

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

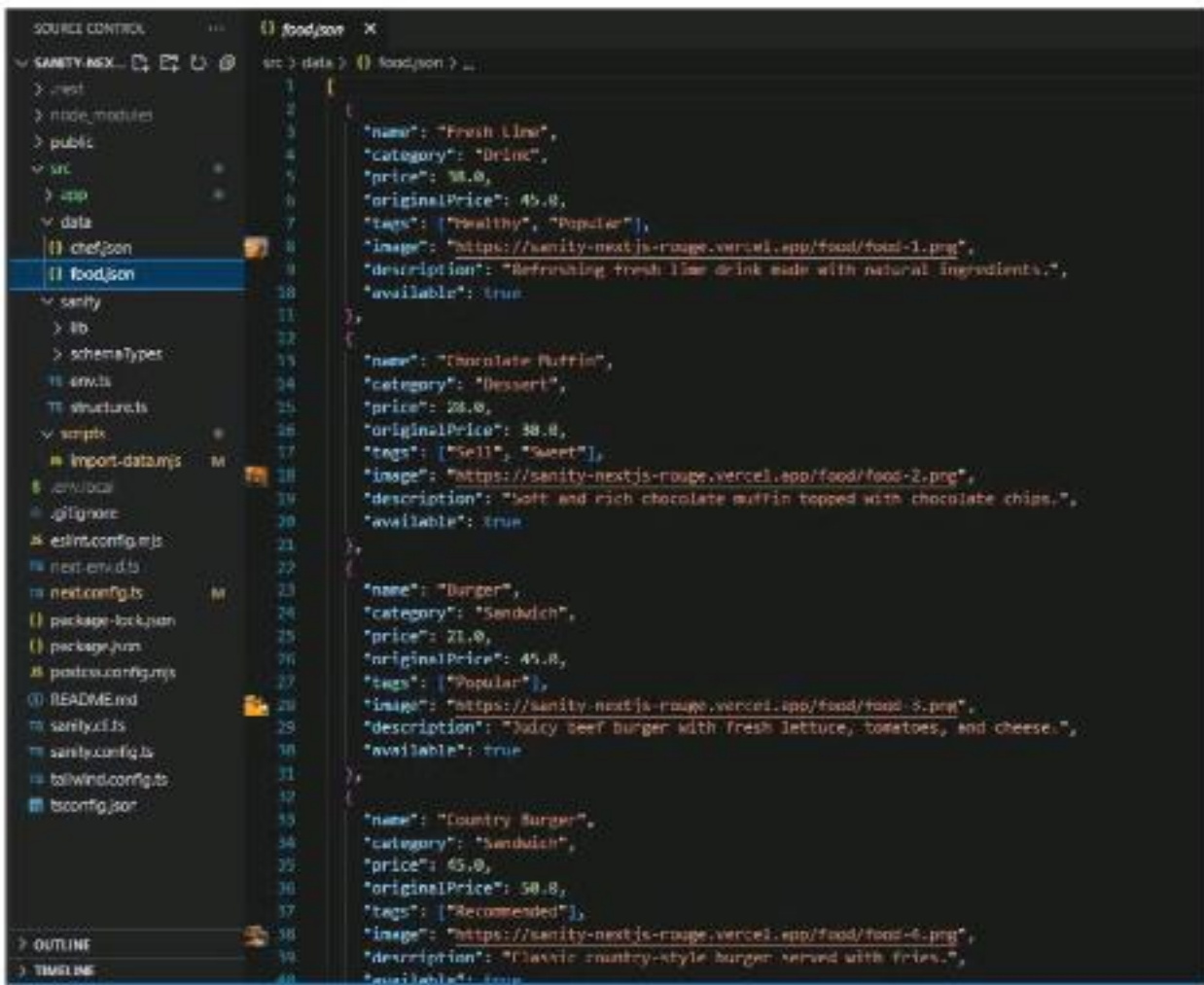
To define the structure of the data in Sanity CMS.

