

DAY:☺3

API INTEGRATION AND DATA MIGRATION REPORT- Elite Food-Tuck

API integration means connecting one application with another to share data or use specific functionalities. we build a functional marketplace backend & learn how to utilize APIs for reference, migrate data into Sanity CMS, and ensure compatibility with their templates



Understand how to integrate APIs into your Next.js project.

Learn to migrate data from APIs into Sanity CMS.

Explore how to use existing data from Q-Commerce platforms like Food panda, WordPress, Salesforce, Custom Backend, Sanity.

Develop skills to handle and validate schema with data sources

Q-commerce API Overview

MY API reference from Template 9 Elite Food Tuck. We Are integrating API and migrating data into Sanity CMS to build a functional marketplace backend In Next.js Project. We Use these Provided APIs to populate their Sanity CMS .

Template 09:

<https://github.com/mubashirimtiaz/sanity-nextjs>

External Data Sources: We integrate data from popular Q-Commerce platform and custom back end using their headless API & data export features in sanity

Steps for Day 3:

Provide the API

We have identified the key endpoints:

- **Food Listings** (e.g., /food)
- **Chefs** (e.g., /categories)

Validate and Adjust Our Schema

our existing Sanity CMS schema Day 2 with the API data structure. Adjust field names, types & category (example) API Field: foods-names, Chefs-details

Our schema to match or map fields during migration

Data Migration

Using the Provided API Type [Food, Chef]:

We are manually Import Our sanity's built-in import tools to upload the data & Include validation check & migrate scripts to handle errors gracefully.

Map fields to your Sanity schema API

API Integration in Next.js:

Test API Integration in thunder client & postman

We Render Data in Components & Utility Functions

Output:

1. Sanity CMS populated with imported data from Provided API
2. Functional API integration in Next.js displaying Our Food Categories & Chefs Details.

Screenshots of:

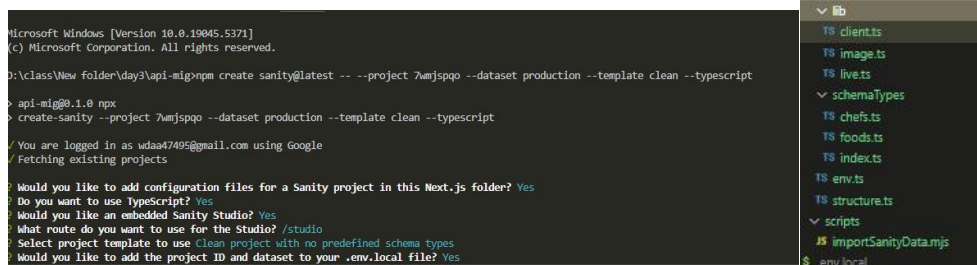
- ♣ API calls.
- ♣ Data successfully displayed in the frontend.
- ♣ Populated Sanity CMS fields.

API INTEGRATION & DATA MIGRATION (Backend)

- Create Sanity Project



- Install Sanity on my Terminal



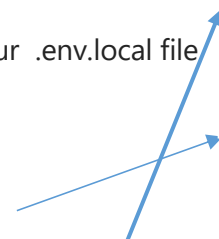
- API Token

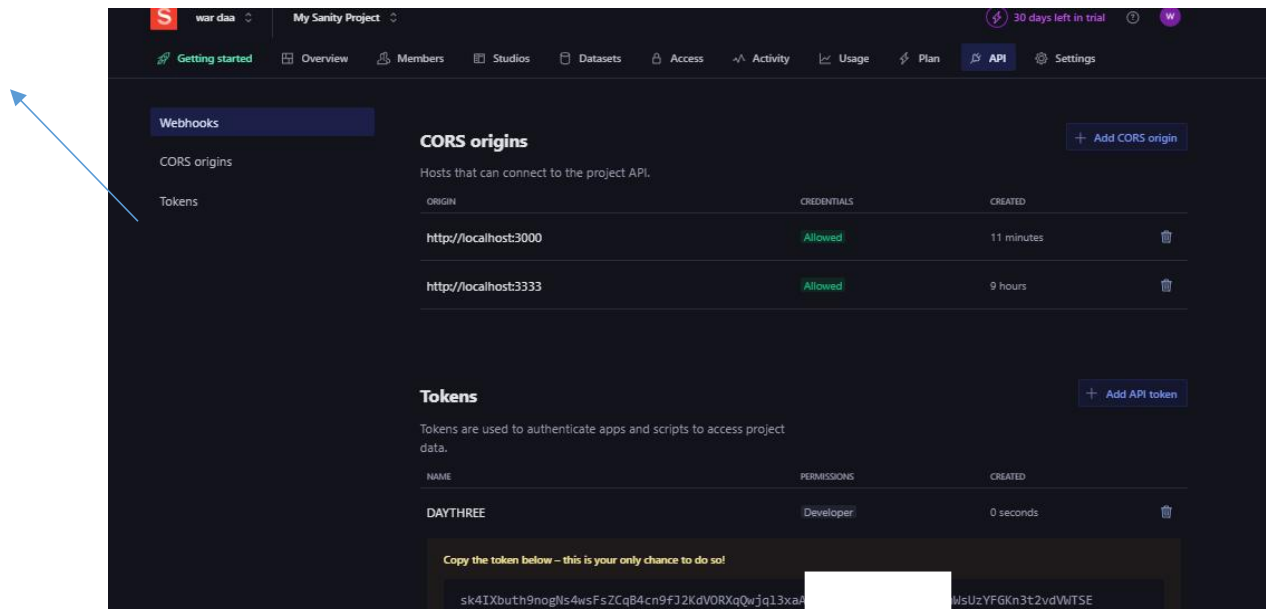
Generate a Sanity API token & Copy the generated token Paste your .env.local file

