# 1)API Integration Process

## Objective:

The goal of the API integration was to enable seamless interaction between the Q-Commerce platform and the external data sources, ensuring that food items and other details are dynamically fetched and displayed.

## Steps Involved:

1. **Setting Up Sanity Client**
    - Configuration file (`client.js`) was set up with project ID, dataset, and API version to authenticate requests.

```
export const client = createClient({
  projectId,
  dataset,
  apiVersion,
  useCdn: true,
});
```

2. **Creating Queries for Data Fetching:**
    - Defined queries to fetch relevant data like food items and categories from the Sanity CMS using GROQ (Sanity's query language).

```
const query = '*[_type == "foodItem"]';
const data = await client.fetch(query);
```

3. **Handling Data Fetching in React Components:**
    - Integrated the API data fetching logic into the React components, using `useEffect` and `useState` hooks to load and manage state.

```
useEffect(() => {
  const fetchData = async () => {
    const foodData = await client.fetch(query);
    setFoods(foodData);
  };
  fetchData();
```

4. **Image Handling:**
    - Integrated Sanity's image URL builder to display images fetched from the CMS in the front-end.
    -

```
import imageUrlBuilder from '@sanity/image-url';

const builder = imageUrlBuilder(client);
```

5. `const urlFor = (source) => builder.image(source).url();`

# 2) Adjustments Made to Schemas:

## Objective:

To ensure that the data model fits the needs of the Q-Commerce platform, adjustments were made to the Sanity schemas to include food items, categories, and associated image fields.

## Steps Taken:

1. **Creating/Modifying the Food Item Schema:**
    - The schema for food items was adjusted to include fields such as `name`, `price`, `description`, `category`, and `image`.

```
export default {

  name: 'foodItem',
  title: 'Food Item',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Food Name',
      type: 'string',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'reference',
      to: [{ type: 'category' }],
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
      options: {
        hotspot: true,
      },
    },
  ],};
```

# 3) Migration Steps and Tools Used:

## Objective:

To migrate food items, categories, and images from the existing data structure into Sanity CMS, ensuring the integrity and availability of data for the front-end.

## Steps Taken:

1. **Exporting Data from Existing System:**

    - The existing data was exported into a structured format (e.g., JSON, CSV, or other relevant formats).

2. **Mapping Data to Sanity Schema:**
    - The exported data was mapped to fit the Sanity CMS schema, ensuring the correct fields (e.g., food items, prices, categories) were matched with the appropriate schema fields.

3. **Using Sanity's Import Tool:**
    - Data was imported into Sanity CMS using the sanity dataset import command, which allows for the bulk import of data into the defined schemas.

# Conclusion:

The integration of Sanity CMS with the Q-Commerce food restaurant platform has been successfully completed. The following objectives were met:

- API integration was completed using next-sanity, allowing dynamic fetching of food data and images.

- Sanity schemas were adjusted to include necessary fields for food items and categories.

- Data was migrated into Sanity CMS using structured formats and the import tools provided by Sanity.