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:



```

SOURCE CONTROL
SANITY-REX...
> .next
> node_modules
> public
> src
  > app
    > data
      (I) chef.json
      (I) food.json
  > sanity
    > lib
    > schemaType
  > env.js
  > structure.js
  > scripts
    > import-data.js
  > .env.local
  > .gitignore
  > eslintrc.json
  > next-env.d.ts
  > next.config.js
  > package-lock.json
  > package.json
  > postcss.config.js
  > README.md
  > sanity.d.ts
  > sanity.config.js
  > tailwind.config.js
  > tsconfig.json
  > OUTLINE
  > TIMELINE

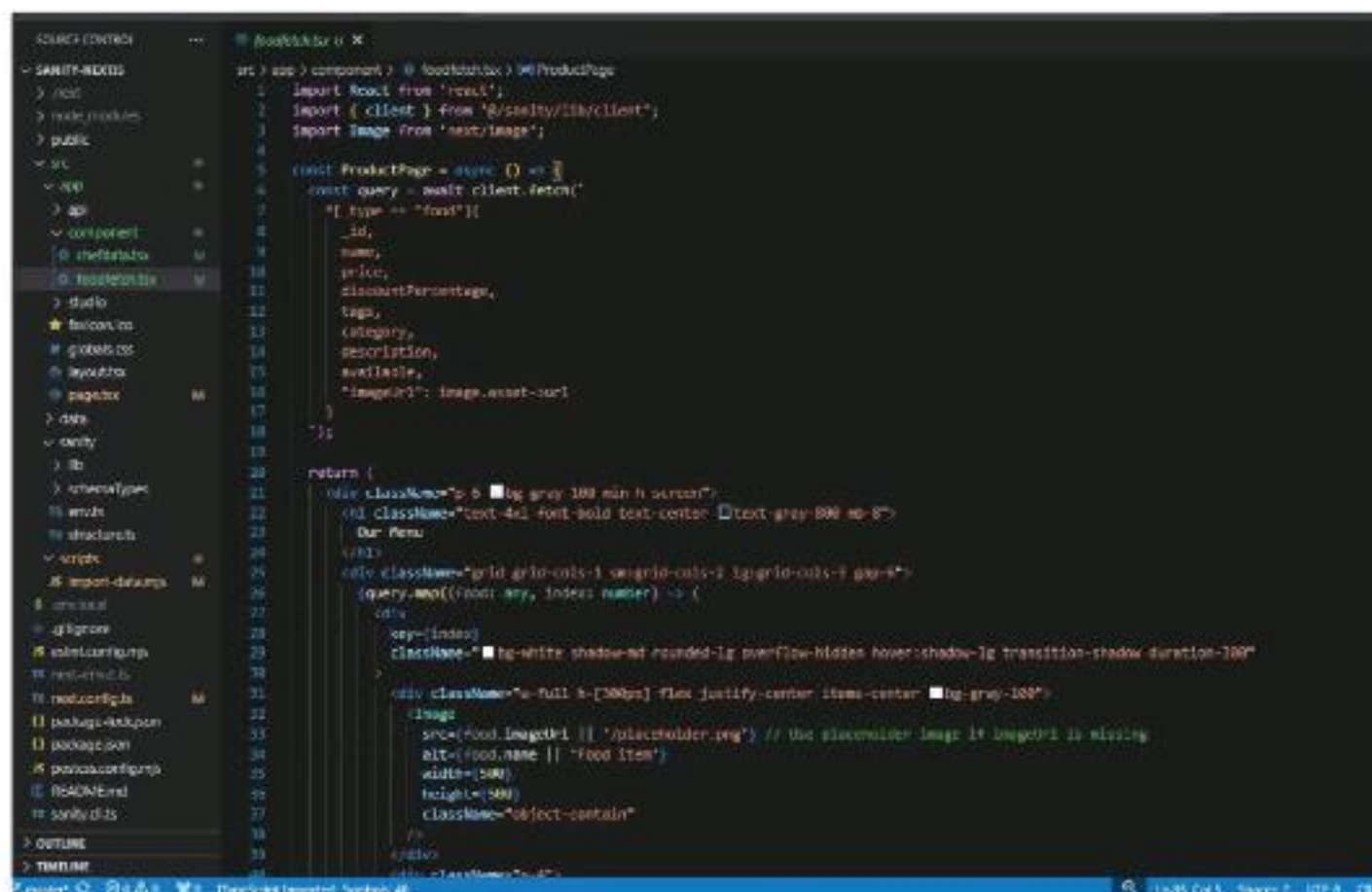
src > data > (I) food.json > ...
{
  "name": "Fresh Lime",
  "category": "Drink",
  "price": 38.0,
  "originalPrice": 45.0,
  "tags": ["Healthy", "Popular"],
  "image": "https://sanity-nextjs-rouge.vercel.app/food/food-1.png",
  "description": "Refreshing fresh lime drink made with natural ingredients.",
  "available": true
},
{
  "name": "Chocolate Muffin",
  "category": "Dessert",
  "price": 28.0,
  "originalPrice": 38.0,
  "tags": ["Sell", "Sweet"],
  "image": "https://sanity-nextjs-rouge.vercel.app/food/food-2.png",
  "description": "Soft and rich chocolate muffin topped with chocolate chips.",
  "available": true
},
{
  "name": "Burger",
  "category": "Sandwich",
  "price": 21.0,
  "originalPrice": 45.0,
  "tags": ["Popular"],
  "image": "https://sanity-nextjs-rouge.vercel.app/food/food-3.png",
  "description": "Juicy beef burger with fresh lettuce, tomatoes, and cheese.",
  "available": true
},
{
  "name": "Country Burger",
  "category": "Sandwich",
  "price": 65.0,
  "originalPrice": 58.0,
  "tags": ["Recommended"],
  "image": "https://sanity-nextjs-rouge.vercel.app/food/food-4.png",
  "description": "Classic country-style burger served with fries.",
  "available": true
}
```