Token

Add
CORS

API





API
Token

Add our Token in .env.local file

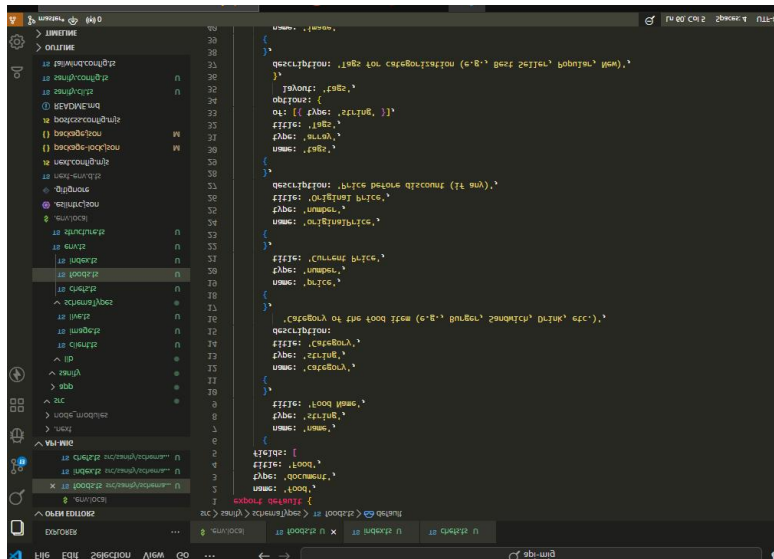
```

$ .env.local x
$ .env.local
1 NEXT_PUBLIC_SANITY_PROJECT_ID="7wmjspoqo"
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 SANITY-API-TOKEN="sk4IXbuth9nogNs4wsFsZCqB4cn9fJ2KdVORXqQwjql3xaA3nxMpijjhPQTuDbc5J3
4 nWsUzYFGKn3t2vdvWTSELSAAi8diJRd24jQPIra3fHxBbhhaGpbDbQJFLPUoN6gMCDHKZ4aqAX6k4ukW48kRSS6gVegm9Kxcakq
5 Anet2tMY8fA7ike"
6
7

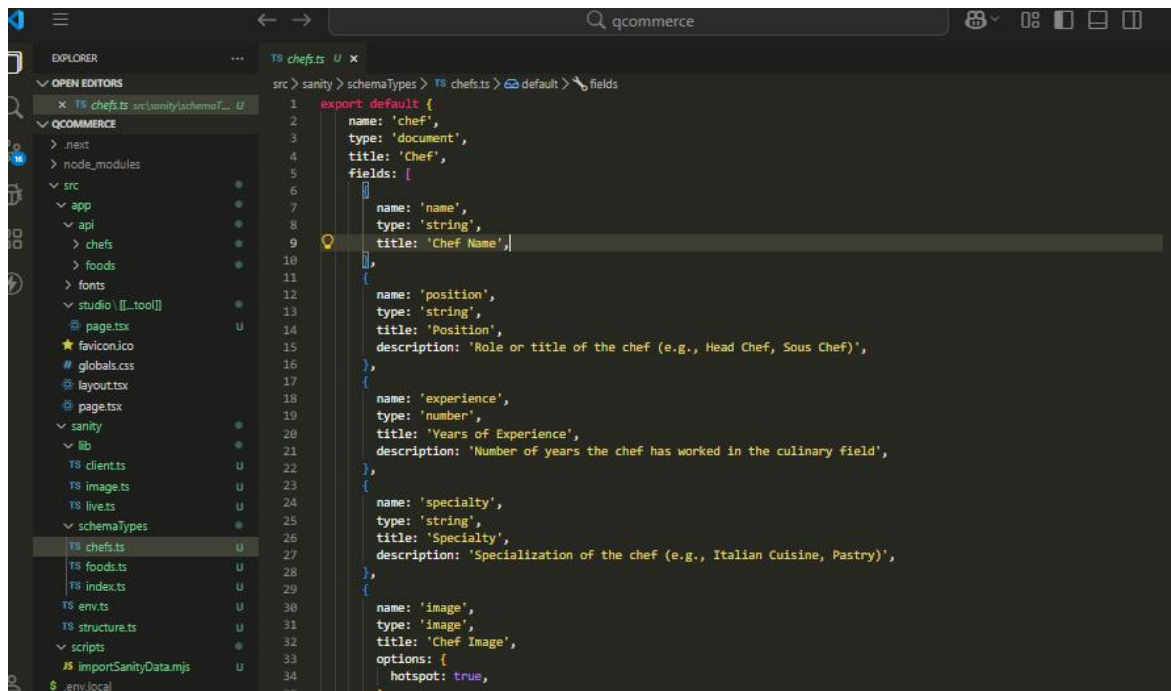
```

