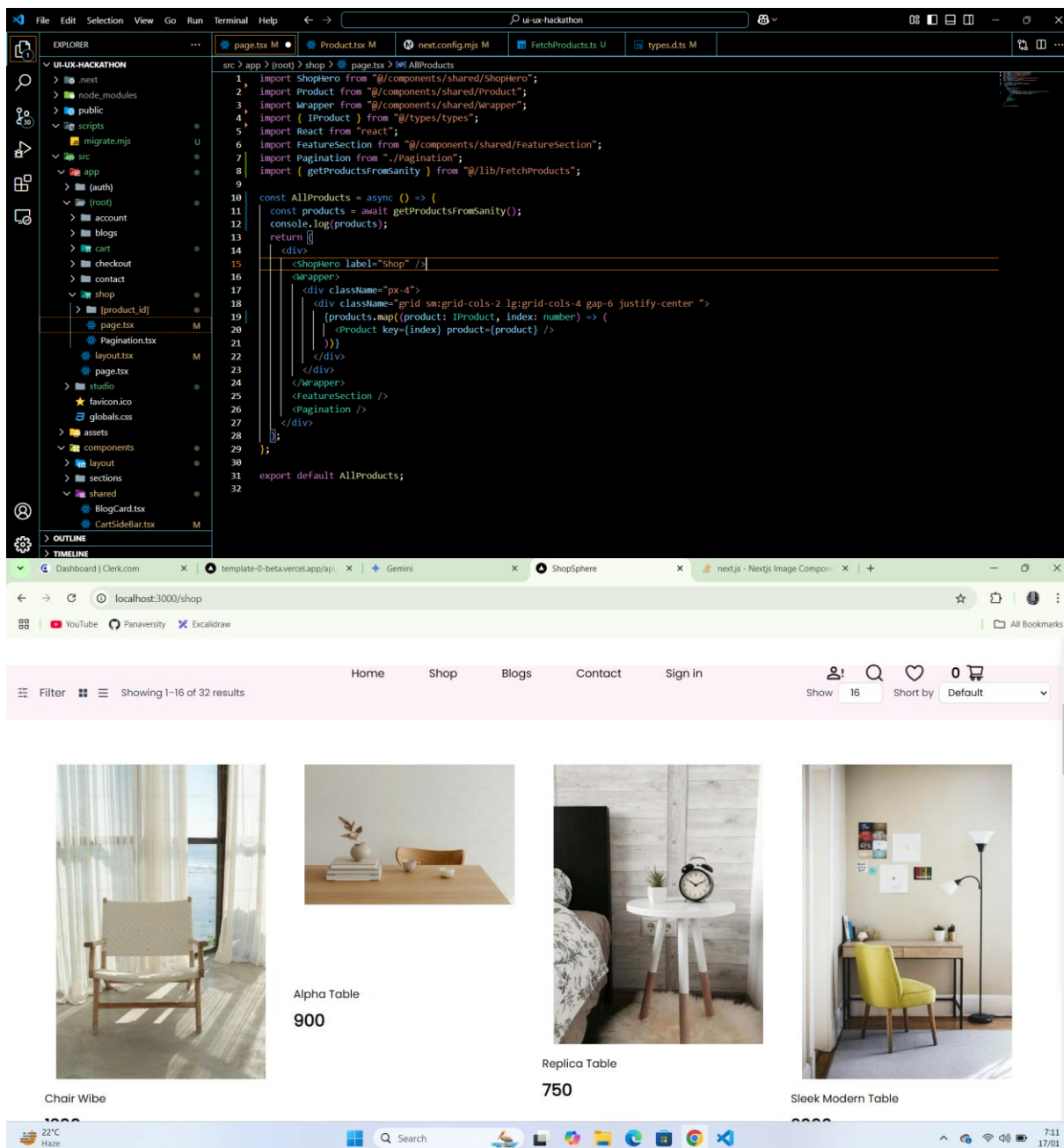
```

•     type: "number",
•   },
•   {
•     name: "description",
•     title: "Description",
•     type: "text",
•   },
•   {
•     name: "discountPercentage",
•     title: "Discount Percentage",
•     type: "number",
•   },
•   {
•     name: "isFeaturedProduct",
•     title: "Is Featured Product",
•     type: "boolean",
•   },
•   {
•     name: "stockLevel",
•     title: "Stock Level",
•     type: "number",
•   },
•   {
•     name: "category",
•     title: "Category",
•     type: "string",
•   },
• ],
• };
•
• export default product;
•