3. Building the API

Objective:

Steps:

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



```
1 import React from 'react';
2 import { client } from '@sanity/client';
3 import Image from 'next/image';
4
5 const ProductPage = async () => {
6   const query = await client.fetch(
7     `*[_type == "food"]{
8       _id,
9       name,
10      price,
11      discountPercentage,
12      tags,
13      category,
14      description,
15      available,
16      "imageId": image.asset.url
17    }`
18   );
19
20   return (
21     <div className="p-6 lg:gray-100 min-h-screen">
22       <h1 className="text-4xl font-bold text-center text-gray-800">
23         Our Menu
24       </h1>
25       <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">
26         {query.map((food, index) => {
27           <div
28             key={index}
29             className="bg-white shadow-md rounded-lg overflow-hidden hover:shadow-lg transition-shadow duration-100"
30           >
31             <div className="h-full h-[300px] flex justify-center items-center bg-gray-100">
32               <img
33                 src={food.imageId || "/placeholder.png"} // the placeholder image if imageId is missing
34                 alt={food.name || "Food item"}
35                 width={500}
36                 height={500}
37                 className="object-contain"
38               />
39             </div>
40             <div className="p-4">
41               <h3>{food.name}</h3>
42               <p>{food.description}</p>
43               <p>{food.price}</p>
44               <p>{food.discountPercentage}</p>
45               <p>{food.tags}</p>
46             </div>
47           </div>
48         })}
```

