Hackathon Day 03

API-INTERGRATION AND MIGRATION

Hackathon Day 3, where we migrated data, created schemas, built APIs, imported data into Sanity CMS, and displayed the data on a front-end application using Next.js.

1. Data Migration

Objective :

To migrate data from an API source and prepare it for use in our application..

Steps:

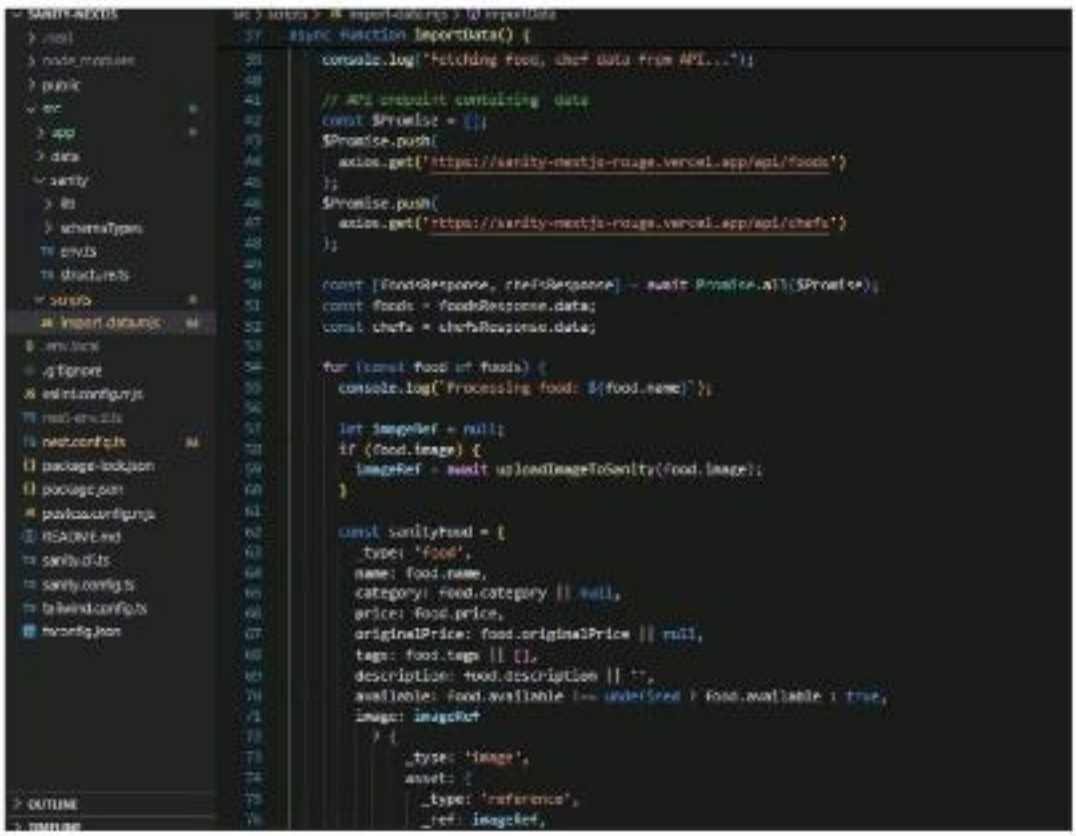
1.Write a script to fetch the data from the API.

2.Example: https://sanity-nextjs-rouge.vercel.app/api/foods'

Write a script to fetch the data from the API..

. Use libraries like axios or node-fetch in your JavaScript/Node.js script.

Example :



Clean and transform the data (if necessary) to match the desired format.

Save the transformed data into a format compatible with Sanity (e.g., JSON).

2. Creating the Schema

To define the structure of the data in Sanity CMS.

