

# Hackathon day 3

## API INTEGRATION AND DATA MIGRATION

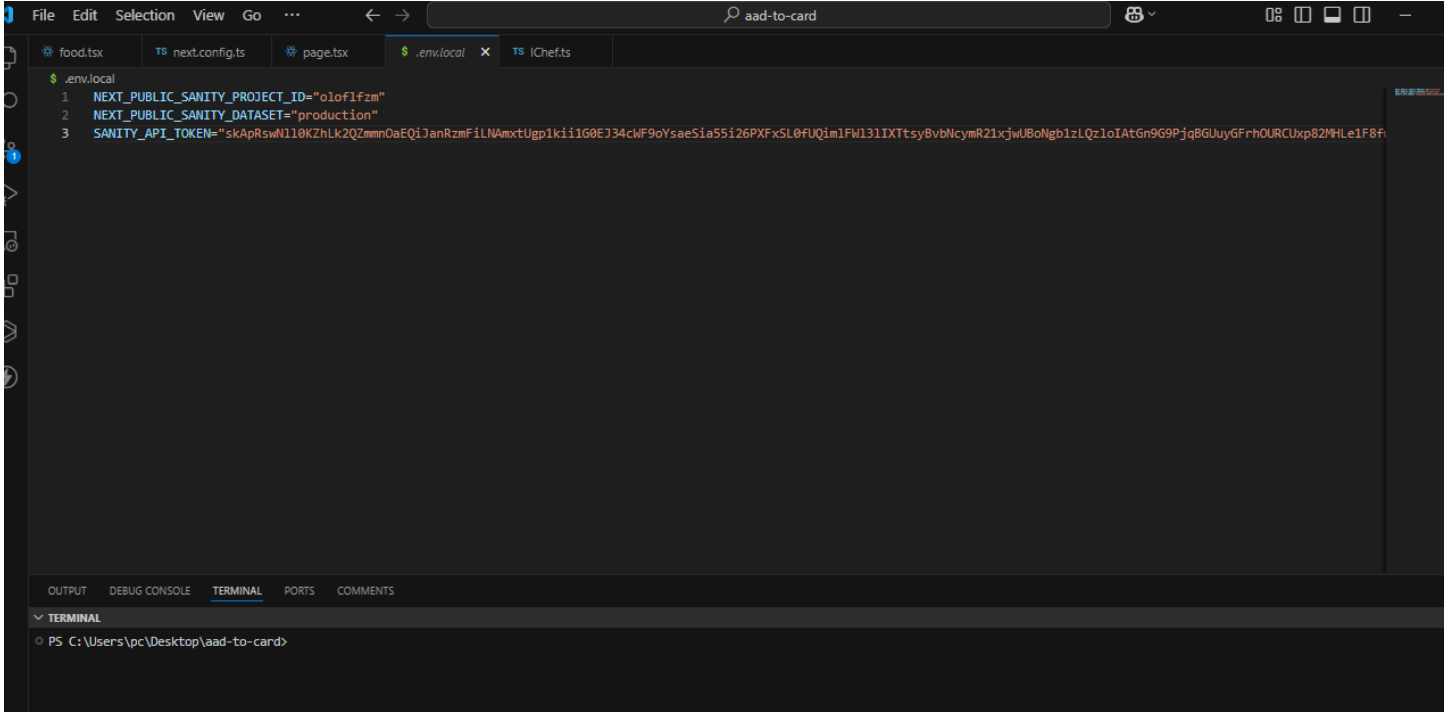
Title : Food tuck

api file

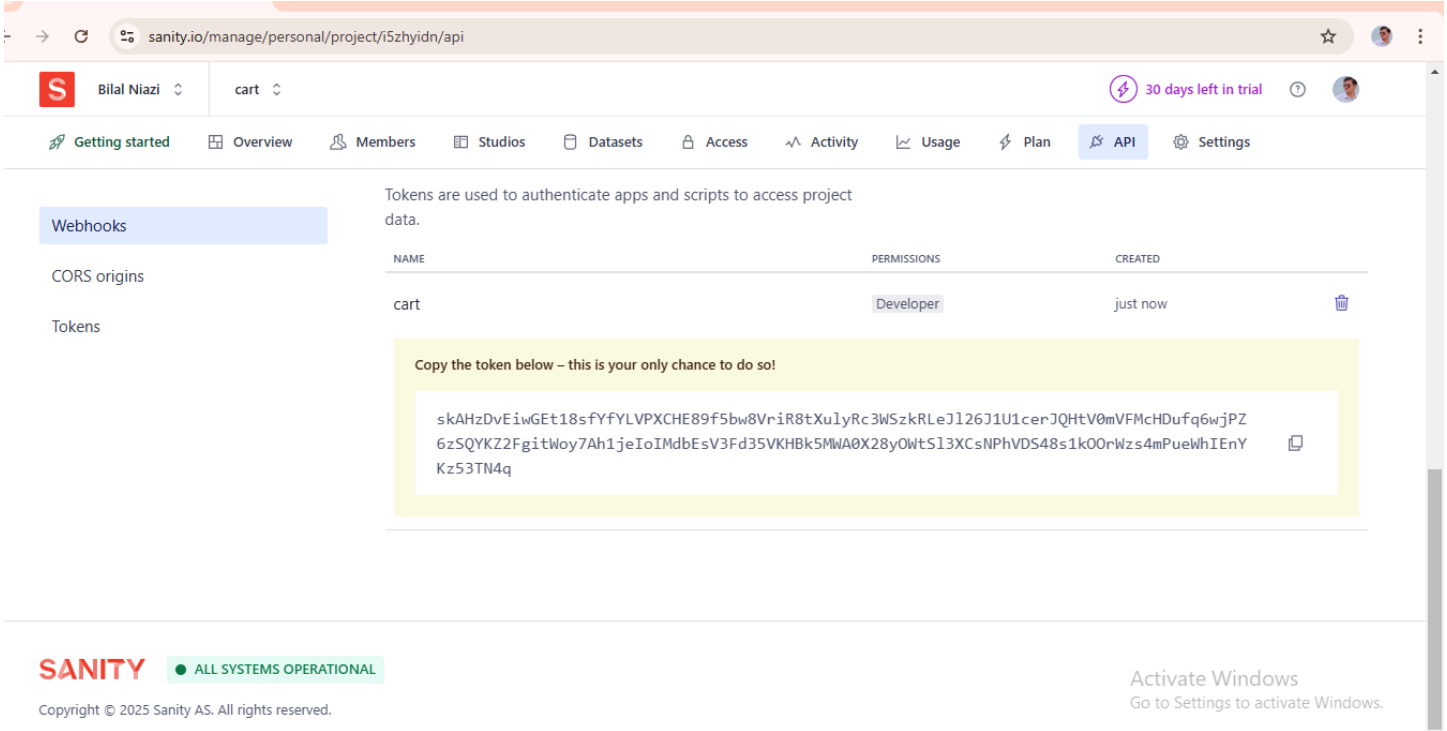
The screenshot shows the Sanity.io dashboard for a project named 'hacthon-day3'. The top navigation bar includes a search tab, the project name 'hacthon-day3', and a trial status '25 days left in trial'. The main content area is divided into two sections: 'Next steps' and 'Project members'. The 'Next steps' section contains a box with the text 'Initialize your project with the CLI' and a code snippet: `npm create sanity@latest -- --project olof1fzm --dataset produ`. Below the code is a 'Copy' button and a link 'Having issues with the CLI?'. The 'Project members' section shows a group of four user icons, with the text 'Invite your first team member' and a button '+ Invite project members'. The bottom of the dashboard has a 'Usage' section with a 'View more' link and an 'Activate Windows' watermark.