Create the sanity schema

Step 1. Create a file called Foods.ts In sanity/schematypes



Step 2. Create a file called Chef.ts In sanity/schemas



Step 3. Add types sanity/schemas/index.ts

```

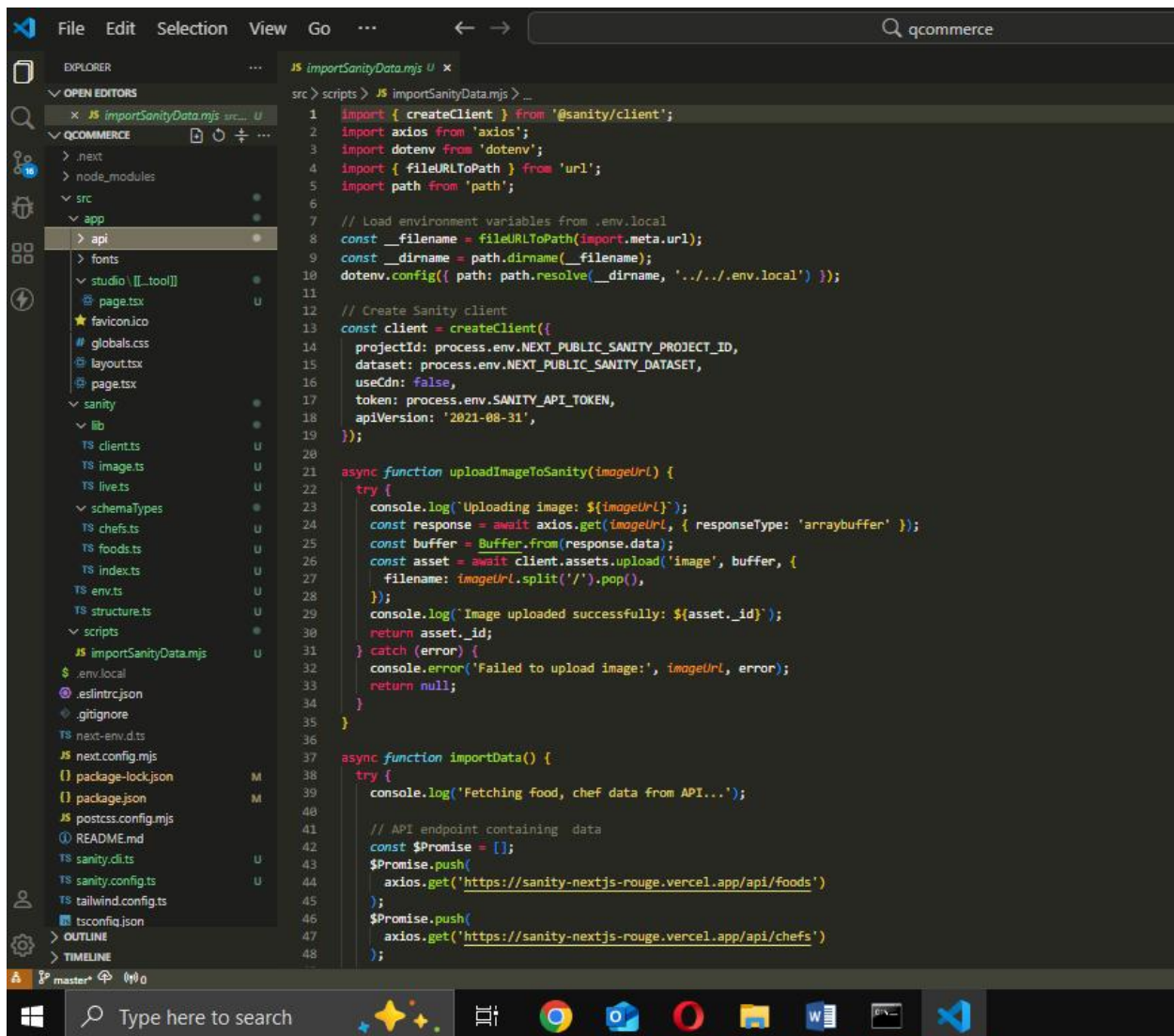
src > sanity > schemaTypes > TS index.ts > schema > types

2  import foods from './foods'
3  import chefs from './chefs'
4
5  export const schema: { types: SchemaTypeDefinition[] } = {
6    types: [foods, chefs],
7  }
8

