

# Day 3 - API Integration Report - Food Marketplace

## 1. Introduction

This report outlines Day 3's progress on the Food Marketplace project by Team 9, focusing on integrating the Sanity CMS API, creating data schemas, and migrating structured data into the local database.

## 2. API Integration

The integration of the Sanity CMS API facilitated seamless communication between the backend and the content management system. Key configuration steps included:

- **API Connection:** Utilized Project ID and API Token to establish connectivity with Sanity CMS.
- **Secure Credential Management:** Employed environment variables to safeguard sensitive credentials.

This integration established a robust foundation for structured data storage, retrieval, and management.

## 3. Schema Creation

Two schemas were developed to organize the data effectively:

### Food Schema (food.ts )

This schema was designed to structure and store details of food items. Key properties include:

- **Name:** Represents the name of the food item.
- **Description:** Provides a detailed overview of the food item.
- **Image:** Stores the URL of the food item's image.
- **Category:** Categorizes the food item (e.g., vegetarian, non-vegetarian).
- **Price:** Specifies the cost of the item.

```
src > data > ts chef.json > ...
1
2
3 {
4   "name": "Fatima Rumi",
5   "position": "Head Chef",
6   "experience": 12,
7   "specialty": "Italian Cuisine",
8   "image": "https://sanity-nextjs-rouge.vercel.app/chef/chef-1.png",
9   "description": "Expert in crafting authentic Italian dishes and pastries.",
10  "available": true
11 }
12
13 {
14   "name": "Zorina Begum",
15   "position": "Sous Chef",
16   "experience": 8,
17   "specialty": "Pastry and Desserts",
18   "image": "https://sanity-nextjs-rouge.vercel.app/chef/chef-2.png",
19   "description": "Specializes in creative pastries and dessert innovations.",
20   "available": true
21 }
22
23 {
24   "name": "Ali Mohammad",
25   "position": "Grill Master",
26   "experience": 10,
27   "specialty": "Grilled Dishes",
28   "image": "https://sanity-nextjs-rouge.vercel.app/chef/chef-3.png",
29   "description": "Renowned for creating perfectly grilled meats and vegetables.",
30   "available": true
31 }
```

### Chef Schema (chef.ts)

This schema was designed to structure and store details of food items. Key properties include:

- **Name:** Full name of the chef (String).

## Chef Schema (chef.ts)

- **Name:** Full name of the chef (String).
- **Bio:** A concise biography or description highlighting the chef's background and expertise (Text).
- **Photo:** A high-quality image of the chef (Image).

```
src > sanity > schemaTypes > TS chefs.ts > default
1 export default {
2   name: 'chef',
3   type: 'document',
4   title: 'Chef',
5   fields: [
6     {
7       name: 'name',
8       type: 'string',
9       title: 'Chef Name',
10    },
11    {
12      name: 'position',
13      type: 'string',
14      title: 'Position',
15      description: 'Role or title of the chef (e.g., Head Chef, Sous Chef)',
16    },
17    {
18      name: 'experience',
19      type: 'number',
20      title: 'Years of Experience',
21      description: 'Number of years the chef has worked in the culinary field',
22    },
23    {
24      name: 'specialty',
25      type: 'string',
26      title: 'Specialty',
27      description: 'Specialization of the chef (e.g., Italian Cuisine, Pastry)',
28    },
29  ],
30 }
```

## 4. Data Migration

The data migration process encompassed the transfer of food and chef-related data from Sanity CMS to the local application. The following steps were involved:

### Script Setup

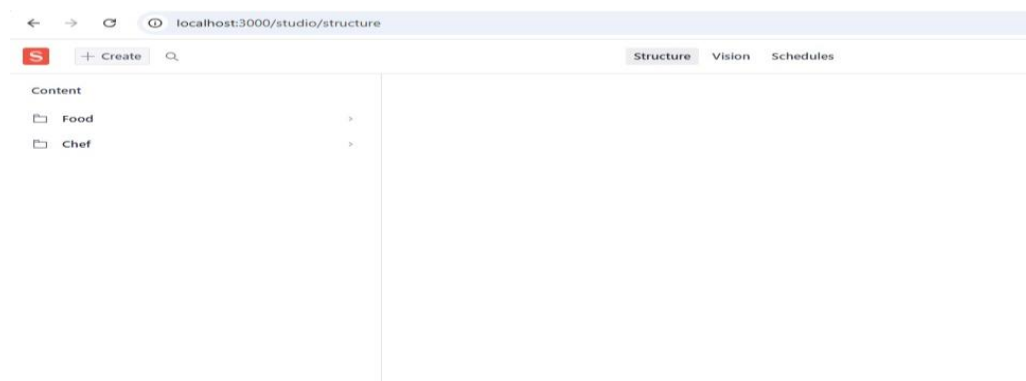
A "scripts" directory was established at the root of the project.

