

DAY 3

General E-commerce

DAY3_API_INTEGRATION_AND_DATA_MIGRATION

**PREPARED BY:
MUHAMMD HAMMAD**

Day 3 API Integration and Data Migration

Objective:

The objective of day 2 is to integrating APIs and migrating data into sanity CMS to build a functional marketplace backend

1. Sanity schema design

This is schema of product with fields include:

- title: Title of Product
- price: Price of Product
- priceWithoutDiscount: Actual Price of Product
- badge: Badge on Product e.g New, Sales
- image: Image of Product
- description: Desription of Product

Code:

```
File Edit Selection View Go ... my-app
src > sanity > schemaTypes > products.ts > @productSchema > fields
1 import { defineType } from "sanity";
2
3 export const productSchema = defineType({
4   name: "products",
5   title: "Products",
6   type: "document",
7   fields: [
8     {
9       name: "title",
10      title: "Product Title",
11      type: "string",
12    },
13    {
14      name: "price",
15      title: "Price",
16      type: "number",
17    },
18    {
19      name: "priceWithoutDiscount",
20      title: "Price without Discount",
21      type: "number",
22    },
23    {
24      name: "badge",
25      title: "Badge",
26      type: "string",
27    },
28    {
29      name: "image",
30      title: "Product Image",
31      type: "image",
32    },
33    {
34      name: "description",
35      title: "Product Description",
36      type: "text",
37    },
38  ],
39 });
```

