Day 3 - API Integration Report –General Ecommerce Comforty-website

1. API Integration Process

Overview of the API Integration:

On Day 3 of the project, I focused on integrating the required external API into the **Comforty** project to fetch product data and display it on the product page. This integration will allow dynamic interaction with external data sources such as product details, prices, and inventory information.

Setting Up the API Client:

- I installed the necessary dependencies for the Sanity API using the following commands.

```
npm install @sanity/client
```

Configured the client by creating a connection to the Sanity project..

```
import sanityClient from '@sanity/client';

export const client = sanityClient({
  projectId: 'your-project-id',
  dataset: 'production',
  useCdn: true,
});
```

Fetching Data:

- I set up an asynchronous function using **useEffect** to fetch product data from the API:

```
useEffect (() => {
  const fetchProducts = async () => {
    const query = `*[_type == "products”] {title, price, description, imageUrl, slug} `;
    const fetched Data = await client. Fetch(query);
    setProducts (fetched Data);
  };
  fetchProducts ();
}, []);
```

Challenges Faced:

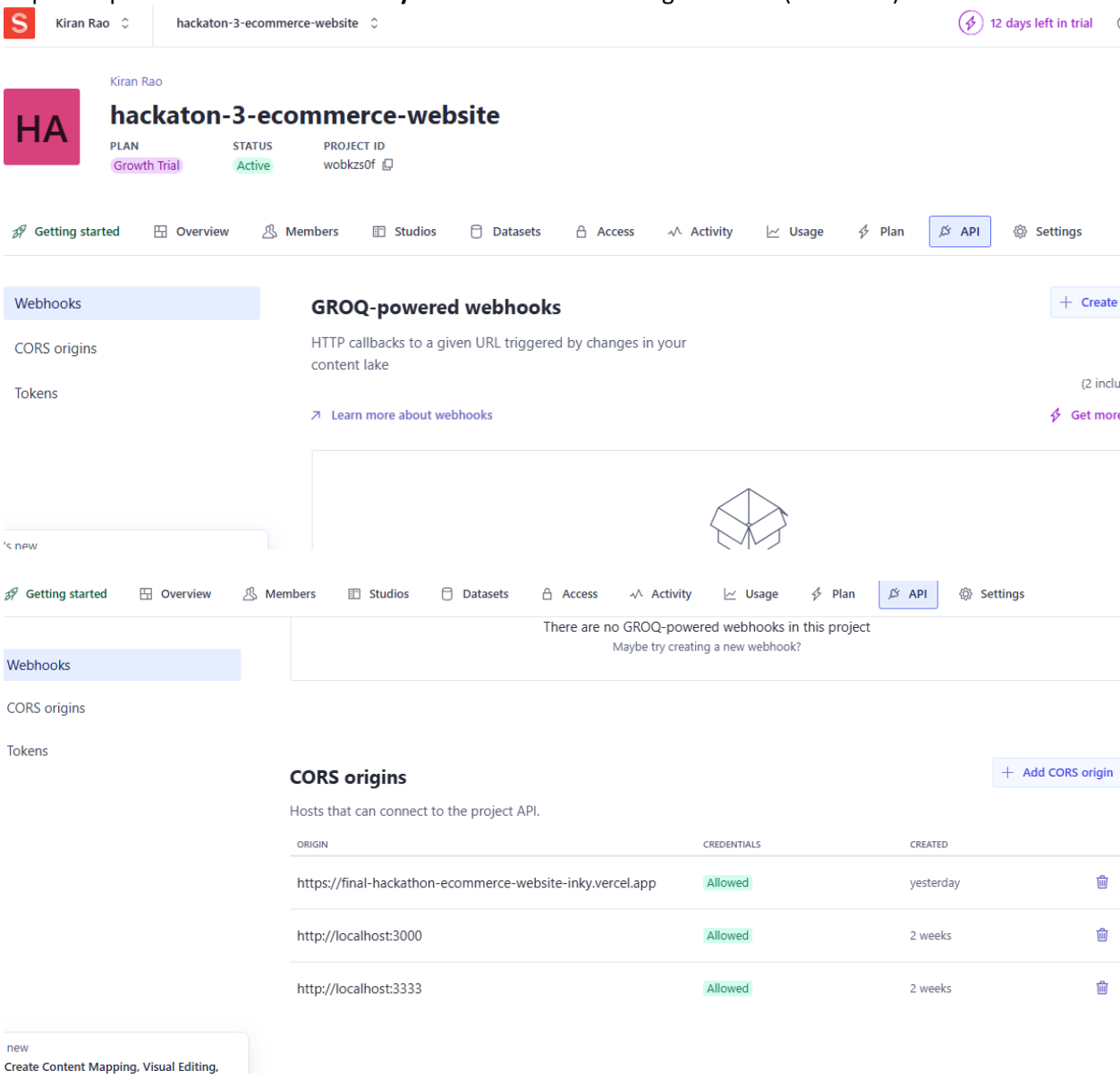
- **API Response Delays**: Initially, some products were not loading due to slow API response times. This was mitigated by adding a loading state.
- **Error Handling**: I added error handling to display a user-friendly message when the data fetch failed.