```

Step 4. Set up the Data import Script & install sanity/client axios

npm install @sanity/client axios dotenv



```

PROBLEMS  COMMENTS  DEBUG CONSOLE  OUTPUT  TERMINAL  PORTS

added 910 packages, and audited 1278 packages in 4m

245 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

added 9 packages, and audited 1287 packages in 14s

245 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

Success! Your Sanity configuration files has been added to this project

C:\ecommerce\qcommerce>npm install @sanity/client axios dotenv

added 12 packages, removed 7 packages, changed 4 packages, and audited 1292 packages

246 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\ecommerce\qcommerce>

```

Step 5. Add a new Import-data script

```

() package.json M x
() package.json > {} scripts > import-data
4   "private": true,
   > Debug
5   "scripts": {
6     "dev": "next dev",
7     "build": "next build",
8     "start": "next start",
9     "lint": "next lint",
10    "import-data": "node src/scripts/importSanityData.mjs"
11  },

```

Step 5. Run Project in terminal


```
Node.js v20.11.1

C:\ecommerce\qcommerce>npm run import-data

> qcommerce@0.1.0 import-data
> node src/scripts/importSanityData.mjs

Fetching food, chef data from API...
Processing food: Fresh Lime
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-1.png
Image uploaded successfully: image-e155a50cd0e02ca81649945bd30a2fc111276-1248x1068-png
Uploading food to Sanity: Fresh Lime
Food uploaded successfully: FoP6QgkAI9c8N6wTb8QZTB
Processing food: Chocolate Muffin
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-2.png
Image uploaded successfully: image-2392122ad8508b5c890aebd8ca36436c214fd7db-1248x1068-png
Uploading food to Sanity: Chocolate Muffin
Food uploaded successfully: Cpl9FWqkp36faTVGL62vg
Processing food: Burger
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-3.png
Image uploaded successfully: image-95a970acfaa0bc5e7df93be9527c2d8a1bc93562-1248x1068-png
Uploading food to Sanity: Burger
Food uploaded successfully: igLuitmzTAn5GgSnt8jb3o
Processing food: Country Burger
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-4.png
Image uploaded successfully: image-ea501a87e900fc084fb7deacca2c3538b167a477-1248x1068-png
Uploading food to Sanity: Country Burger
Food uploaded successfully: Cpl9FWqkp36faTVGL640m
Processing food: Pizza
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-5.png
Image uploaded successfully: image-59499bc14b7749f088d0017b8871e06a47a9deb1-1248x1068-png
Uploading food to Sanity: Pizza
Food uploaded successfully: Cpl9FWqkp36faTVGL65IS
Processing food: Chicken Chup
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-6.png
Image uploaded successfully: image-79c2f2263eb44b0c35a3b7d0f53bd9a34a5700b6-1248x1068-png
Uploading food to Sanity: Chicken Chup
Food uploaded successfully: igLuitmzTAn5GgSnt8jcyW
Processing chef: Tahmina Rumi
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-1.png
Image uploaded successfully: image-62f6ba8dd16f1b27ea593b692ee692bfb8eb860c-1248x1517-png
Uploading chef to Sanity: Tahmina Rumi
Chef uploaded successfully: Cpl9FWqkp36faTVGL66a8
Processing chef: Jorina Begum
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-2.png
Image uploaded successfully: image-a8a4535b34a230733d2ef6eb5c0a4169f65226d5-1248x1517-png
Uploading chef to Sanity: Jorina Begum
Chef uploaded successfully: igLuitmzTAn5GgSnt8jeiu
Processing chef: M. Mohammad
Uploading chef to Sanity: M. Mohammad
Chef uploaded successfully: igLuitmzTAn5GgSnt8jftM
Processing chef: Munna Kathy
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-4.png
```