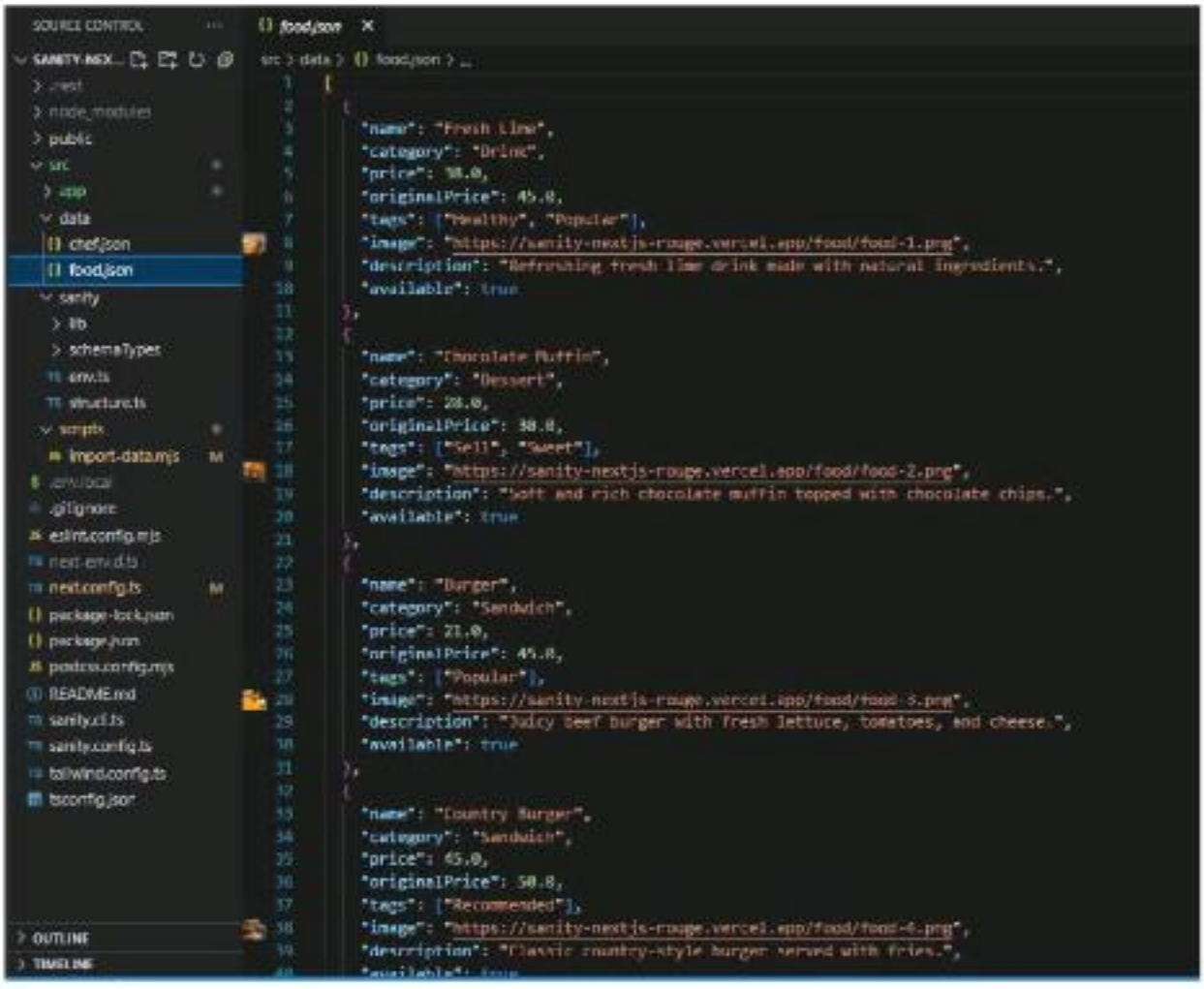
Steps:

1.Install Sanity CLI and initialize a new Sanity project:

npm install -g @sanity/cli

Define a schema for the data in the schemas folder of your Sanity project

Example Sehema:

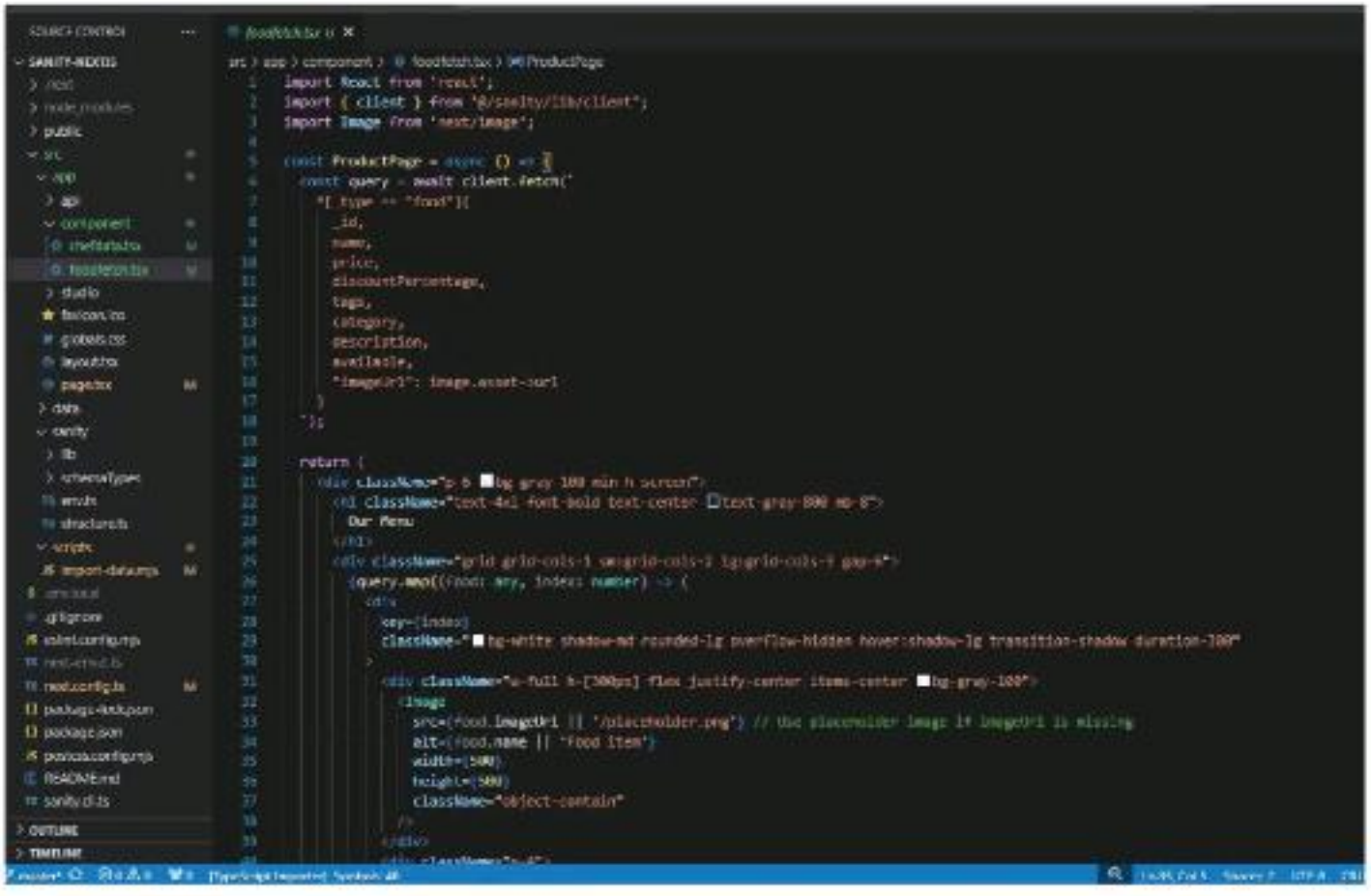


3. Building the API

Objective:

Steps:

Create a new API route in your Next.js application under the pages/api directory.



4. Importing Data into Sanity CMS

1.Make a project on sanity .