A migration script, **importData.mjs**, was developed to facilitate the data transfer.

### Execution:

The migration process was initiated by running the following command in the terminal:

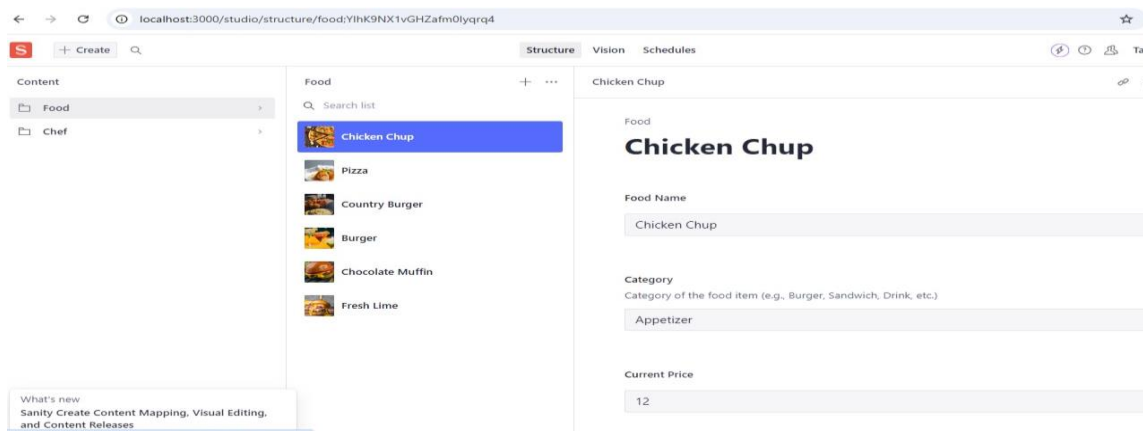
```
npm run importData
```



## Verification

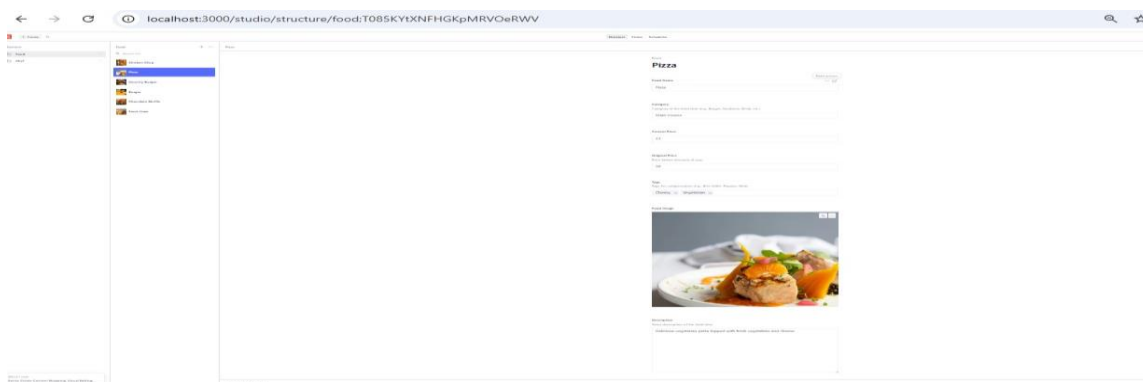
Following the migration, the success was validated through the following steps:

- The project was executed locally, and the /studio/structure endpoint was accessed.
- The same Sanity CMS account was logged in to verify the presence of the migrated data.



## 5. Migration Script

The following code snippet was utilized in the **importData.mjs** script to facilitate the data migration.



## 6. Conclusion:

On Day 3, the Sanity CMS API was successfully integrated, and data migration for the Food Marketplace project was completed. Key achievements include developing food and chef schemas, automating migration with the **importData.mjs** script, and verifying data integrity via Sanity Studio. This sets a strong backend foundation for future frontend integration and user interaction.