📅 Day 3 - API Integration and Data Migration for Q-Commerce 🍔 Website

📋 Task Overview

On Day 3, the focus was on integrating APIs and migrating data into Sanity CMS to build a functional backend for the Q-Commerce 🍔 website. This phase replicated real-world practices by utilizing APIs, validating schemas, and managing data effectively. Below is the detailed documentation of the work completed.

🎯 Goals Achieved on Day 3

1️⃣. Understanding the Provided API

Reviewed the provided API documentation and tested the following key endpoints to ensure data availability and compatibility:

🗂️ Key Endpoints Tested:

● Fetch Foods:

○ Endpoint: https://sanity-nextjs-rouge.vercel.app/api/foods

○ Method: GET

Response Example:

```
{

"id": 1,

"name": "🍔 Burger",

"price": 200,

"originalPrice": 250,

"category": "Fast Food",

"tags": ["Best Seller", "Popular"],

"available": true
```

}

○

● **Fetch Chefs:**

○ Endpoint: https://sanity-nextjs-rouge.vercel.app/api/chefs

○ Method: GET

Response Example:

```
{

 "id": 1,

 "name": "👨‍🍳 Chef John",

 "position": "Head Chef",

 "experience": 15,

 "specialty": "Italian Cuisine",

 "available": true

}
```

2️⃣. Validating and Adjusting Schema

To ensure alignment between API data and Sanity CMS, the schema was validated and adjusted as needed. Below are the final schemas:

📄 **Final Food Schema in Sanity:**

```
export default {

 name: 'food',

 type: 'document',

 title: 'Food',

 fields: [

 {
```

```
name: 'name',
type: 'string',
title: 'Food Name',
},
{
name: 'category',
type: 'string',
title: 'Category',
description: 'Category of the food item (e.g., Burger, Sandwich, Drink, etc.)',
},
{
name: 'price',
type: 'number',
title: 'Current Price',
},
{
name: 'originalPrice',
type: 'number',
title: 'Original Price',
description: 'Price before discount (if any)',
},
{
name: 'tags',
type: 'array',
title: 'Tags',
of: [{ type: 'string' }],
options: {
layout: 'tags',
},
description: 'Tags for categorization (e.g., Best Seller, Popular, New)',
},
{
name: 'image',
type: 'image',
title: 'Food Image',
options: {
hotspot: true,
},
},
{
name: 'description',
type: 'text',
title: 'Description',
description: 'Short description of the food item',
},
{
name: 'available',
type: 'boolean',
title: 'Available',
description: 'Availability status of the food item',
 },
 ],
};
```

📄 **Final Chef Schema in Sanity:**

```
export default {
 name: 'chef',
 type: 'document',
 title: 'Chef',
 fields: [
 {
 name: 'name',
 type: 'string',
 title: 'Chef Name',
 },
 {
 name: 'position',
 type: 'string',
 title: 'Position',
 description: 'Role or title of the chef (e.g., Head Chef, Sous Chef)',
 },
 {
 name: 'experience',
 type: 'number',
 title: 'Years of Experience',
 description: 'Number of years the chef has worked in the culinary field',
},
{
name: 'specialty',
type: 'string',
title: 'Specialty',
description: 'Specialization of the chef (e.g., Italian Cuisine, Pastry)',
},
{
name: 'image',
type: 'image',
title: 'Chef Image',
options: {
hotspot: true,
},
},
{
name: 'description',
type: 'text',
title: 'Description',
description: 'Short bio or introduction about the chef',
},
{
name: 'available',
type: 'boolean',
title: 'Currently Active',
 description: 'Availability status of the chef',
 },
 ],
};
```

3️⃣. Data Migration Options Explored

Three methods of migrating data into Sanity CMS were evaluated:

1. Using Scripts to Fetch and Import Data

● Wrote scripts to fetch product data from the API and insert it into Sanity CMS.

2. Manual Data Import

● Exported API data as JSON files and uploaded it manually into Sanity CMS using the Sanity Studio interface.

3. Using External Tools

● Tested importing data via CSV using third-party tools integrated with Sanity CMS.

4️⃣. API Integration in Next.js Frontend

Integrated the API into the Next.js frontend to display dynamic data. Key features include:

🛒 **Food Listing:**

● Dynamically fetched and displayed food data on the homepage using API integration.

Example Code:

```
 export async function getServerSideProps() {
 const foods = await fetch('https://sanity-nextjs-rouge.vercel.app/api/foods').then(res => res.json());
 return { props: { foods } };
}
```

🔍 **Search Functionality:**

- Implemented a search bar to filter foods by name or category.

- Used client-side state management to manage the search results.

🧑‍🍳 **Chef Showcase:**

- Displayed a list of chefs fetched dynamically from the API.

🏁 Final Outcome for Day 3

By the end of Day 3, the following deliverables were completed:

1. Successfully integrated the API into the Next.js frontend.
2. Validated and updated Sanity CMS schemas to align with API data.
3. Migrated food and chef data into Sanity CMS.
4. Implemented dynamic food listing, chef showcase, and search functionality on the frontend.
5. Added error handling for better user experience.

📤 Submission Details

- Document Title: **Day 3 - API Integration and Data Migration for Q-Commerce 🍔 Website**

- Prepared by: **Riaz Hussain**

- Date: **16 January 2025**