18-jan-2025

```

OEL \ 390 \ 332
OEL \zmqto 390 \ 352m2
OEL \zmqto 390 \ 8528qms
^ combittq6 \zmqto\[[...foo]] \ 892 (qetq moqntez)
o combittq6 \zmqto\[[...foo]] ...
OEL \ 390 \ 3528qms
OEL \ 390 \ 3528qms (323 moqntez)
^ combittq6 \ 3 a'qz (qetq moqntez)
o combittq6 \ ...
^ qeqqal \ 2'jz
^ z'qalq6...

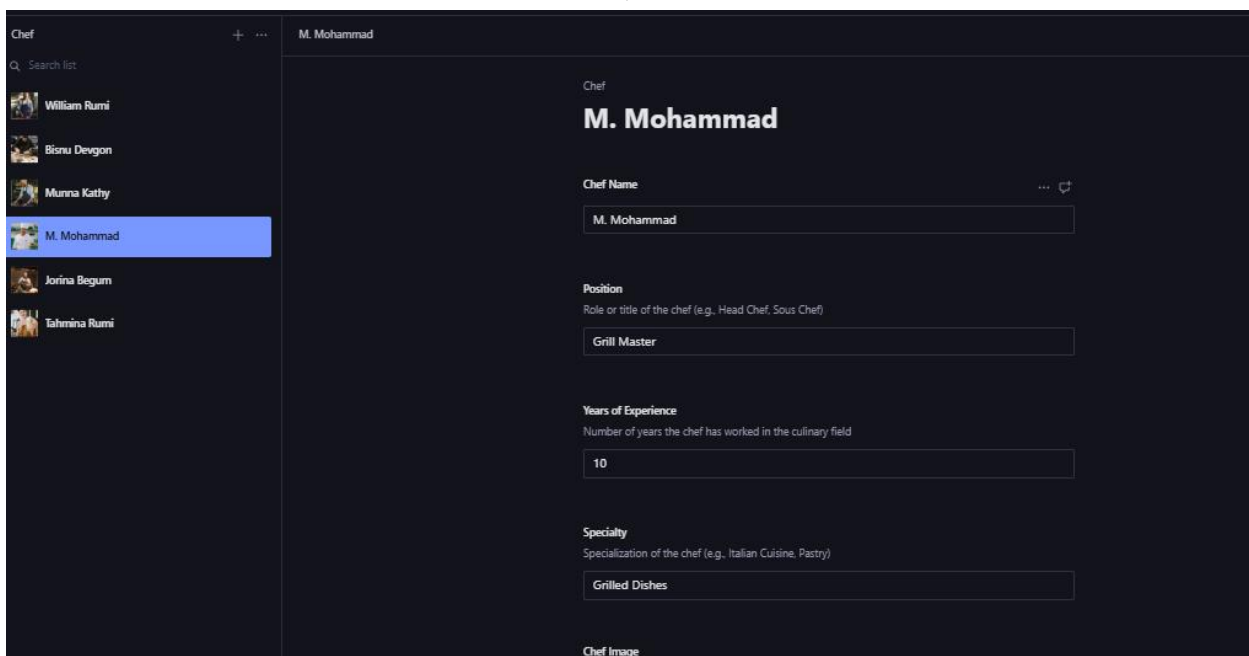
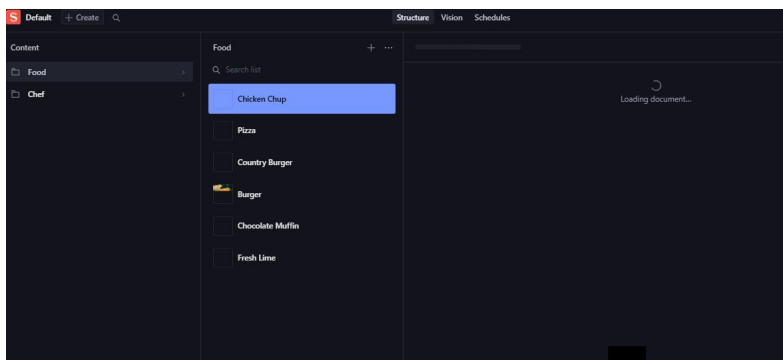
- Evalqomseuz: "evalqocj
- qocaj: mrfb:\\qocajmozf:3999
^ next'jz 3e'3'3z

> next qal
> dcommslc6@J'8 qal

C:/ecommslc6/dcommslc6>ubw lru qal

para twbolc combttq6q zncce2zntjji