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}"
```

Example

.env.local file



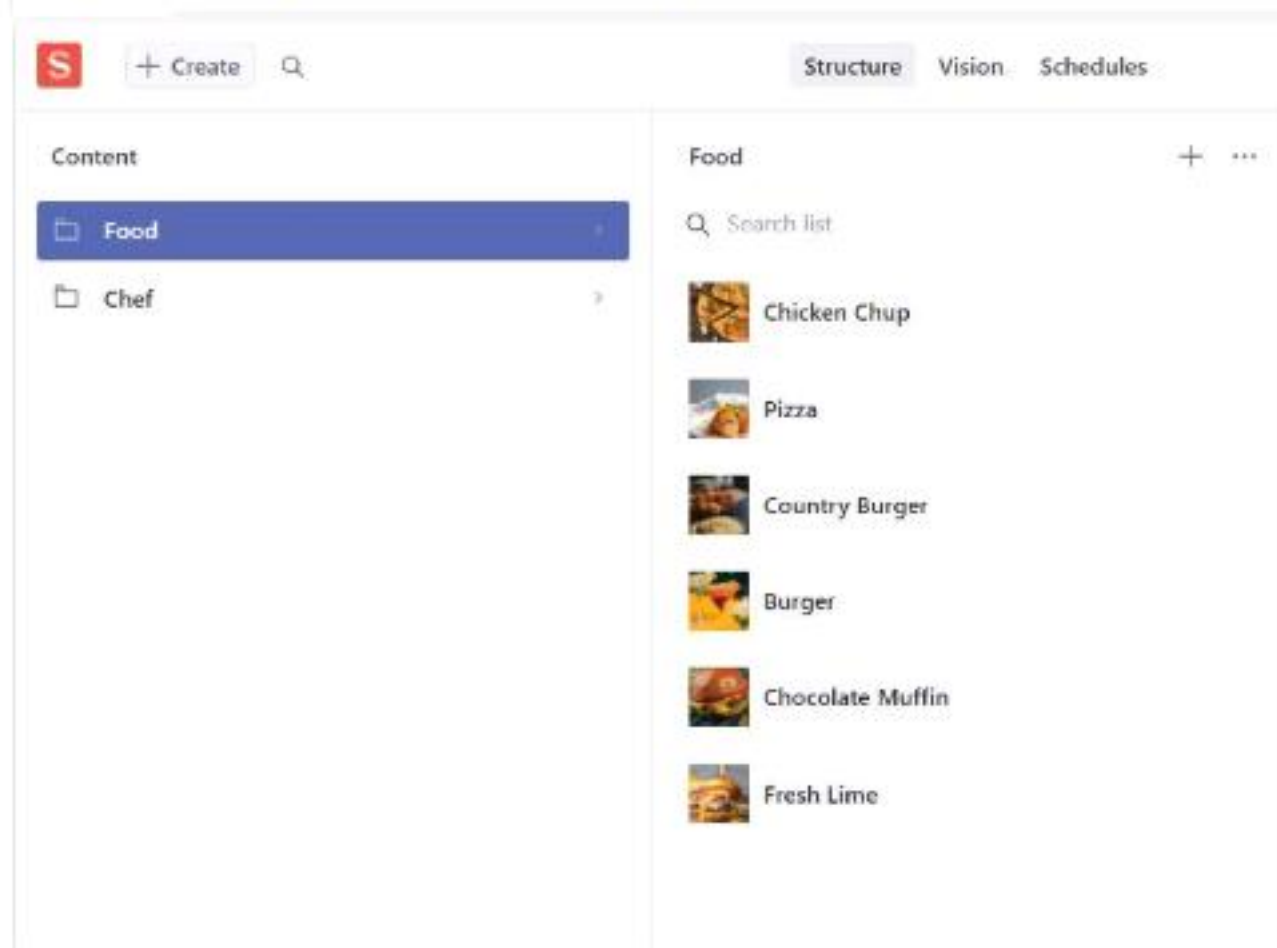
Import Data into sanity cammand :

npm run import-data on sanity .

and npm run dev.

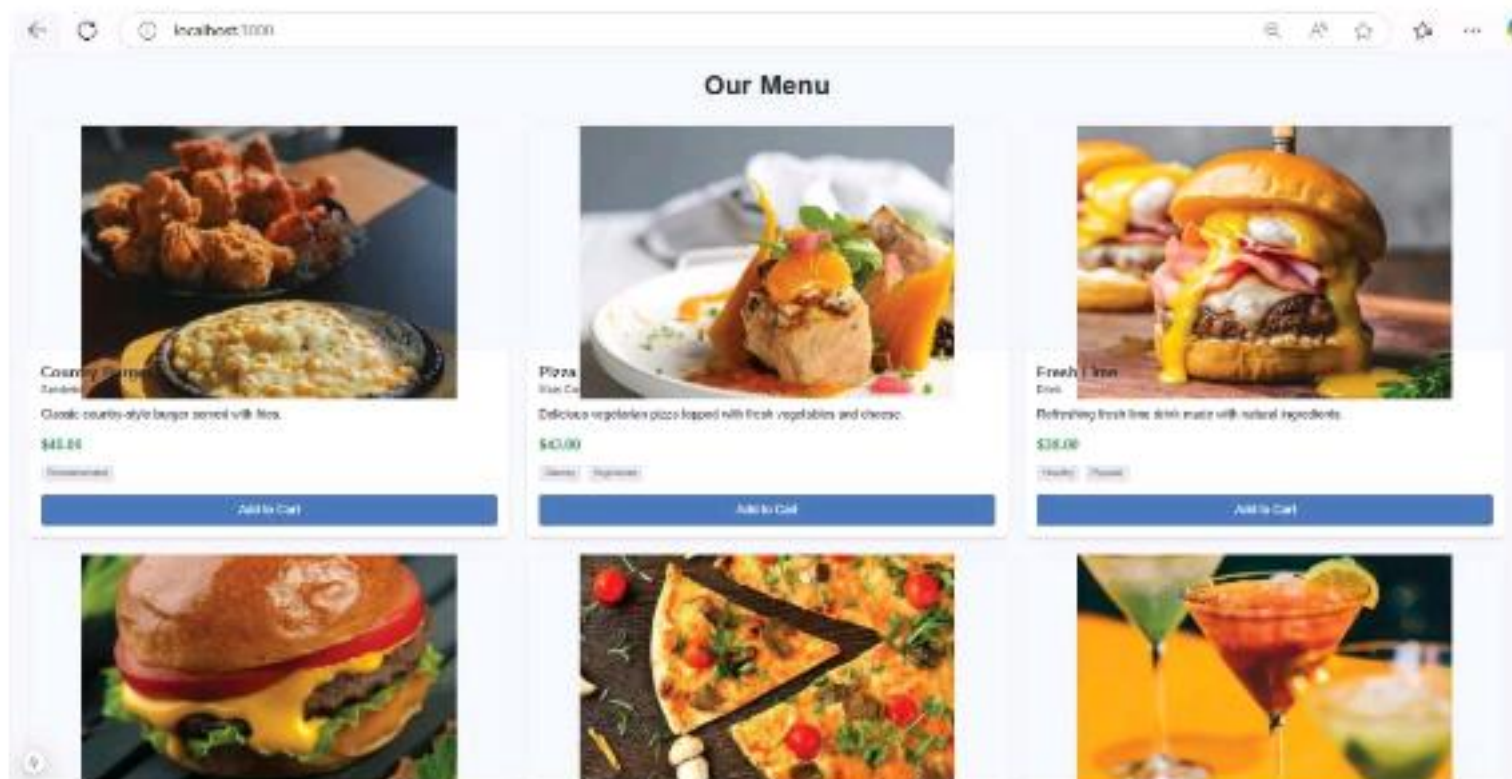
<http://localhost:3000/studio>

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