2. Data Migration

I have fetched data from the API to Sanity CMS

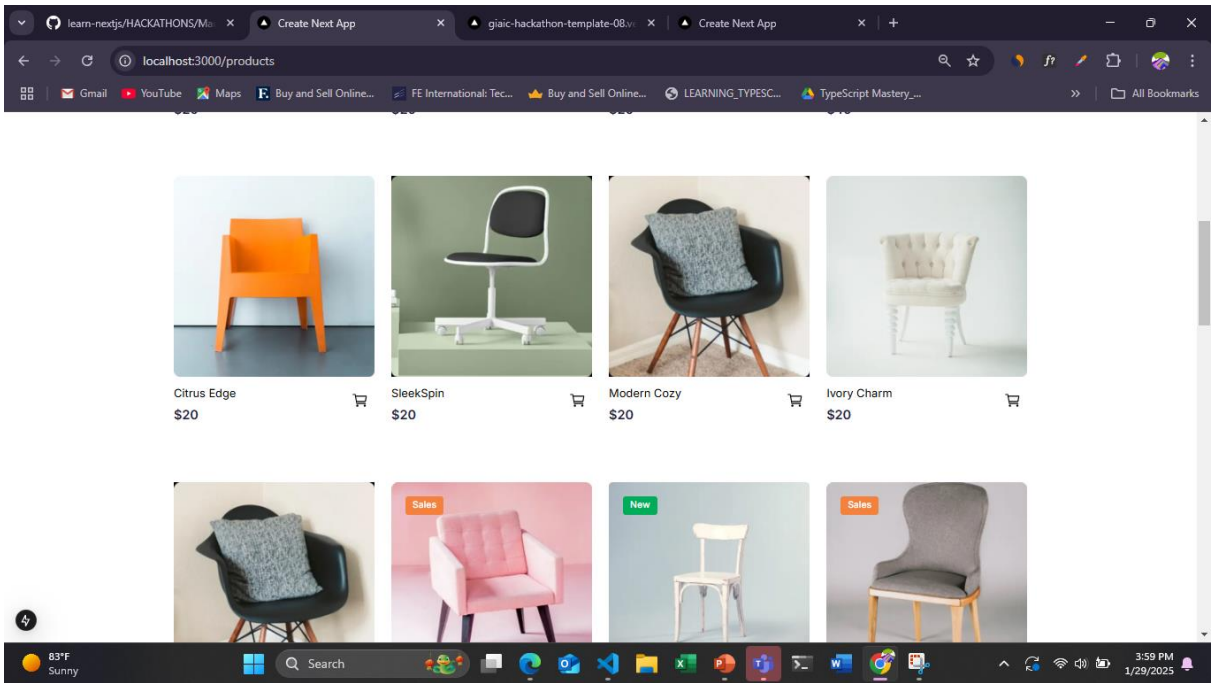
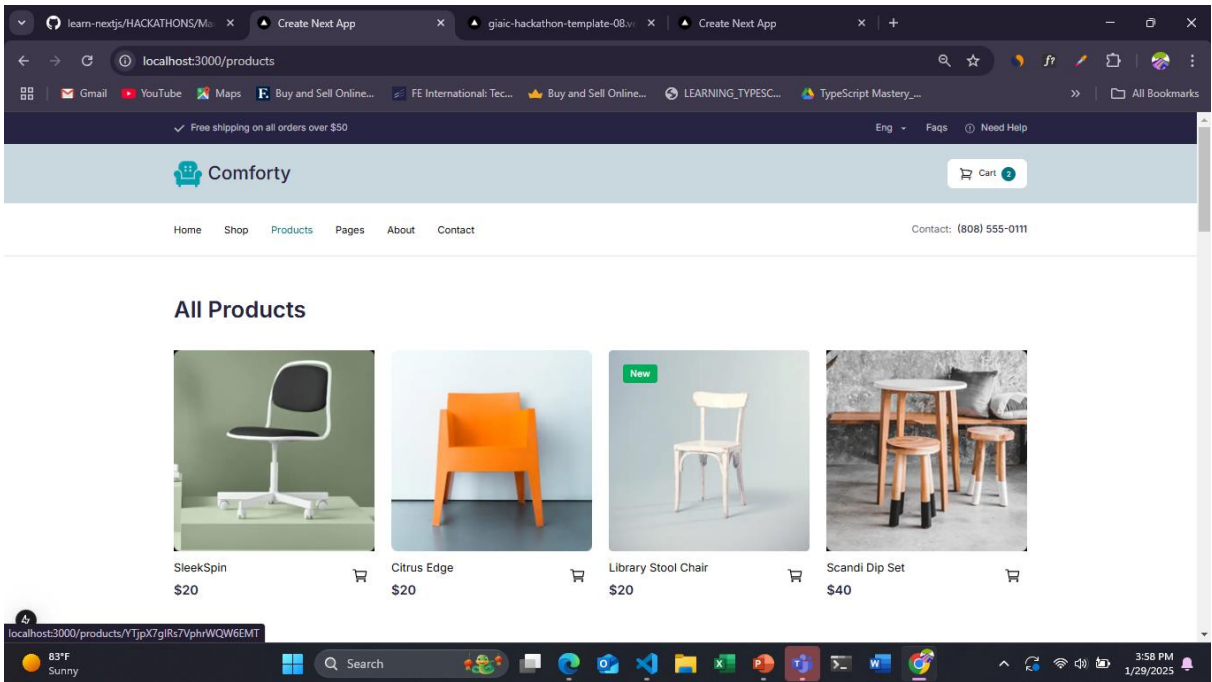
```
File Edit Selection View Go ... my-app
scripts > migrate.mjs
1 // Import environment variables from .env.local
2 import "dotenv/config";
3
4 // Import the Sanity client to interact with the Sanity backend
5 import { createClient } from "sanity/client";
6
7 // Load required environment variables
8 const {
9   NEXT_PUBLIC_SANITY_PROJECT_ID, // Sanity project ID
10   NEXT_PUBLIC_SANITY_DATASET, // Sanity dataset (e.g., "production")
11   NEXT_PUBLIC_SANITY_AUTH_TOKEN, // Sanity API token
12   BASE_URL, // https://glaii-hackathon-template-40.vercel.app, // API base URL for products and categories
13 } = process.env;
14
15 // Check if the required environment variables are provided
16 if (!NEXT_PUBLIC_SANITY_PROJECT_ID || !NEXT_PUBLIC_SANITY_AUTH_TOKEN) {
17   console.error("Missing required environment variables. Please check your .env.local file.");
18   process.exit(1); // Stop execution if variables are missing
19 }
20
21 // Create a Sanity client instance to interact with the target Sanity dataset
22 const targetClient = createClient({
23   projectId: NEXT_PUBLIC_SANITY_PROJECT_ID, // Your Sanity project ID
24   dataset: NEXT_PUBLIC_SANITY_DATASET || "production", // Default to "production" if not set
25   useCdn: false, // Disable CDN for real-time updates
26   apiVersion: "2023-01-01", // Sanity API version
27   token: NEXT_PUBLIC_SANITY_AUTH_TOKEN, // API token for authentication
28 });
29
30 // Function to upload an image to Sanity
31 async function uploadImageToSanity(imageUrl) {
32   try {
33     // Fetch the image from the provided URL
34     const response = await fetch(imageUrl);
35     if (!response.ok) throw new Error("Failed to fetch image: ${imageUrl}");
36
37     // Convert the image to a buffer (binary format)
38     const buffer = await response.arrayBuffer();
39
40     // Upload the image to Sanity and get its asset ID
41     const uploadedAsset = await targetClient.assets.upload("image", Buffer.from(buffer), {
42       filename: imageUrl.split("/").pop(), // Use the file name from the URL
43     });
44
45     return uploadedAsset._id; // Return the asset ID
46   } catch (error) {
47     console.error("Error uploading image:", error.message);
48     return null; // Return null if the upload fails
49   }
50 }
```