```



18-jan-2025

version

Fetch our data On Vession

The screenshot displays the Swagger UI interface for a REST client. The top navigation bar includes tabs for "Default", "Create", and "CL", and a "Structure" tab. The main area is divided into several sections: "DATASET" (production), "API VERSION" (Other), "CUSTOM API VERSION" (v2025-01-21), "RESPONSE" (raw), and "QUERY URL" (https://api.swagger.io/v2/production?query={\"type\": \"fries\"}). The "QUERY" section shows a GET request to the production endpoint. The "RESPONSE" section displays the JSON output, which is a list of food items. The first item is a chocolate muffin, and the second is a country-style burger. The interface also shows the execution time (17ms) and the total number of items (255ms).

DATASET	API VERSION	CUSTOM API VERSION	RESPONSE	QUERY URL (COPY TO CLIPBOARD)
production	Other	v2025-01-21	raw	https://api.swagger.io/v2/production?query={\"type\": \"fries\"}

```

{
  "type": "fries",
  "id": 1,
  "description": "Classic country-style burger served with fries.",
  "updated_at": "2025-01-21T12:49:02Z",
  "name": "Country Burger",
  "image": {
    "type": "image",
    "asset": {
      "type": "reference",
      "ref": "CPL9FWkq36faTVGL6ZrU"
    }
  },
  "category": "Dessert",
  "available": true,
  "type": "food",
  "updated_at": "2025-01-21T12:48:55Z",
  "description": "Soft and rich chocolate muffin topped with chocolate chips.",
  "created_at": "2025-01-21T12:48:55Z",
  "name": "Chocolate Muffin",
  "image": {
    "type": "image",
    "asset": {
      "type": "reference",
      "ref": "CPL9FWkq36faTVGL6ZrU"
    }
  },
  "tags": [
    {
      "type": "tag",
      "name": "Sweet"
    }
  ],
  "price": 28,
  "sell": 0,
  "id": 0
}
  