2.Create a .env.local file in the root of the project directory:

3.Open .env.local and add the following environment variables:

```
NEXT_PUBLIC_SANITY_PROJECT_ID="{your-sanity-project-id}"
NEXT_PUBLIC_SANITY_DATASET="production"
SANITY_API_TOKEN="{your-sanity-api-token}"
```

Eample

.env.local file

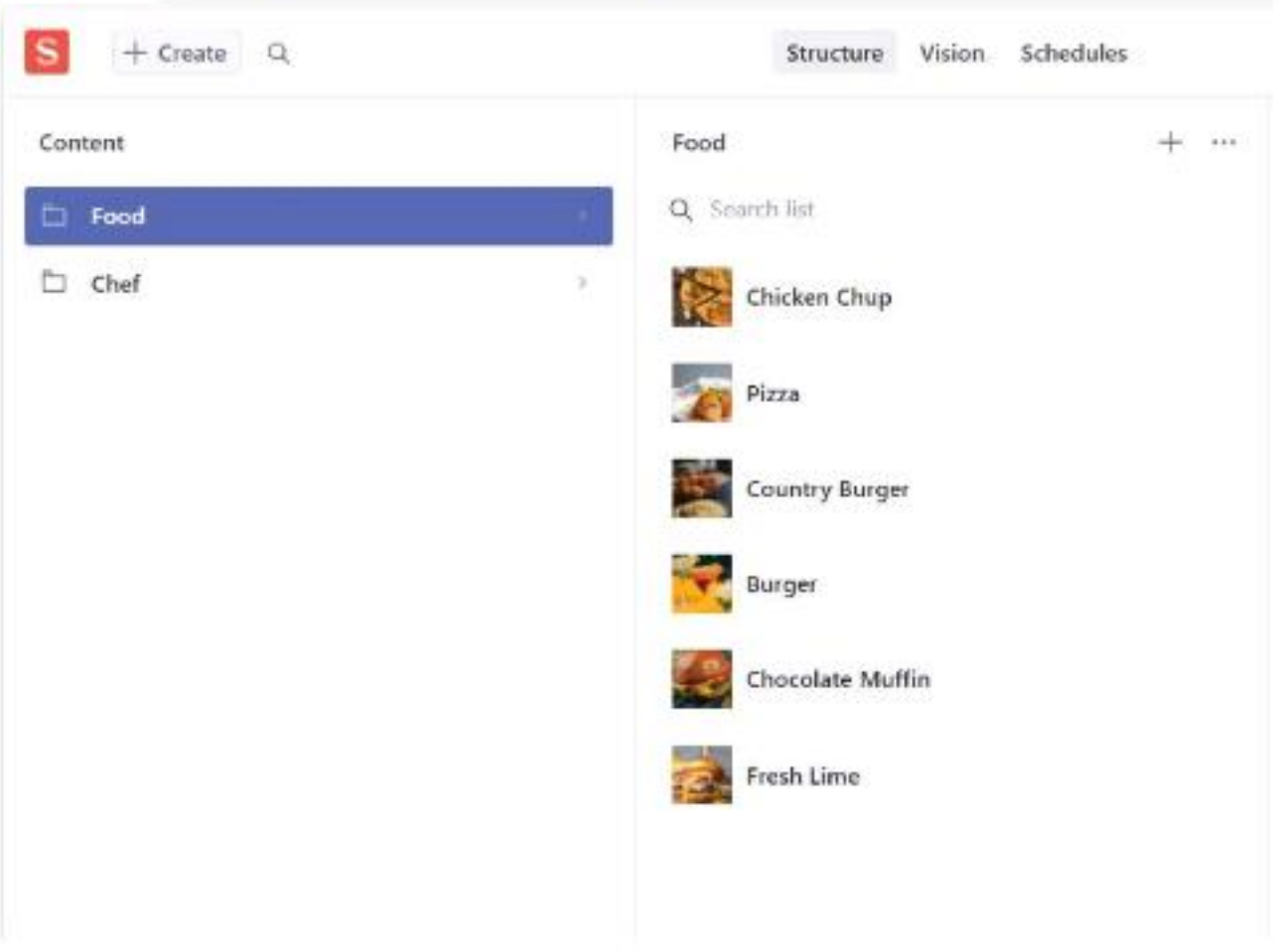
Import Data into sanity cammand :

npm run import-data on sanity .

and npm run dev.

