

SANITY MIGRATION API DAY 03:

Prerequisites

Before starting, ensure you have:

1. **Sanity CLI** installed.
 2. **A Sanity project** set up (with a valid projectId and dataset).
 3. **MockAPI endpoint** ready for use.
 4. **Node.js** and **npm/yarn** installed on your system.
-

Step 1:

1: Install Required Dependencies

Begin by installing the required libraries in your Next.js project:

Step 2: Set Up Environment Variables

Create a .env file in the root of your project to store sensitive information like your Sanity project ID and dataset.

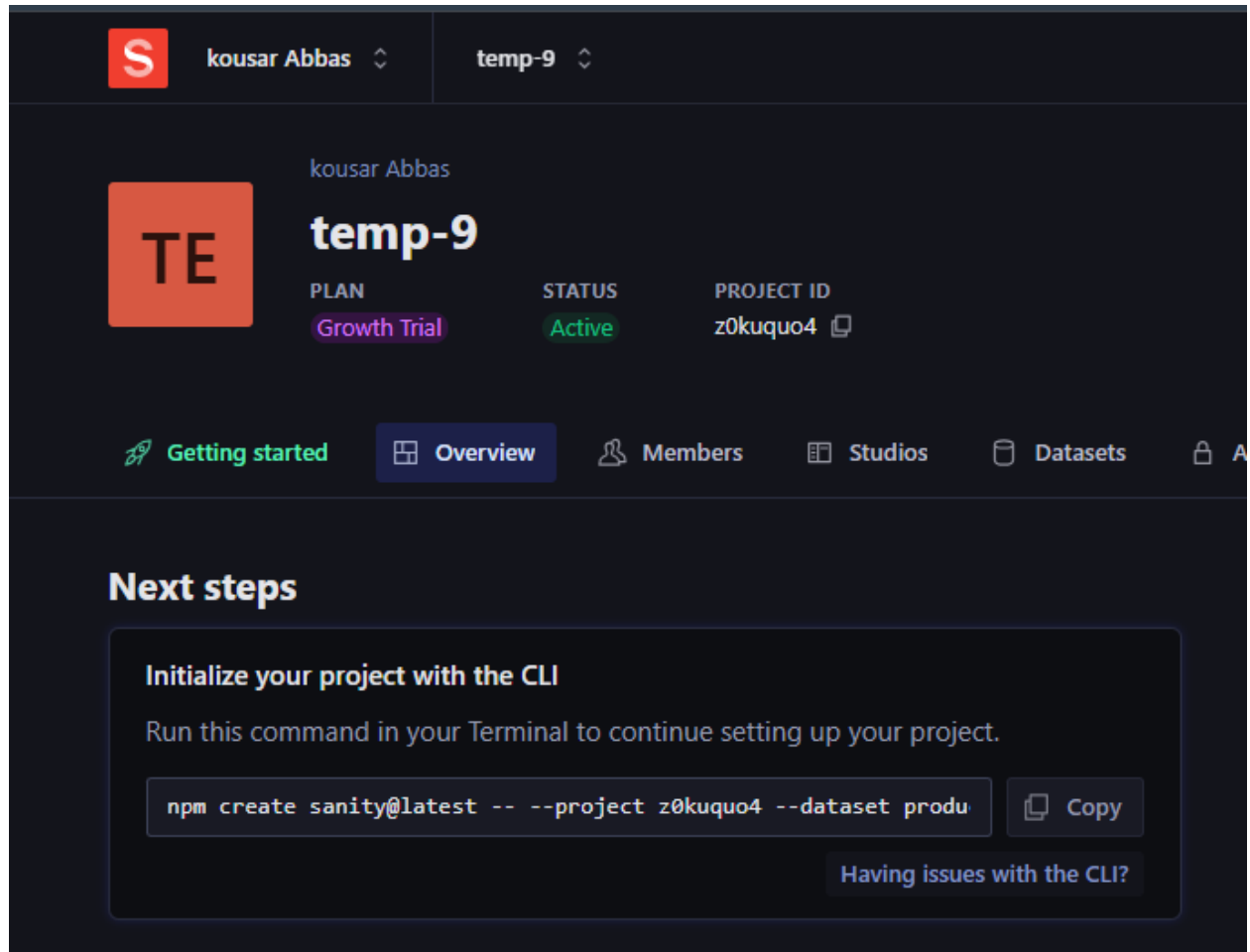
```
$ .env.local
1 NEXT_PUBLIC_SANITY_PROJECT_ID="z0kuquo4"
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 SANITY_API_TOKEN="skSICJTachDUaKZdvsap1FuU8Lv7imzJW611NkUx2Hunxfen8rLDz4Hpm3dzPE1bf5qRU14Ndm2wBXa5Te2fHy8DYBqwtH1y2PGecaE5
  RE4Cy7f1ymw5hbKGohCQ8oP6dNE8Fv43rbIaXB0nKCZfq8c0BJFab0dojxwu2mK9C7h0GXExx0UX"
```

