

Day 3:

API Integration and Data Migration – FoodTuck

Prepared by: Muhammad Hassan

Objective

The goal for Day 3 was to integrate API data into Sanity CMS for the Culinary Marketplace project. This integration aimed to enable dynamic content updates for chefs and food items in the marketplace, streamlining data management and ensuring scalability.

1. Sanity CMS Schema Design

To efficiently handle data for chefs and food items, I designed two schemas in Sanity CMS with the following fields:

Food Schema

- - name: Food name (string).
- - category: Food category (string).
- - price: Current price (number).
- - originalPrice: Original price before discount (number).
- - tags: Tags for categorization (array of strings).
- - image: Food image (image).
- - description: Short description of the food item (text).
- - available: Availability status (boolean).

Chef Schema

- - name: Chef name (string).
- - position: Role or title of the chef (string).
- - experience: Years of experience (number).
- - specialty: Specialization of the chef (string).
- - image: Chef image (image).
- - description: Short bio or introduction (text).
- - available: Currently active status (boolean).

2. API Integration and Data Migration

Food Data Integration

```
4
5   export const getData = async () => {
6     const data = await client.fetch(
7       `*[_type == "food"]{
8         name,
9         category,
10        price,
11        originalPrice,
12        tags,
13        "imageUrl": image.asset->url,
14        description,
15        available
16      }`
17    );
18    return data;
19  };
20
```

Fetches food data using GROQ queries to retrieve fields such as name, category, price, originalPrice, tags, image, description, and availability. Data was dynamically mapped to the frontend.

Chef Data Integration

```
export const getData = async () => {
  const data = await client.fetch(`*[_type == "chef"] | order(_createdAt desc)[0...4]{
    name,
    position,
    "imageUrl": image.asset->url,
    description,
    available
  }`
  );
  return data
};
```

Fetches chef data using GROQ queries to retrieve fields such as name, position, experience, specialty, image, description, and availability. Ensured proper alignment with Sanity CMS schema.

3. Steps Taken for Data Migration

1. 1. Validated API Endpoints:

- Tested endpoints for food and chef data using Postman to confirm response structures.

2. 2. Adjusted Schemas:

- Ensured compatibility between Sanity CMS schema and API fields.

3. 3. Populated Data:

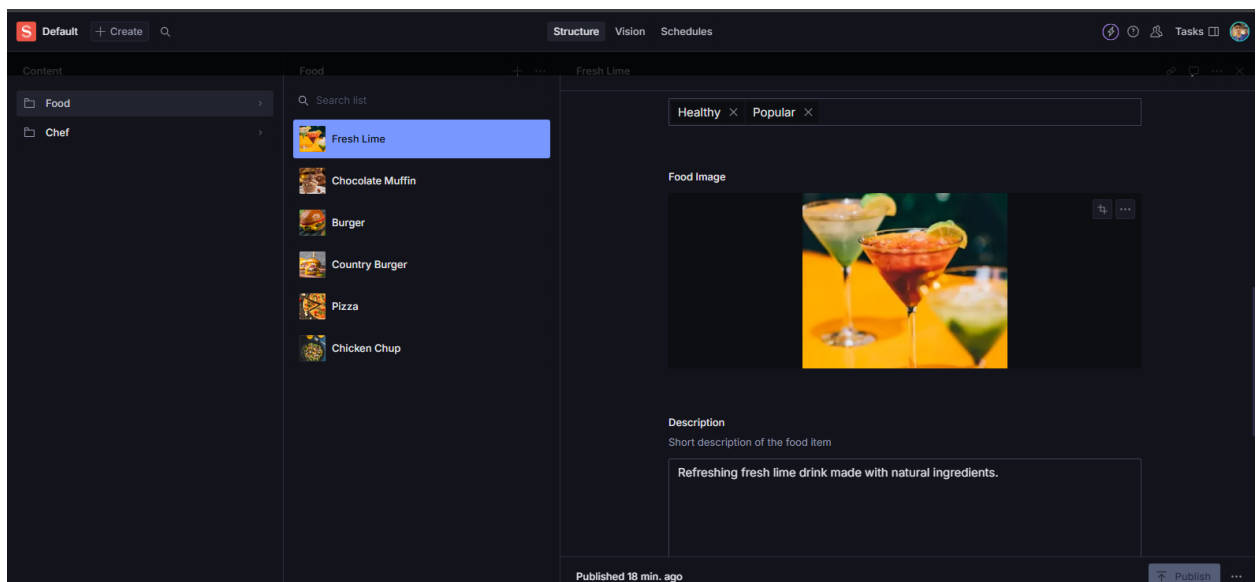
- Imported food and chef data into Sanity CMS using scripts for efficiency.

4. Tools Used

- - Sanity Studio: For schema design and content management.
- - Sanity CLI: For data import and export.

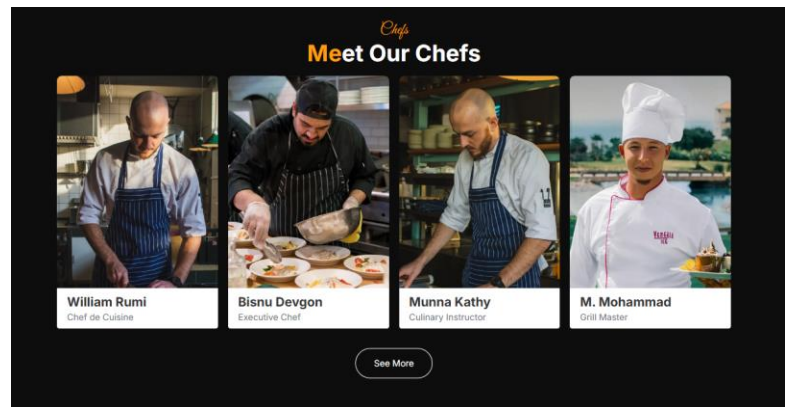
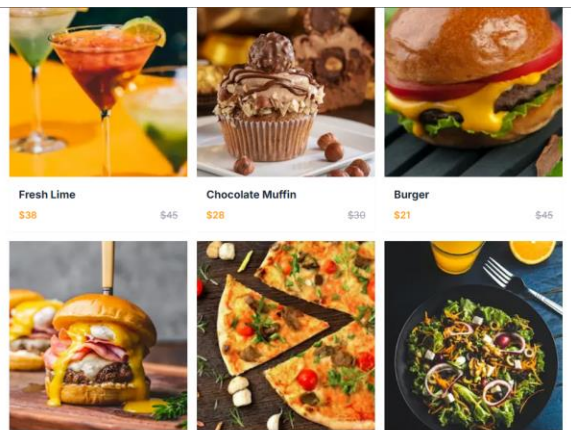
5. Screenshots and Frontend Display

4. 1. Sanity CMS Fields:



- Screenshot showing populated food and chef fields in Sanity Studio.

5. 2. Frontend Display:



- Screenshot showing dynamically displayed food items and chefs.

Conclusion

The API integration and data migration process was successfully completed for the Marketplace project. This integration streamlined data management for chefs and food items, ensuring the marketplace is dynamic, scalable, and easier to maintain.