```

Execution: 17ms End-to-end: 255ms

Save result as JSON

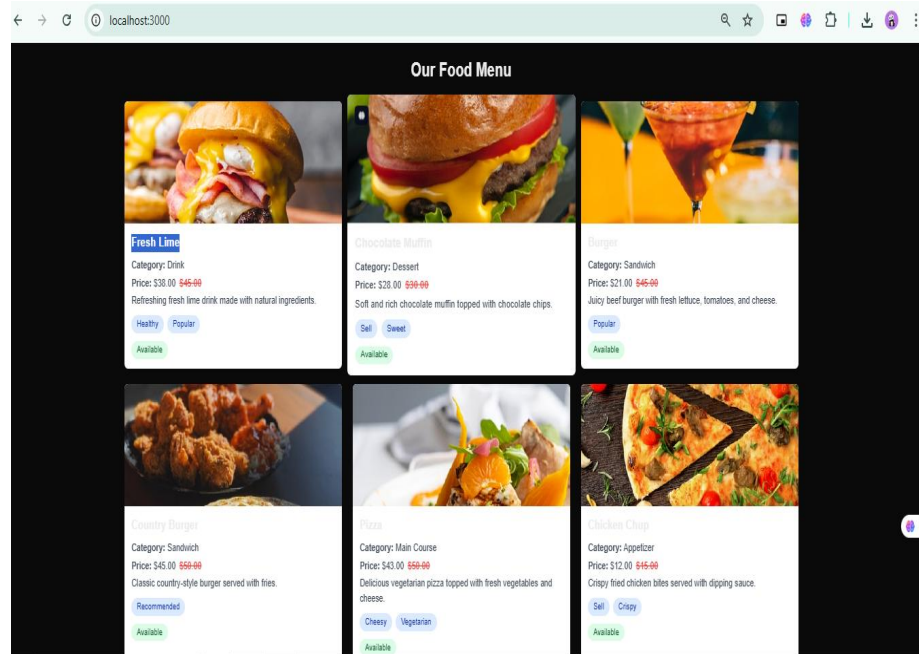
Data successfully displayed in the frontend.

```
src> data > {} foodjson > {} > image
1  [
2
3  {
4    "name": "Fresh Lime",
5    "category": "Drink",
6    "price": 38.0,
7    "originalPrice": 45.0,
8    "tags": ["Healthy", "Popular"],
9    "image": "https://sanity-nextjs-rouge.vercel.app/food/food-1.png",
10   "description": "Refreshing fresh lime drink made with natural ingredients.",
11   "available": true
12 },
13 {
14   "name": "Chocolate Muffin",
15   "category": "Dessert",
16   "price": 28.0,
17   "originalPrice": 30.0,
18   "tags": ["Sell", "Sweet"],
19   "image": "https://sanity-nextjs-rouge.vercel.app/food/food-2.png",
20   "description": "Soft and rich chocolate muffin topped with chocolate chips.",
21   "available": true
22 },
23 {
24   "name": "Burger",
25   "category": "Sandwich",
26   "price": 21.0,
27   "originalPrice": 45.0,
28   "tags": ["Popular"],
29   "image": "https://sanity-nextjs-rouge.vercel.app/food/food-3.png",
30   "description": "Juicy beef burger with fresh lettuce, tomatoes, and cheese.",
31   "available": true
32 },
33 {
34   "name": "Country Burger",
35   "category": "Sandwich",
36   "price": 45.0,
37   "originalPrice": 50.0,
38   "tags": ["Recommended"],
39   "image": "https://sanity-nextjs-rouge.vercel.app/food/food-4.png",
40   "description": "Classic country-style burger served with fries.",
41   "available": true
42 },
43 {
44   "name": "Pizza",
45   "category": "Main Course",
46   "price": 43.0,
47   "originalPrice": 50.0,
48   "tags": ["Cheesy", "Vegetarian"],
49   "image": "https://sanity-nextjs-rouge.vercel.app/food/food-5.png",
50   "description": "Tasty pizza with various toppings and melted cheese.",
51   "available": true
52 }
53 ]
```