2: Fetching Sanity Project ID and API Token

Project ID

To find your Sanity project ID:

1. Log in to your Sanity account at <https://www.sanity.io/manage>
2. Select your project
3. In the project dashboard, you'll see the project ID listed



API Token

To generate a Sanity API token:

1. Go to <https://www.sanity.io/manage> and select your project
2. Navigate to the "API" tab
3. Under "Tokens," click "Add API token"
4. Give your token a name and select the appropriate permissions (usually "Editor" for full read/write access)
5. Copy the generated token

The screenshot shows the Sanity.io API management interface. The top navigation bar includes the Sanity logo, user name 'kousar Abbas', project name 'temp-9', and a trial status '30 days left in trial'. The main navigation menu has options: Getting started, Overview, Members, Studios, Datasets, Access, Activity, Usage, Plan, API (selected), and Settings.

On the left sidebar, there are links for Webhooks, CORS origins, and Tokens. The main content area is divided into two sections:

CORS origins

Hosts that can connect to the project API.

ORIGIN	CREDENTIALS	CREATED	
http://localhost:3000	Allowed	15 hours	
http://localhost:3333	Allowed	15 hours	

[+ Add CORS origin](#)

Tokens

Tokens are used to authenticate apps and scripts to access project data.

NAME	PERMISSIONS	CREATED	
food	Editor	14 hours	

[+ Add API token](#)

What's new
Sanity Create Content Mapping, Visual Editing, and Content Releases

3. Creating the Sanity Schema

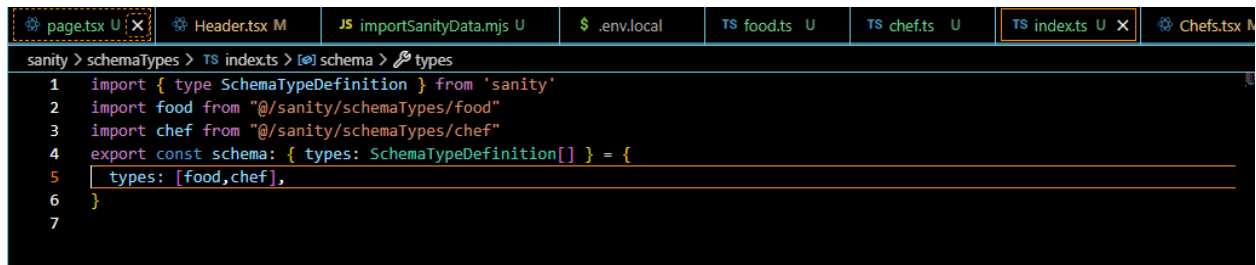
Now, let's create a schema for our website. In your Sanity schema folder (usually `sanity/schemaTypes`), create a new file called `food.ts`:

```
1 export default {
2   name: 'food',
3   type: 'document',
4   title: 'Food',
5   fields: [
6     {
7       name: 'name',
8       type: 'string',
9       title: 'Food Name',
10    },
11    {
12      name: 'category',
13      type: 'string',
14      title: 'Category',
15      description:
16        | 'Category of the food item (e.g., Burger, Sandwich, Drink, etc.)',
17    },
18    {
19      name: 'price',
20      type: 'number',
21      title: 'Current Price',
22    },
23    {
24      name: 'originalPrice',
25      type: 'number',
26      title: 'Original Price',
27      description: 'Price before discount (if any)',
28    },
29    {
30      name: 'tags',
31      type: 'array',
32      title: 'Tags',
33      of: [{ type: 'string' }],
34      options: {
35        | layout: 'tags',
36      },
37      description: 'Tags for categorization (e.g., Best Seller, Popular, New)',
38    },
39    {
40      name: 'image',
41      type: 'image',
42      title: 'Food Image',
43      options: {
44        | hotspot: true,
45      },
46    },
47    {
48      name: 'description',
```