Migration Steps and Tools Used

Data Migration:

To integrate the existing data into the Comforty project, I performed the following migration steps:

1. **Preparing Data for Migration:**
   - I created a backup of the existing data to avoid any loss during the migration process.
   - Imported product data into the **Sanity Studio** from the existing database (if needed)



.env

.env

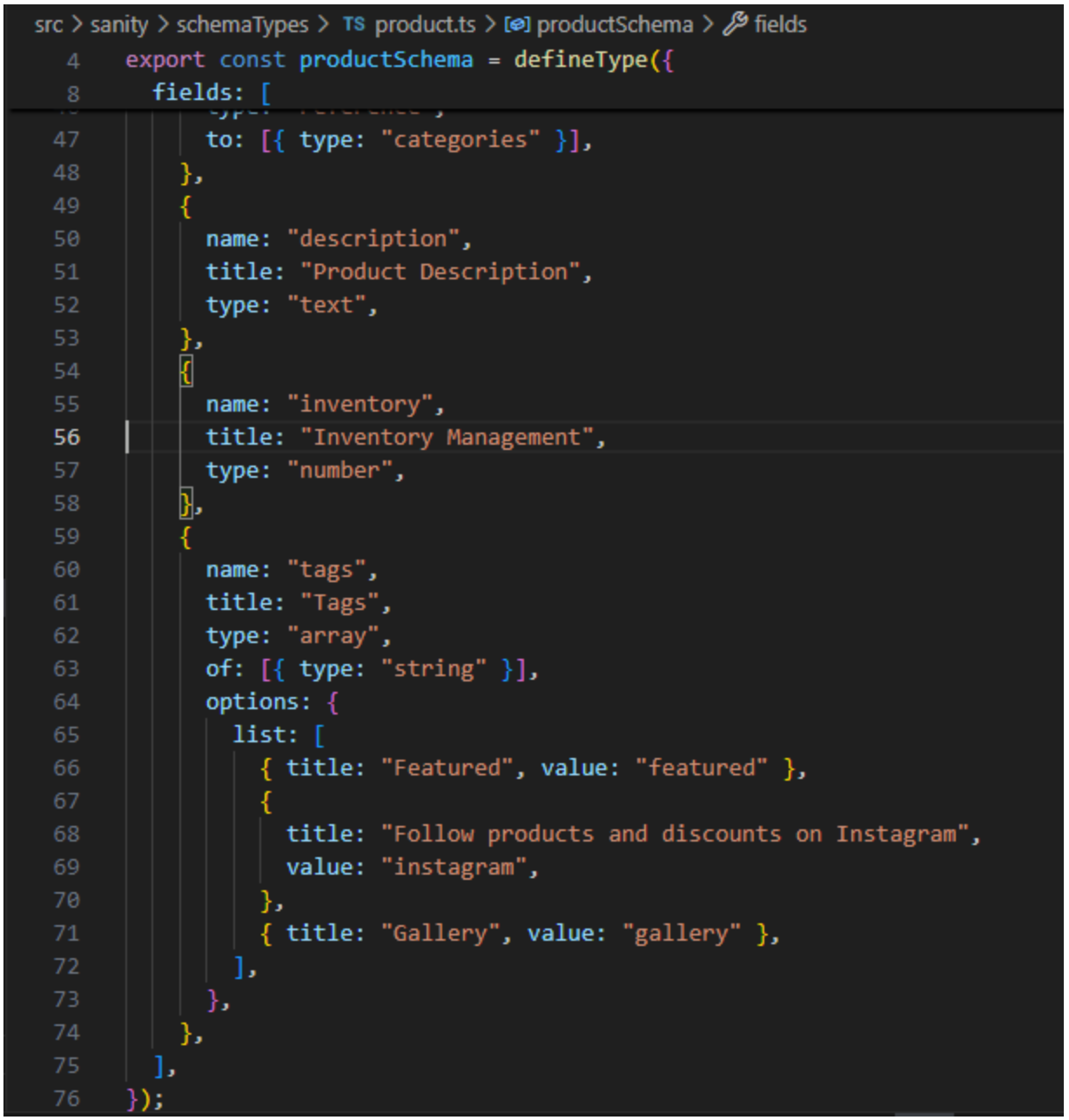
```
NEXT_PUBLIC_SANITY_PROJECT_ID="wobkzs0f"
NEXT_PUBLIC_SANITY_DATASET="production"
SANITY_API_TOKEN="skwHzDDaqmsCRN2UH5ZgFBvDfF3lSgTu4CkNrBtp5DJHPycs6CG8rTHJnuYvNWG
2Qi2RryteOPURBn6eQJWTNXTGkJx26VzAPBOpWOpwbcJ0wGATGxE8AgSx529s97Q8g9RavdY8ndZwvoDc
FW8yk4GqBjsuCLQ7jONUp0Z71ANIZIlo4fHy"