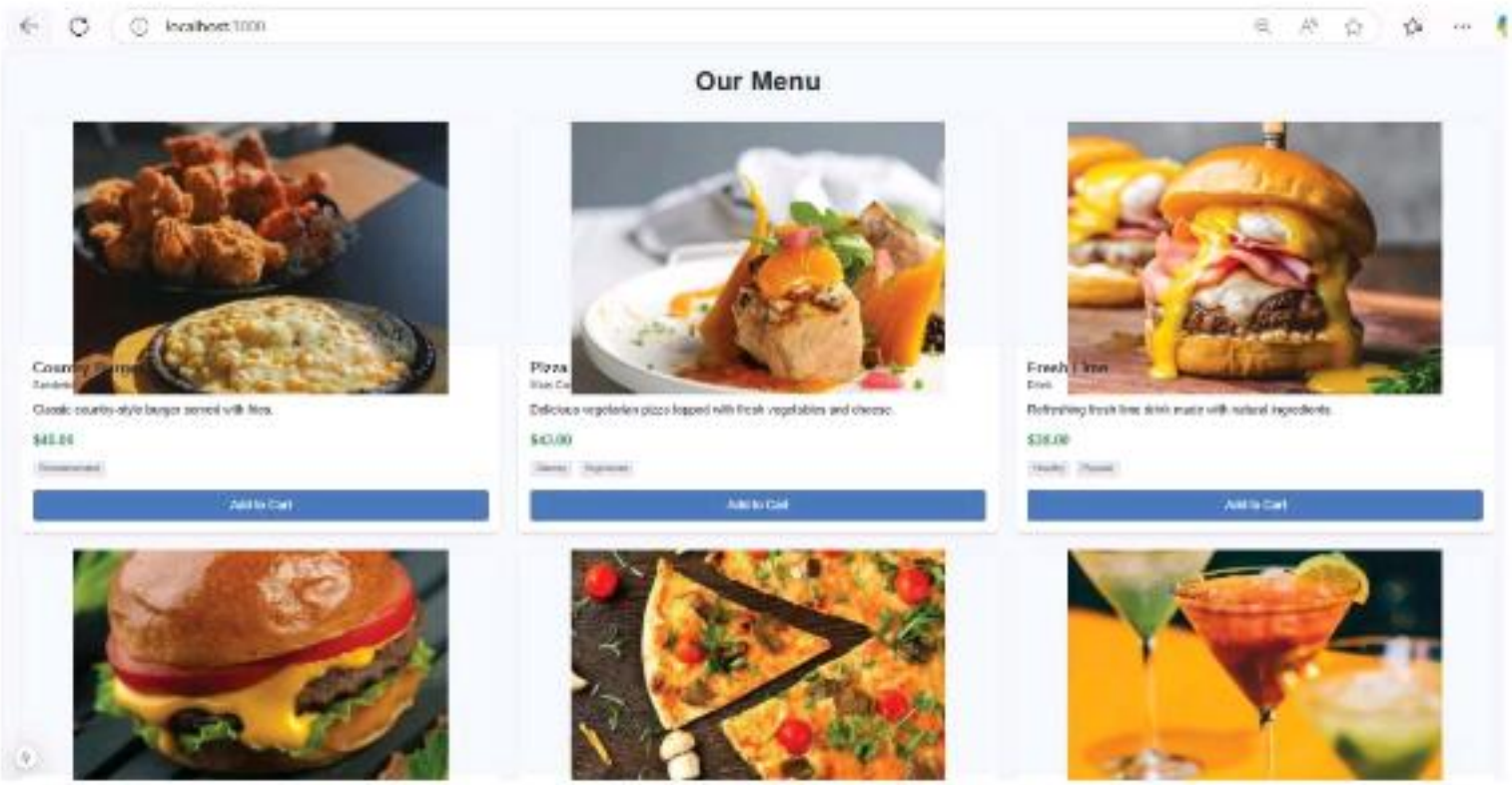
http://localhost:3000/studio

Step 5: Verify the Data in Sanity Studio



Step 6:Displaying Data on the Front-End

To fetch and display the data in a Next.js front-end application.



7. Conclusion

By following these steps, we successfully:

- Migrated data from an API.
- Created a schema in Sanity CMS.
- Built a custom API to fetch data.
- Imported data into Sanity.
- Displayed the data on the front-end using Next.js.

This workflow can be adapted for similar projects to streamline the process of data migration API creation, and front-end integration.

THANK YOU