Now, let's create a schema for our website. In your Sanity schema folder (usually `sanity/schemaTypes`), create a new file called `chef.ts`:

```
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      name: 'image',
31      type: 'image',
32      title: 'Chef Image',
33      options: {
34        | hotspot: true,
35      },
36    },
37    {
38      name: 'description',
39      type: 'text',
40      title: 'Description',
41      description: 'Short bio or introduction about the chef',
42    },
43    {
44      name: 'available',
45      type: 'boolean',
46      title: 'Currently Active',
47      description: 'Availability status of the chef',
48    },
49  ],
50 }
```

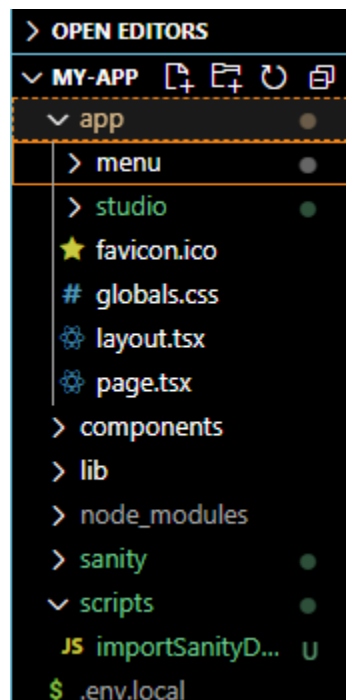
Then, update your `sanity/schemaTypes/index.ts` file to include the new product schema:



The screenshot shows the VS Code editor with the file `sanity > schemaTypes > TS index.ts` open. The code is as follows:

```
1 import { type SchemaTypeDefinition } from 'sanity'
2 import food from "@sanity/schemaTypes/food"
3 import chef from "@sanity/schemaTypes/chef"
4 export const schema: { types: SchemaTypeDefinition[] } = {
5   types: [food, chef],
6 }
7
```

Setting Up the Data Import Script



Now, let's create a script to import data from an external API into Sanity. Create a new file `scripts/importSanityData.mjs` in your project root:

Create the Import Script

Create a new directory named scripts in your project root and add a file called importData.js.

```
scripts > # importSanityData.mjs > @sanity/client > token
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import dotenv from 'dotenv';
4 import { fileURLToPath } from 'url';
5 import path from 'path';
6
7 // Load environment variables from .env.local
8 const __filename = fileURLToPath(import.meta.url);
9 const __dirname = path.dirname(__filename);
10 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12 // Create Sanity client
13 const client = createClient({
14   projectId: 'z8kuquod',
15   dataset: 'production',
16   useCdn: false,
17   token: 'sk5iCITachOUakZdvsaP1FuU8Lv7Imz3w611MduX2Hunxfen8rLDz4Hpm3dzPE1bf5q8Uj4Ndm2wKa5Te2fHy8DYBqWthly2PGeCaESRE4Cy7FlywShbKgoHCQ8oP6dNEBFV43rbIaxB0nxCZfq8c883Fab0dojxwu2mk9C7h8GXEeX8BUK',
18   apiVersion: '2021-08-31',
19 });
20
21 async function uploadImageToSanity(imageUrl) {
22   try {
23     console.log('Uploading image: ${imageUrl}');
24     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
25     const buffer = Buffer.from(response.data);
26     const asset = await client.assets.upload('image', buffer, {
27       filename: imageUrl.split('/').pop(),
28     });
29     console.log('Image uploaded successfully: ${asset._id}');
30     return asset._id;
31   } catch (error) {
32     console.error('Failed to upload image:', imageUrl, error);
33     return null;
34   }
35 }
36
37 async function importData() {
38   try {
39     console.log('Fetching food, chef data from API...');
40
41     // API endpoint containing data
42     const $Promise = [];
43     $Promise.push(
44       axios.get('https://sanity-nextjs-rouge.vercel.app/api/foods')
45     );
46     $Promise.push(
47       axios.get('https://sanity-nextjs-rouge.vercel.app/api/chefs')
48     );
49   }
50 }
```

Now, let's install the necessary packages. Run the following command in your terminal:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\Kousar Chrome Dow\FOOD RESTURANT\my-app> npm install @sanity/client axios dotenv

up to date, audited 1294 packages in 11s

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

1 moderate severity vulnerability

To address all issues, run:
  npm audit fix --force

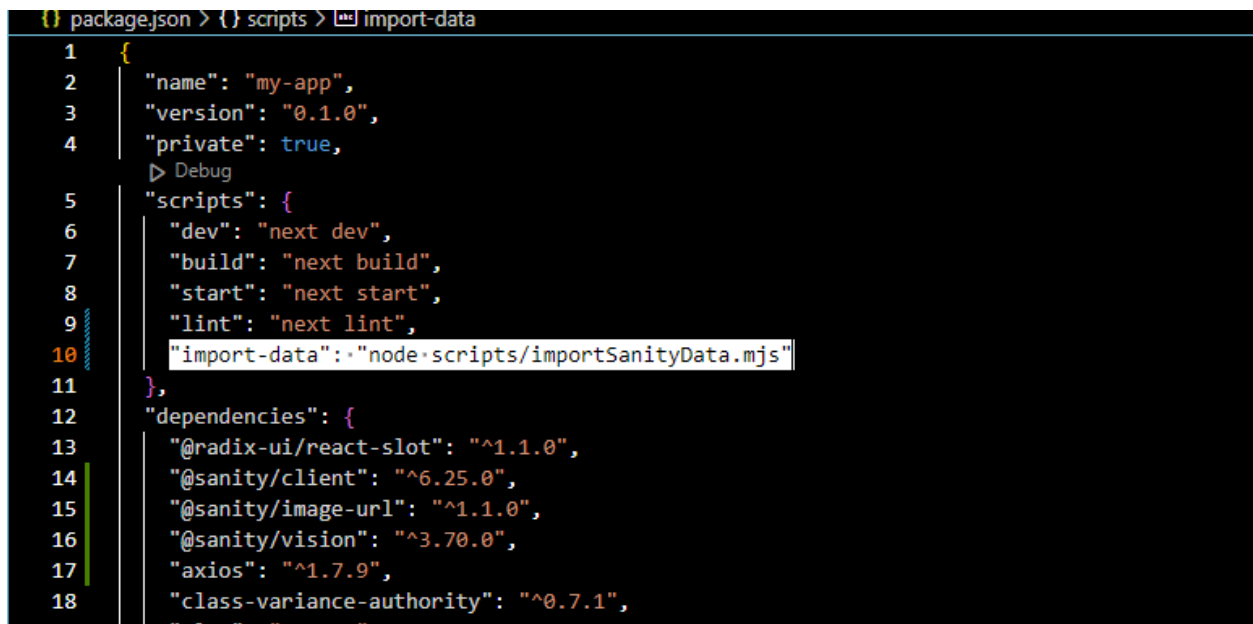
Run `npm audit` for details.
PS E:\Kousar Chrome Dow\FOOD RESTURANT\my-app>
```

Explanation:

1. **Environment Variables:** The script uses the .env file for configuration.
2. **Fetch Data:** Data is retrieved from MockAPI using node-fetch.

3. **Sanity Client:** The client connects to your Sanity project to push data.
4. **Schema Mapping:** Map fields from MockAPI data to your Sanity schema.
5. **Unique IDs:** Use unique IDs to prevent duplicate entries.

Running the Import Script