```

src > sanity > schemaTypes > TS index.ts > ...

```ts
import { type SchemaTypeDefinition } from 'sanity'
import  { productSchema } from './product'
import { categorySchema } from './categories'

export const schema: { types: SchemaTypeDefinition[] } = {
  types: [productSchema,categorySchema],
}

```

src > sanity > schemaTypes > TS product.ts > productSchema > fields

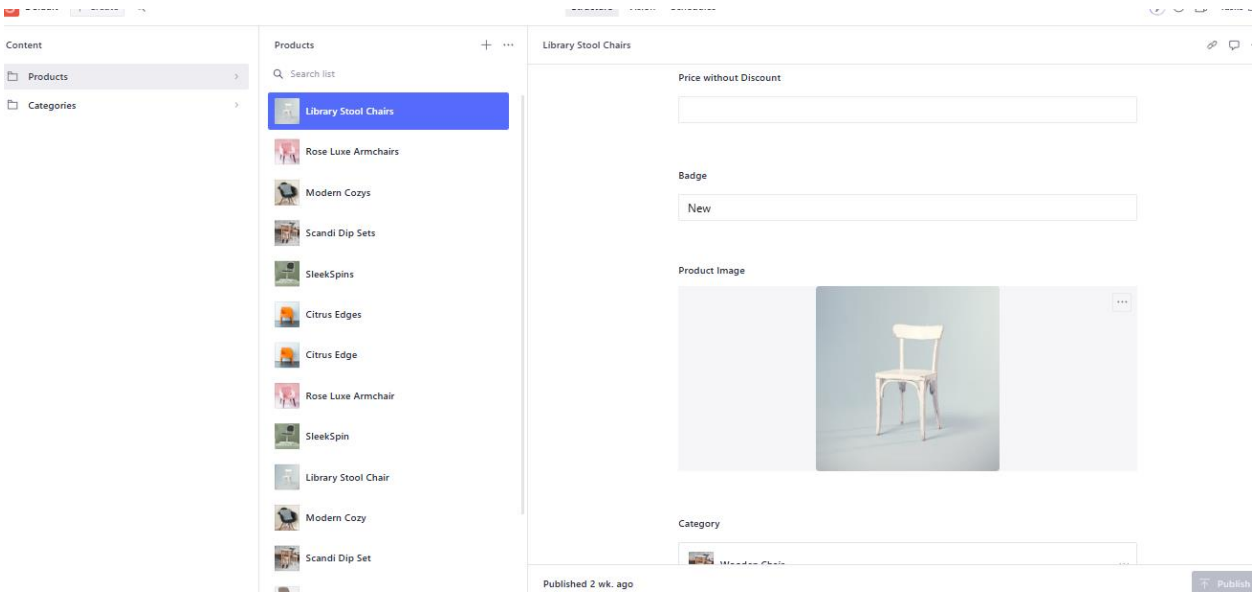
```ts

import { defineType } from "sanity";

export const productSchema = defineType({
  name: "products",
  title: "Products",
  type: "document",
  fields: [
    {
      name: "title",
      title: "Product Title",
      type: "string",
    },
    {
      name: "slug",
      title: "Slug",
      type: "slug",
      options: {
        source: "title",
        maxLength: 96,
      },
    },
    {
      name: "price",
      title: "Price",
      type: "number",
    },
    {
      title: "Price without Discount",
      name: "priceWithoutDiscount",
      type: "number",
    },
```