## Environment Variables

.ev.local

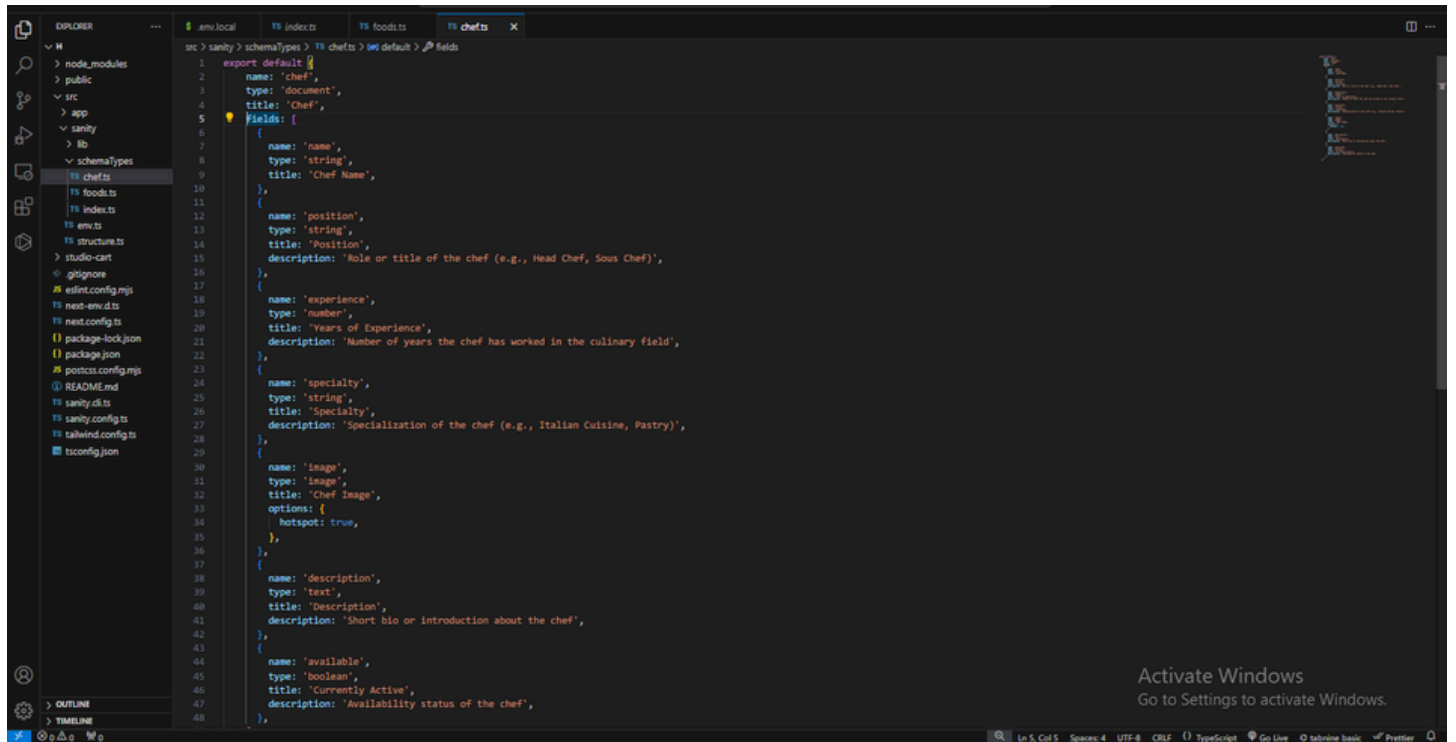


# Api Tokan



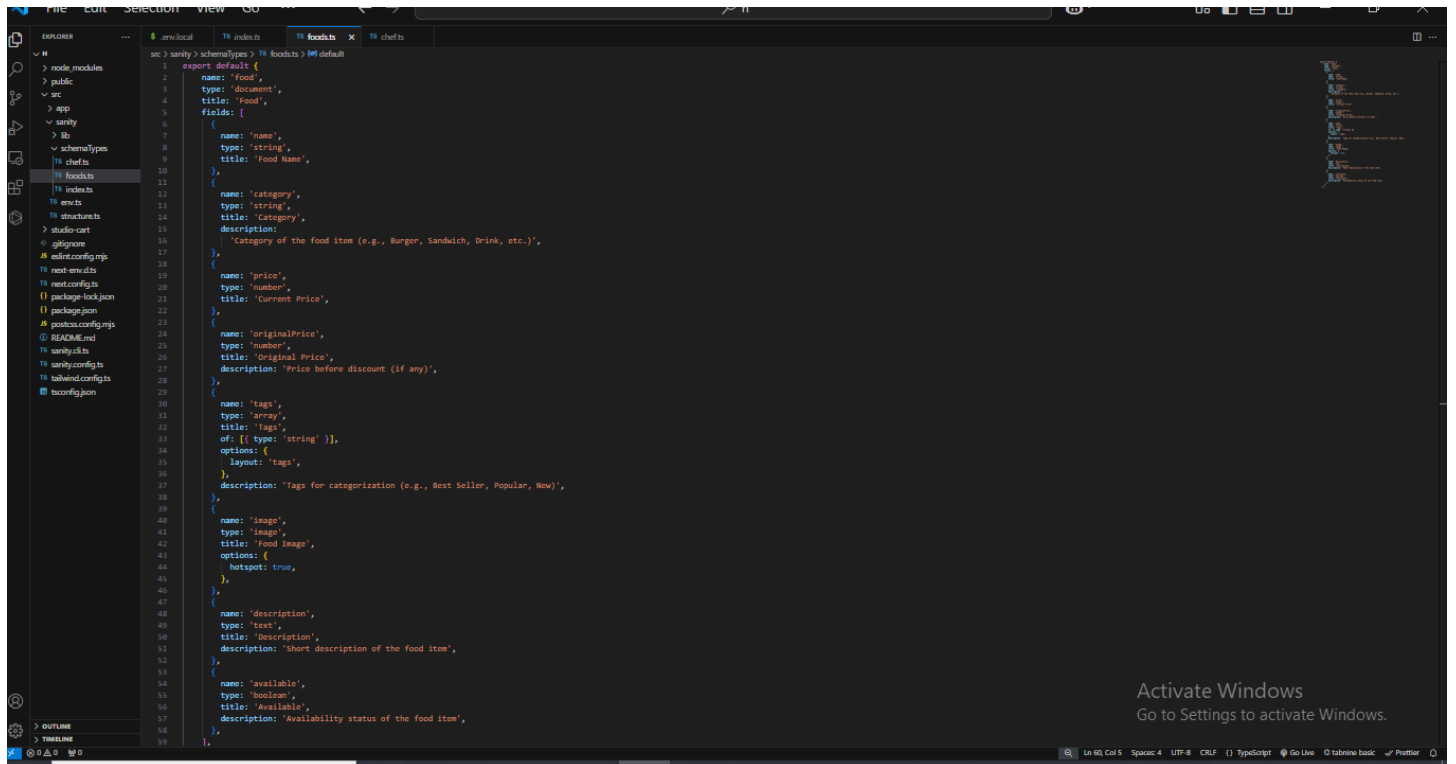
# schemaTypes

# chep schema type



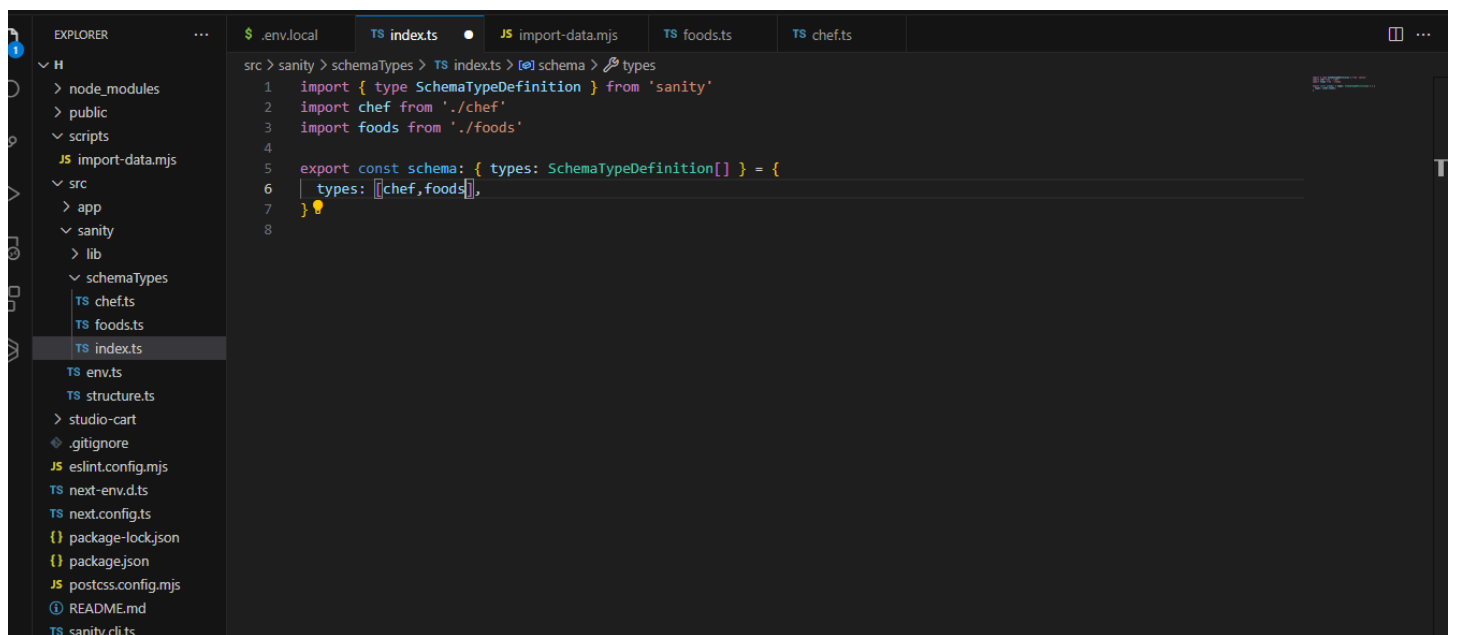
```
1 export default {
2   name: 'chef',
3   type: 'document',
4   title: 'Chef',
5   fields: [
6     {
7       name: 'name',
8       type: 'string',
9       title: 'Chef Name',
10    },
11    {
12      name: 'position',
13      type: 'string',
14      title: 'Position',
15      description: 'Role or 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 }
```

# food schema type



```
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: 'Category of the food item (e.g., Burger, Sandwich, Drink, etc.)',
16    },
17    {
18      name: 'price',
19      type: 'number',
20      title: 'Current Price',
21    },
22    {
23      name: 'originalPrice',
24      type: 'number',
25      title: 'Original Price',
26      description: 'Price before discount (if any)',
27    },
28    {
29      name: 'tags',
30      type: 'array',
31      title: 'Tags',
32      of: [{ type: 'string' }],
33      options: {
34        layout: 'tags',
35      },
36      description: 'Tags for categorization (e.g., Best Seller, Popular, New)',
37    },
38    {
39      name: 'image',
40      type: 'image',
41      title: 'Food Image',
42      options: {
43        hotspot: true,
44      },
45    },
46    {
47      name: 'description',
48      type: 'text',
49      title: 'Description',
50      description: 'Short description of the food item',
51    },
52    {
53      name: 'available',
54      type: 'boolean',
55      title: 'Available',
56      description: 'Availability status of the food item',
57    },
58  ],
59 }
```