```
{ } package.json > { } scripts > import-data
1 {
2   "name": "my-app",
3   "version": "0.1.0",
4   "private": true,
5   "scripts": {
6     "dev": "next dev",
7     "build": "next build",
8     "start": "next start",
9     "lint": "next lint",
10    "import-data": "node scripts/importSanityData.mjs"
11  },
12  "dependencies": {
13    "@radix-ui/react-slot": "^1.1.0",
14    "@sanity/client": "^6.25.0",
15    "@sanity/image-url": "^1.1.0",
16    "@sanity/vision": "^3.70.0",
17    "axios": "^1.7.9",
18    "class-variance-authority": "^0.7.1",
19    "clsx": "^2.1.1",
20    "cva": "^1.0.0",
21    "date-fns": "^3.6.0",
22    "date-fns-tz": "^3.1.3",
23    "embla-carousel-react": "^8.1.5",
24    "embla-carousel-svelte": "^8.1.5",
25    "framer-motion": "^11.0.0",
26    "geist": "^1.3.1",
27    "lucide-react": "^0.399.0",
28    "next": "14.2.5",
29    "next-auth": "4.24.7",
30    "next-themes": "^0.3.0",
31    "openai": "^4.61.1",
32    "react": "18.3.1",
33    "react-dom": "18.3.1",
34    "react-hook-form": "^7.51.5",
35    "react-portal": "^4.2.0",
36    "react-use": "^17.5.0",
37    "recharts": "^2.12.3",
38    "svelte": "4.2.19",
39    "svelte-hmr": "^0.15.3",
40    "tailwind-merge": "^2.3.0",
41    "tailwind-variants": "^0.2.1",
42    "tailwindcss": "3.4.1",
43    "typescript": "5.4.5",
44    "use-debounce": "^9.0.4",
45    "usehooks-ts": "^3.1.0",
46    "zod": "^3.23.8"
47  }
48 }
```

To run the import script, we need to add a new script to our `package.json` file. Open your `package.json` and add the following to the `"scripts"` section:

Now you can run the import script using:

```
npm run import-data
```

```

PS E:\Kousar\Chrome Dow\Food RESTURANT\my-app> npm run import-data

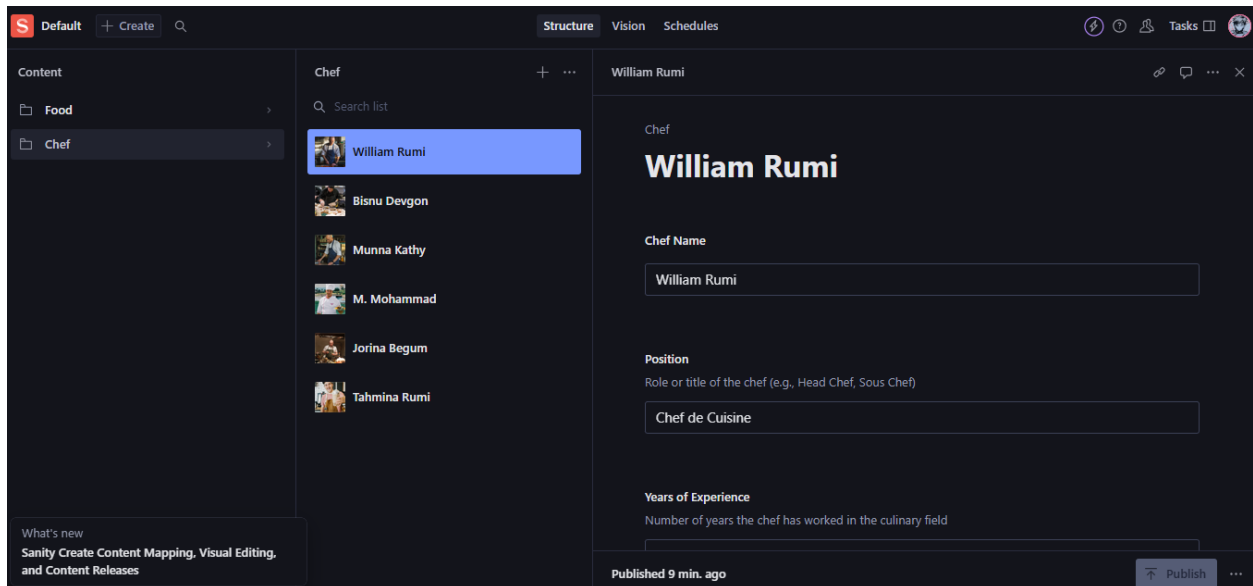
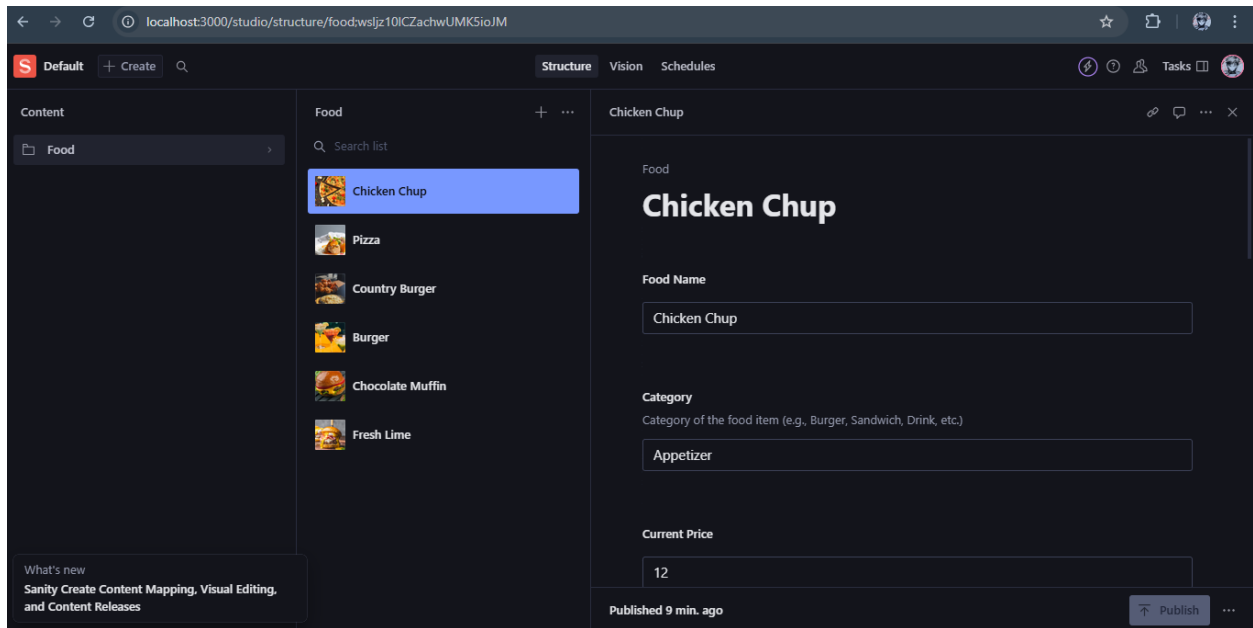
> my-app@0.1.0 import-data
> node scripts/importSanityData.mjs