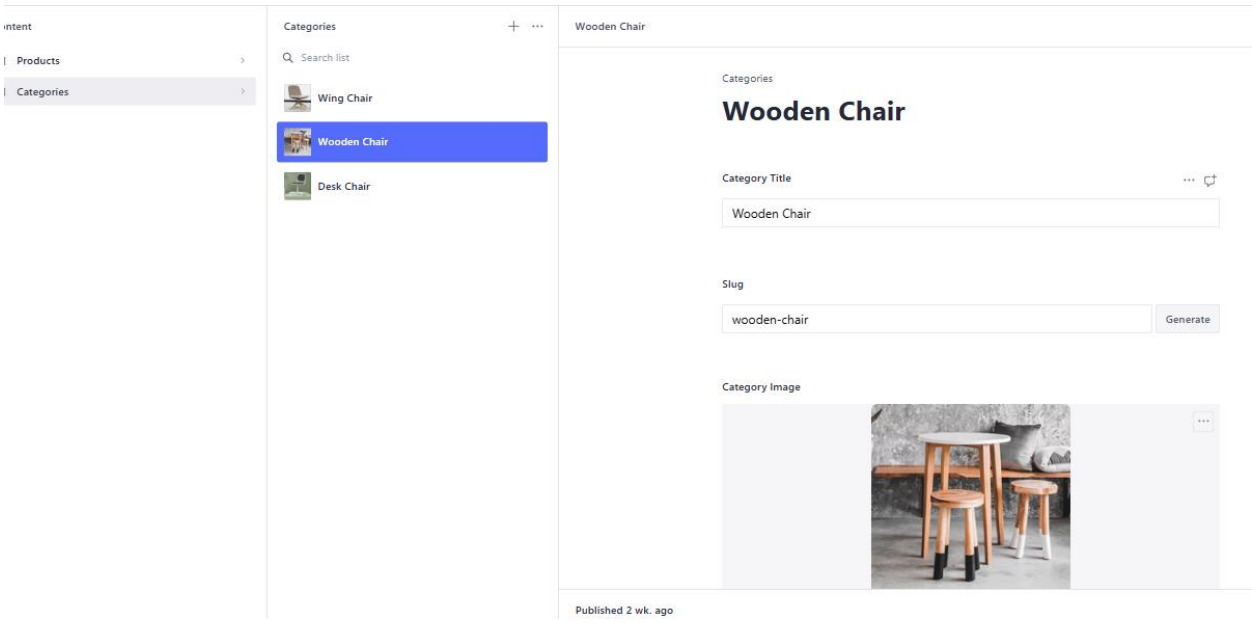


script/Migrate.js

scripts > JS migrate.mjs > ...

```
1
2   // Import environment variables from .env.local
3   import "dotenv/config";
4
5   // Import the Sanity client to interact with the Sanity backend
6   import { createClient } from "@sanity/client";
7
8   // Load required environment variables
9   const {
10    NEXT_PUBLIC_SANITY_PROJECT_ID, // Sanity project ID
11    NEXT_PUBLIC_SANITY_DATASET, // Sanity dataset (e.g., "production")
12    SANITY_API_TOKEN, // Sanity API token
13    BASE_URL = "https://giaic-hackathon-template-08.vercel.app", // API base URL for products and categories
14  } = process.env;
15
16  // Check if the required environment variables are provided
17  if (!NEXT_PUBLIC_SANITY_PROJECT_ID || !SANITY_API_TOKEN) {
18    console.error("Missing required environment variables. Please check your .env.local file.");
19    process.exit(1); // Stop execution if variables are missing
20  }
21
22  // Create a Sanity client instance to interact with the target Sanity dataset
23  const targetClient = createClient({
24    projectId: NEXT_PUBLIC_SANITY_PROJECT_ID, // Your Sanity project ID
25    dataset: NEXT_PUBLIC_SANITY_DATASET || "production", // Default to "production" if not set
26    useCdn: false, // Disable CDN for real-time updates
27    apiVersion: "2023-01-01", // Sanity API version
28    token: SANITY_API_TOKEN, // Correct API token for authentication
29  });
30
31  // Function to upload an image to Sanity
32  async function uploadImageToSanity(imageUrl) {
```





Tools Used:

- **Sanity Studio**: Used for data management and schema creation.
- **Sanity CLI**: Used to import the migrated product data into the Sanity project.
- **Next.js**: Utilized for API integration and rendering product data dynamically.
- **React**: Used for managing component state and API data fetching.

Conclusion

The API integration and data migration steps have been successfully implemented. The Comforty project now fetches and displays product data dynamically, with full integration with Sanity for real-time updates. The necessary adjustments to the product schema were made to match the project requirements, and all data was migrated seamlessly using the Sanity tools.