18-jan-2025

```
src > app > api > chefs > TS route.ts > ...
1 import { NextResponse } from 'next/server';
2 import data from '@data/chef.json';
3
4 export async function GET() {
5   return NextResponse.json(data);
6 }
7
```

```
c:\> app > Foods > @ pages > @ FoodPage > useeffect callback > @ fetchFoodItems
1 // app/food/page.tsx
2
3 'use client';
4
5 import React, { useEffect, useState } from 'react';
6
7 type FoodItem = {
8   name: string;
9   category: string;
10  price: number;
11  originalPrice: number;
12  tags: string[];
13  image: string;
14  description: string;
15  available: boolean;
16 };
17
18 const FoodPage = () => {
19   const [foodItems, setFoodItems] = useState<FoodItem[]>([]);
20   const [loading, setLoading] = useState<boolean>(true);
21
22   useEffect(() => {
23     const fetchFoodItems = async () => {
24       // API call to fetch the data from the server
25       const response = await fetch('/api/foods'); // Ensure the correct API route is used
26       const data = await response.json();
27       setFoodItems(data);
28     } catch (error) {
29       console.error('Error fetching food data:', error);
30     } finally {
31       setLoading(false);
32     }
33   }, []);
34
35   fetchFoodItems();
36 }, []);
37
38 if (loading) {
39   return <div className="text-center mt-10">loading...</div>;
40 }
41
42 return (
43   <div className="container mx-auto p-6">
44     <h1 className="text-3xl font-bold text-center mb-8">Our Food Menu</h1>
45     <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
46       {foodItems.map((food, index) => (
47         <div key={index}>
48           <img alt={food.image} />
49           <h3>{food.name}</h3>
50           <p>Category: {food.category}</p>
51           <p>Price: {food.price} <del>{food.originalPrice}</del></p>
52           <p>{food.description}</p>
53           <div>
54             {food.tags.map((tag) => <span>{tag}</span>)}
55           </div>
56           <div>
57             {food.available ? <span>Available</span> : <span>Out of Stock</span>}
58           </div>
59         </div>
60       ))}
61     </div>
62   </div>
63 );
64
```



```

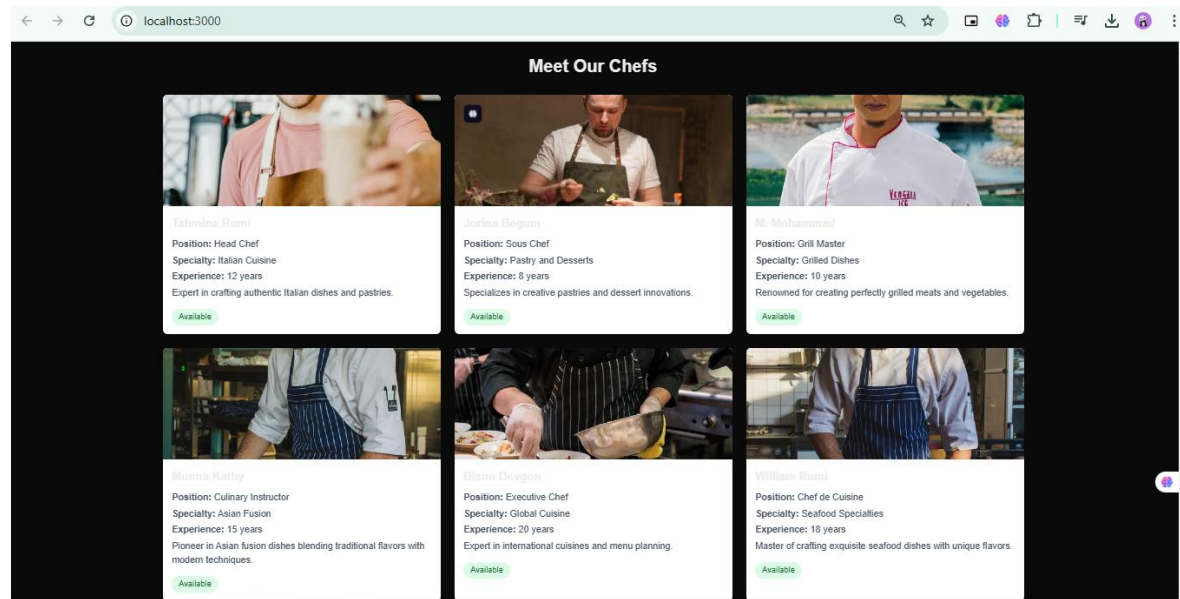
src > app > api > foods > TS route.ts > _
1  import { NextResponse } from 'next/server';
2  import data from '@data/food.json';
3
4  export async function GET() {
5    return NextResponse.json(data);
6  }
7

```

```

@pages.tsx 0 x
src > app > chefs > @pages.tsx > Chef
1  'use client'; // Mark this component as a Client Component
2
3  import React from 'react';
4
5  type Chef = {
6    name: string;
7    position: string;
8    experience: number;
9    specialty: string;
10   image: string;
11   description: string;
12   available: boolean;
13 };
14
15 const chefs: Chef[] = [
16   {
17     name: 'Tahmina Rumi',
18     position: 'Head Chef',
19     experience: 12,
20     specialty: 'Italian Cuisine',
21     image: 'https://sanity-nextjs-rouge.vercel.app/chef/chef-1.png',
22     description: 'Expert in crafting authentic Italian dishes and pastries.',
23     available: true,
24   },
25   {
26     name: 'Jorina Begum',
27     position: 'Sous Chef',
28     experience: 8,
29     specialty: 'Pastry and Desserts',
30     image: 'https://sanity-nextjs-rouge.vercel.app/chef/chef-2.png',
31     description: 'Specializes in creative pastries and dessert innovations.',
32     available: true,
33   },
34   {
35     name: 'M. Mohammad',
36     position: 'Grill Master',
37     experience: 18,
38     specialty: 'Grilled Dishes',
39     image: 'https://sanity-nextjs-rouge.vercel.app/chef/chef-3.png',
40     description: 'Renowned for creating perfectly grilled meats and vegetables.',
41     available: true,
42   },
43   {
44     name: 'Munna Kathy',
45     position: 'Culinary Instructor',
46     experience: 15,
47     specialty: 'Asian Fusion',
48     image: 'https://sanity-nextjs-rouge.vercel.app/chef/chef-4.png',

```



Day 3 Checklist:

Self-Validation Checklist:

API Understanding	✓
Schema Validation	✓
Data Migration	✓
API Integration in Next.js	✓
Submission Preparation:	✓

Hackhton3 Day3 Quarter2

MARKETPLACE API INTEGRATION AND DATA MIGRATION REPORT BY WARDA

Shift: Tuesday 2 to 5 GIAIC

<https://www.linkedin.com>