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-e155a50cdbcd0e02ca81649945bd30a2fc111276-1248x1068-png
Uploading food to Sanity: Fresh Lime
Food uploaded successfully: IZ4ZA8avncRcTDvyuc95C9
Processing food: Chocolate Muffin
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-2.png
Image uploaded successfully: image-2392122ad0508b5c890aebd8ca36436c214fd7db-1248x1068-png
Uploading food to Sanity: Chocolate Muffin
Food uploaded successfully: IZ4ZA8avncRcTDvyuc964a
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: IZ4ZA8avncRcTDvyuc96FT
Processing food: Country Burger
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-4.png
Image uploaded successfully: image-ea501a87e980fc084fb7deacca2c3538b167a477-1248x1068-png
Uploading food to Sanity: Country Burger
Food uploaded successfully: IZ4ZA8avncRcTDvyuc97FA
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: wsljz101CZachwUMK5inhM
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: wsljz101CZachwUMK5ioJM
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: IZ4ZA8avncRcTDvyuc99PR


```

Step 5: Verify the Data in Sanity Studio

1. Open your Sanity Studio.
 2. Navigate to the relevant schema type (e.g., yourSchemaType).
 3. Confirm the imported data appears as expected.
-



FOR CHEFS SECTION:



Vegetable Curry

\$11.49


BBQ Ribs


\$17.99


Prawn Skewers


\$14.99

Meet Our Chefs









William Rumi


Chef de Cuisine

Experience: 18 years

Specialty: Seafood Specialties

Master of crafting exquisite seafood dishes with unique flavors.

Currently Active



M. Mohammad


Grill Master

Experience: 10 years

Specialty: Grilled Dishes

Renowned for creating perfectly grilled meats and vegetables.

Currently Active



Munna Kathy

Culinary Instructor

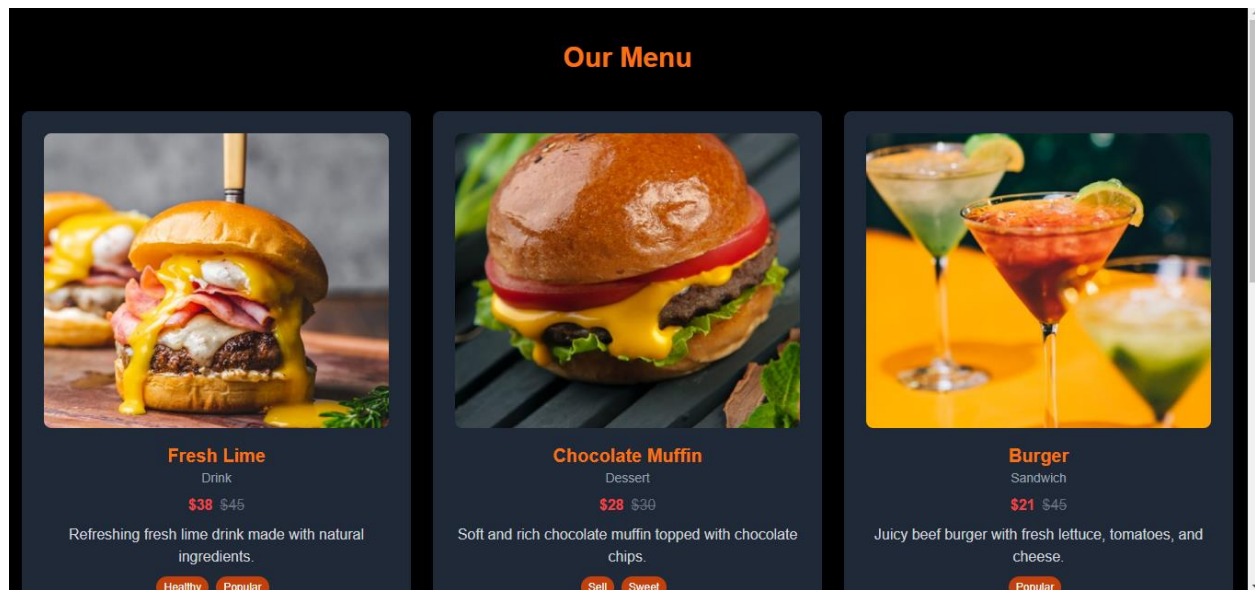
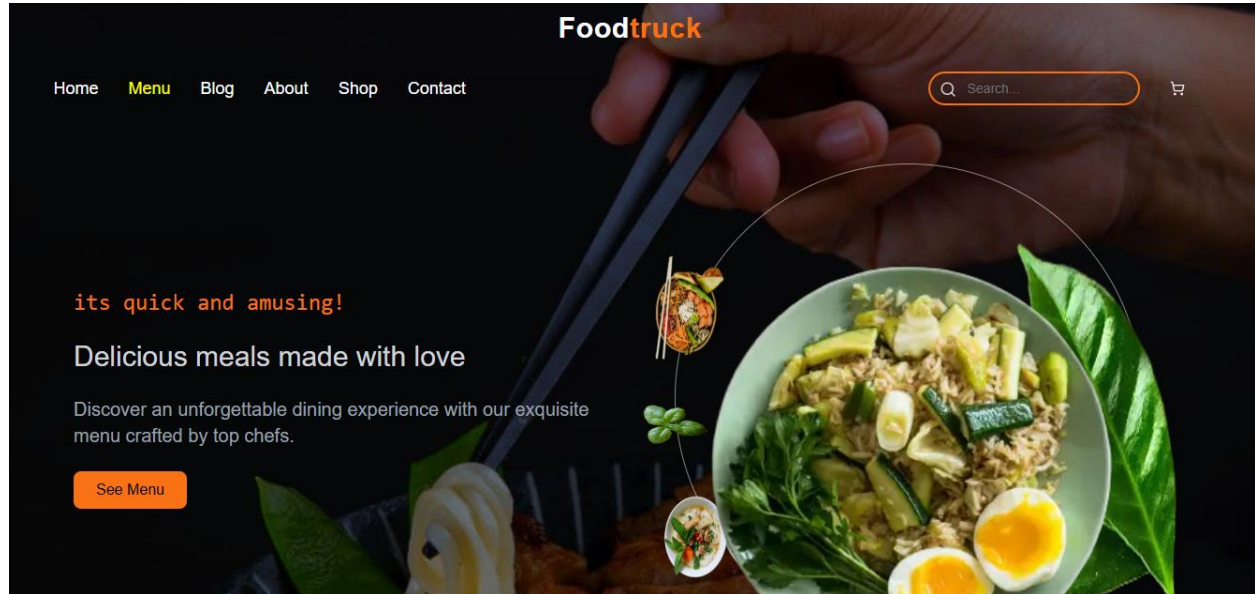
Experience: 15 years

Specialty: Asian Fusion

Pioneer in Asian fusion dishes blending traditional flavors with modern techniques.

Currently Active

FOR MENU PAGE:





Country Burger

Sandwich

\$45 \$50

Classic country-style burger served with fries.

Recommended

Available



Pizza

Main Course

\$43 \$50

Delicious vegetarian pizza topped with fresh vegetables and cheese.

Cheesy

Vegetarian

Available



Chicken Chup

Appetizer

\$12 \$15

Crispy fried chicken bites served with dipping sauce.

Sell

Crispy

Available