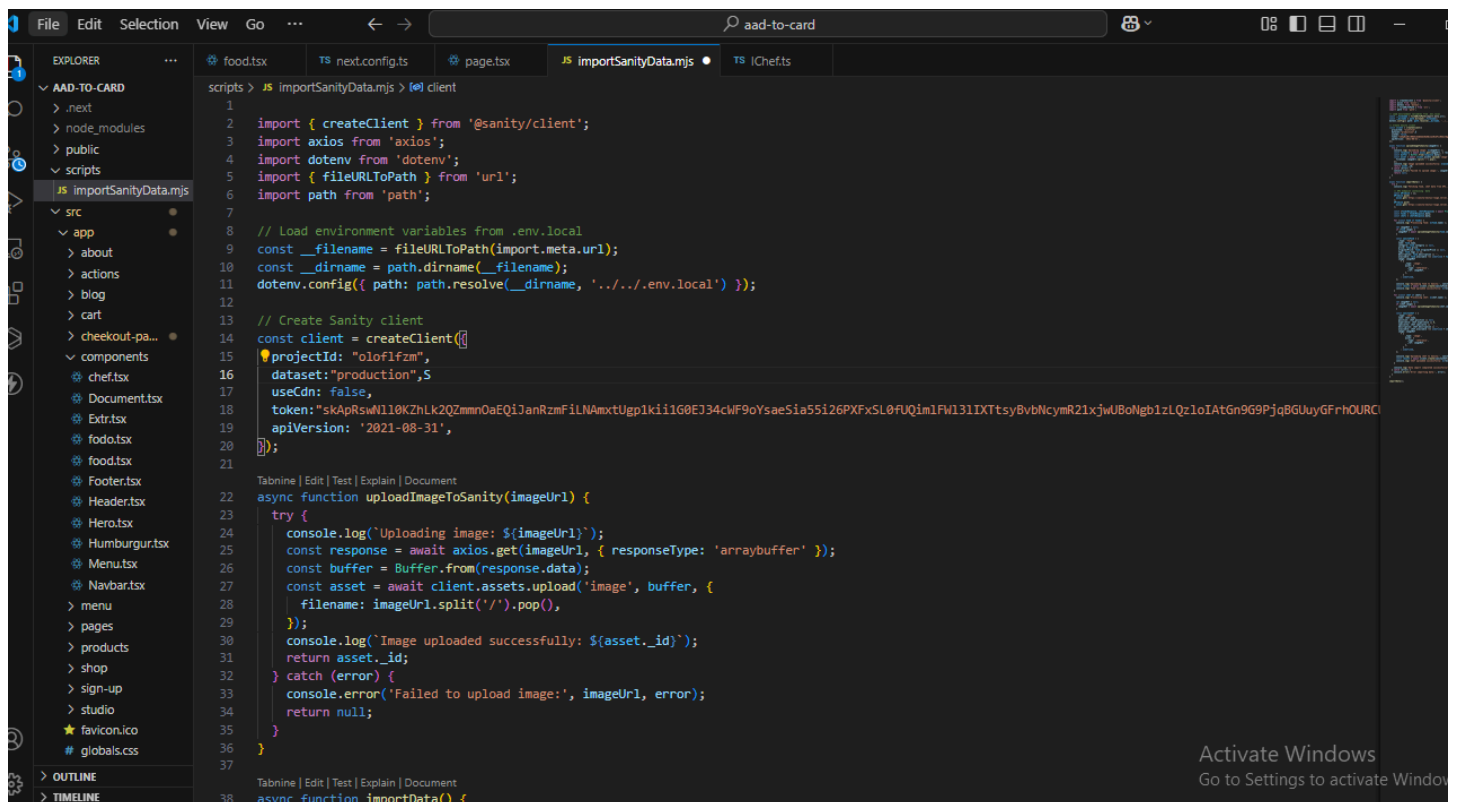
# Index.js



The screenshot shows the VS Code interface with the Explorer sidebar on the left and the editor window on the right. The Explorer sidebar shows a project structure with folders like 'node\_modules', 'public', 'scripts', 'src', 'sanity', 'lib', 'schemaTypes', and 'studio-cart'. The 'src' folder is expanded, showing files like 'app', 'sanity', 'lib', 'schemaTypes', 'TS chefs.ts', 'TS foods.ts', and 'TS index.ts'. The 'TS index.ts' file is selected. The editor window shows the content of 'index.ts' with the following code:

```
src> sanity > schemaTypes > TS index > schema > types
1 import { type SchemaTypeDefinition } from 'sanity'
2 import chef from './chef'
3 import foods from './foods'
4
5 export const schema: { types: SchemaTypeDefinition[] } = {
6   types: [chef, foods],
7 }
8
```