```

Displaying Migrated Data in Frontend

1. **GROQ Query Implementation**
 - Constructed GROQ queries to fetch the newly migrated data from Sanity CMS.
 - Optimized queries for efficient data retrieval.
2. **Next.js Frontend Integration**
 - Successfully displayed the migrated data on the Next.js application.
 - Tested UI to confirm data accuracy and responsiveness.



Migration Steps and Tools Used

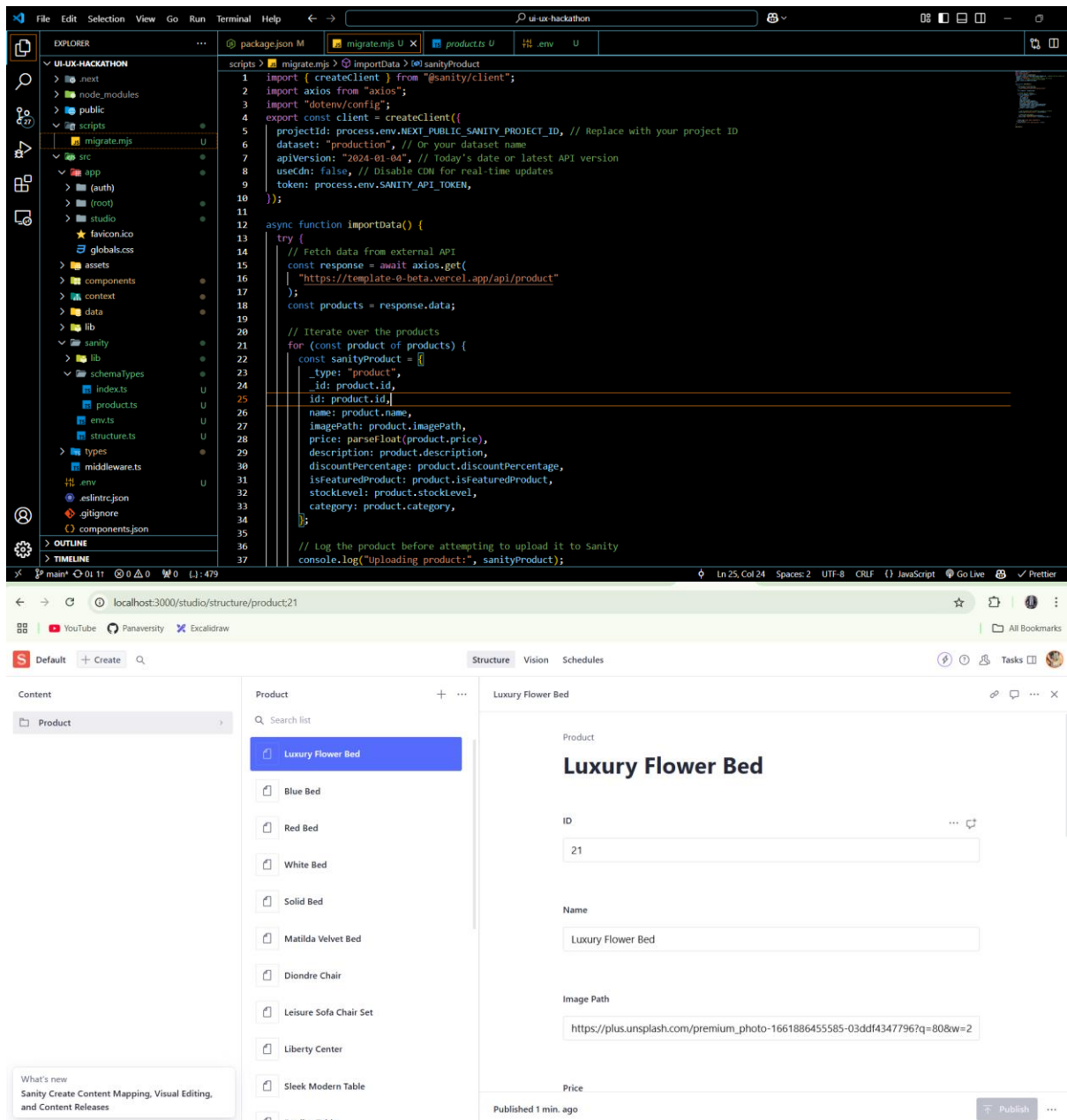
- **Tools Used:**
 - Sanity CLI for schema management.
 - Custom scripts for data migration.
 - GROQ queries for fetching and displaying data.
- **Migration Steps:**