3. API Data Fetching

Now the data display in frontend

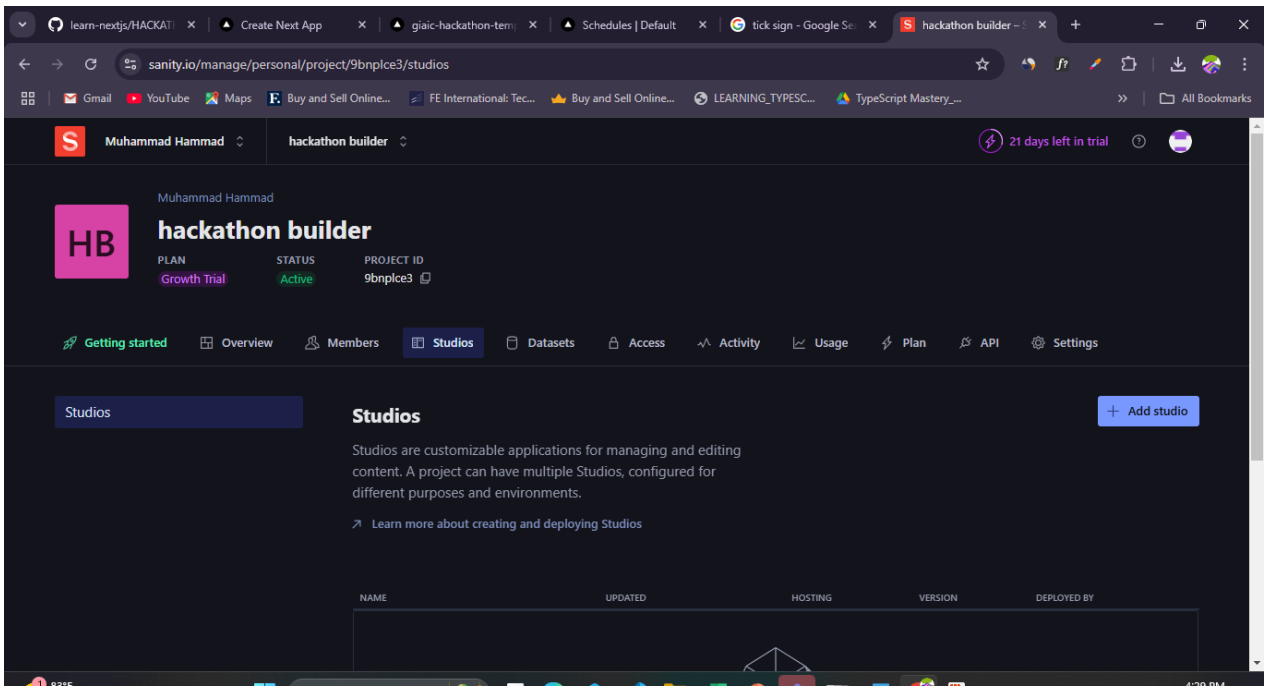
```
export const getProductData = async () => {
  const res = await client.fetch(`*[_type=='products']{
    title,
    _id,
    badge,
    priceWithoutDiscount,
    price,
    description,
    image
  }`)
  return res
}
```