1. Prepared the API and mapped its fields to corresponding Sanity schema fields.

2. Ran the migration script to transfer data into Sanity.
3. Validated the data in Sanity CMS before rendering it in the frontend.

Screenshots Required

1. Script execution and successful migration logs.
2. Data displayed successfully in the frontend.
3. Populated Sanity CMS fields.



EXPLORER

UI-UX-HACKATHON

src

app

auth

account

blogs

cart

checkout

contact

shop

product_id

page.tx

Pagination.tx

layout.tx

page.tx

studio

favicon.ico

globals.css

assets

components

layout

sections

shared

BlogCard.tx

CartSideBar.tx

OUTLINE

TIMELINE

src > app > (root) > shop > page.tx > AllProducts

1 import ShopHero from "@components/shared/ShopHero";

2 import Product from "@components/shared/Product";

3 import Wrapper from "@components/shared/Wrapper";

4 import { IProduct } from "@types/types";

5 import React from "react";

6 import FeatureSection from "@components/shared/FeatureSection";

7 import Pagination from "../Pagination";

8 import { getProductsFromSanity } from "@lib/FetchProducts";

9

10 const AllProducts = async () => {

11 const products = await getProductsFromSanity();

12 console.log(products);

13 return (

14 <div>

15 <ShopHero label="Shop" />

16 <div>

17 <div className="px-4">

18 <div className="grid sm:grid-cols-2 lg:grid-cols-4 gap-6 justify-center">

19 {products.map((product: IProduct, index: number) => (

20 <Product key={index} product={product} />

21))}

22 </div>

23 </div>

24 </Wrapper>

25 <FeatureSection />

26 <Pagination />

27 </div>

28);

29

30

31 export default AllProducts;

32

template-0-beta.vercel.app/api

Gemini

ShopSphere


next.js - Nextjs Image Compon

localhost:3000/shop

Show 16 Short by Default


Filter Showing 1-16 of 32 results

Home Shop Blogs Contact Sign in




Chair Wibe

1000




Alpha Table

900



Replica Table

750



Sleek Modern Table

0000

22°C Haze

Search

7:11 17/01

Dashboard | Clerk.com

template-0-beta.vercel.app/api

Gemini

ShopSphere

next.js - Next.js Image Compon

localhost:3000/shop


YouTube Panaversity Excalidraw

All Bookmarks


Filter Showing 1-16 of 32 results

HomeShopBlogsContact


Show 16Short by Default




Chair Wibe
1200




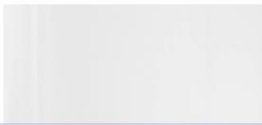


Alpha Table
900



Replica Table
750



Sleek Modern Table
2000



22°C Haze

Search

7:16 17/01

Dashboard | Clerk.com

template-0-beta.vercel.app/api

Gemini

ShopSphere

next.js - Next.js Image Compon

localhost:3000/shop/10




YouTube Panaversity Excalidraw


All Bookmarks

Home > Shop > | Alpha Table

HomeShopBlogsContact

Show 16Short by Default





Alpha Table

★★★★★ 5 Customer Review

Rs. 900

A sturdy oak chair with a sleek and minimalist design.

Size

LXLXS

Color

-1+

Add To Cart

SKU : SS001

Category : Table

Tags : Sofa, Chair, Home, Shop

Share : [f](#) [in](#) [❤](#)

22°C Haze

Search

7:32 17/01