4. Screenshots of Sanity and Marketplace

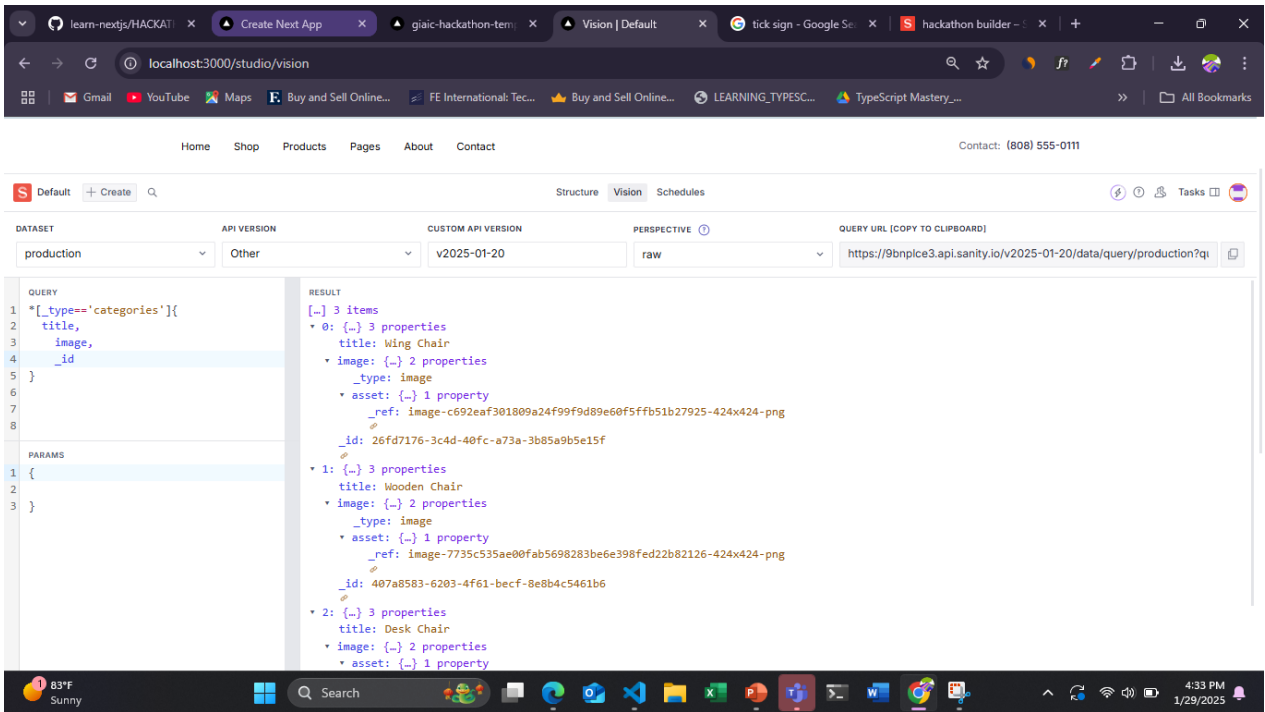


5. Tools Used

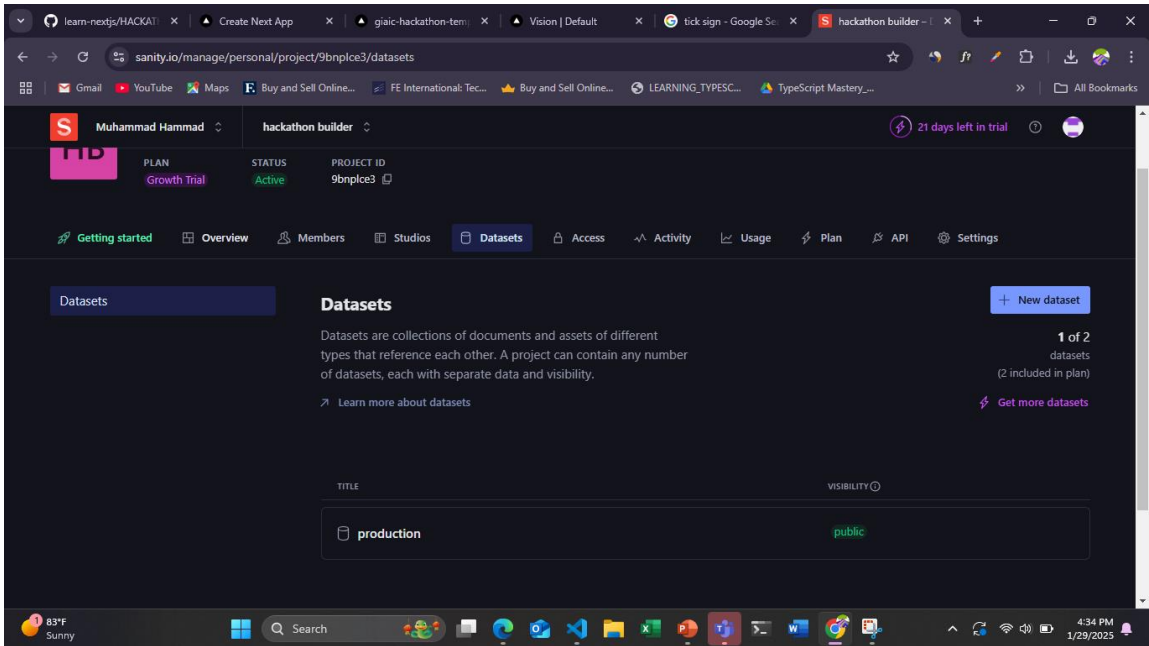
- Sanity Studio



- Sanity Vision



- Sanity Dataset



Checklist

API Understanding



Schema Validation



Data Migration



API Integration in NextJS



Submission Preparation