# importSanityData.mjs

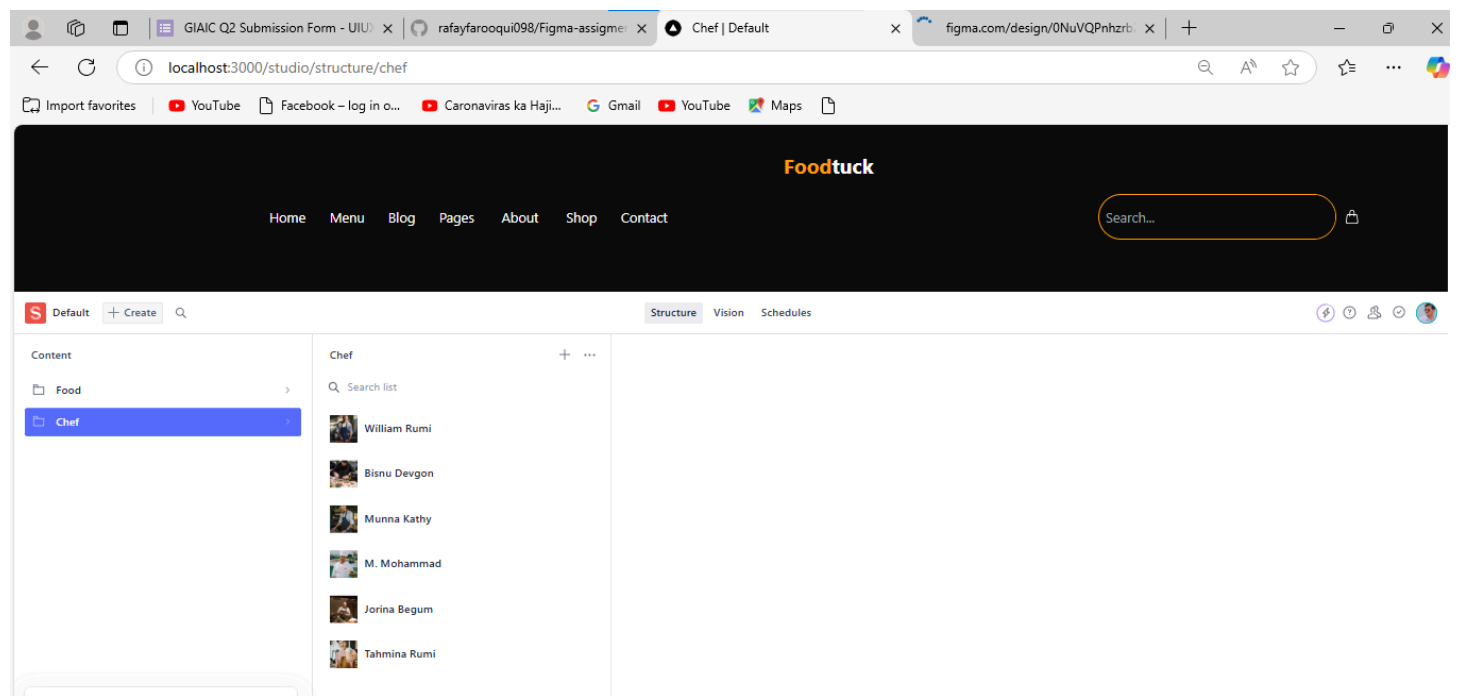


The screenshot shows the VS Code interface with the Explorer sidebar on the left and the editor window on the right. The Explorer sidebar shows a project structure with folders like 'node\_modules', 'public', 'scripts', 'src', 'sanity', 'lib', 'schemaTypes', and 'studio-cart'. The 'src' folder is expanded, showing files like 'app', 'sanity', 'lib', 'schemaTypes', 'TS chefs.ts', 'TS foods.ts', and 'TS index.ts'. The 'TS index.ts' file is selected. The editor window shows the content of 'importSanityData.mjs' with the following code:

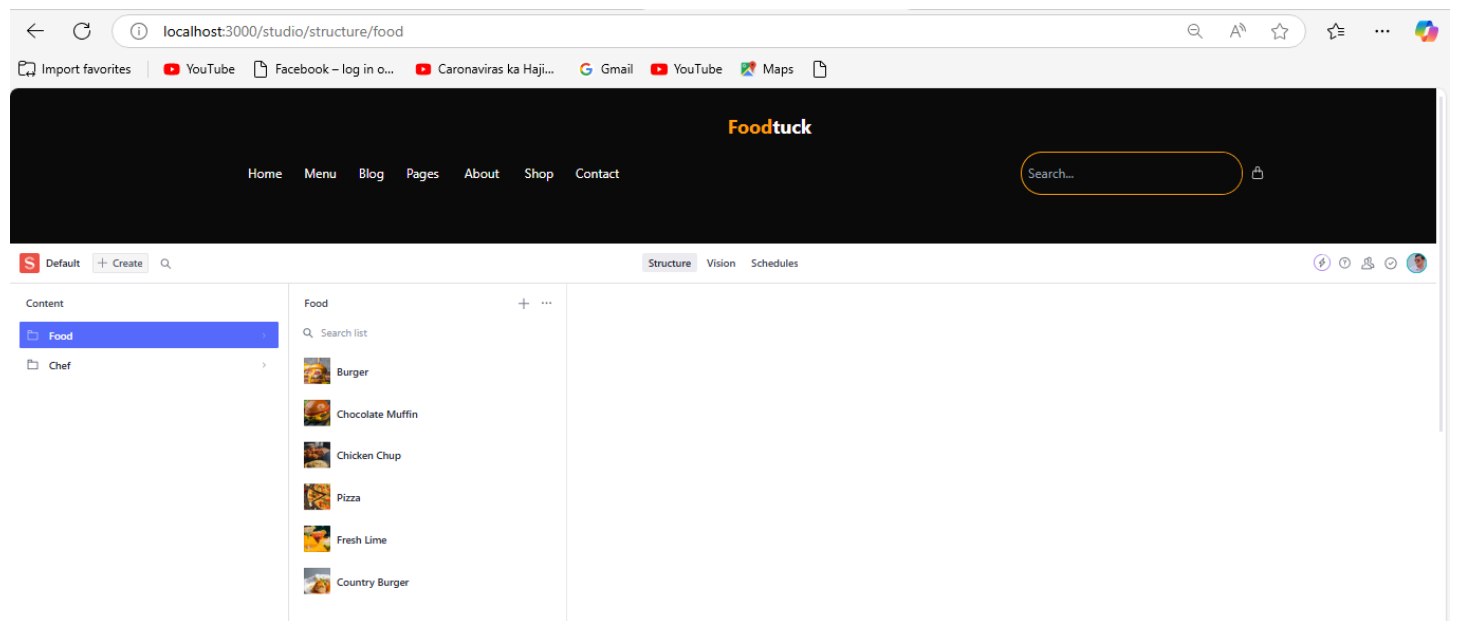
```
File Edit Selection View Go ... aad-to-card
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import dotenv from 'dotenv';
4 import { fileURLToPath } from 'url';
5 import path from 'path';
6
7 // Load environment variables from .env.local
8 const __filename = fileURLToPath(import.meta.url);
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: "olof1fzm",
15   dataset: "production",
16   useCdn: false,
17   token: "skApRsw1l0KZhLk2QZmmnOaEQiJanRzmFilNAmxTUp1kii1G0E734cWF9oYsaeSia55126PXFxSL0fUQim1FW131IXTsyBvNcymR21xjWUBoNgb1zLQz1oIAtGn9G9PjqBGuuyGFrhOURC",
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
```

# api data import

# chef data import



# food data import



# All data import sanity

localhost:3000/studio/vision

Import favorites YouTube Facebook – log in o... Caronaviras ka Haji... Gmail YouTube Maps

Default + Create

Structure Vision Schedules

DATASET: production API VERSION: Other CUSTOM API VERSION: v2025-01-24 PERSPECTIVE: raw QUERY URL (COPY TO CLIPBOARD): https://oioffm.api.sanity.io/v2025-01-24/data/query/production?query=%5B

QUERY

```
1 *[_type == "chef" || _type == "food"]
2
```

PARAMS

```
1 {
2
3 }
```

RESULT

```
[...] 12 items
0: {
  image: {
    _type: image
    asset: {
      _type: reference
      _ref: image-e155a50cdbc0e02ca81649945bd30a2fc111276-1248x1068-png
    }
    available: true
    price: 38
    category: Main Course
  }
  slug: {
    _type: slug
    current: burger
    originalPrice: 45
    _rev: NGbH4pjcDaYIxrHyZUDIX6
    description: Juicy beef burger with fresh lettuce, tomatoes, and cheese.
    _id: NGbH4pjcDaYIxrHyZODH0a
  }
  _type: food
  tags: [
    0: Healthy
    1: Popular
  ]
  _createdAt: 2025-01-25T03:37:25Z
  _updatedAt: 2025-01-26T13:22:21Z
  name: Burger
}
```

Fetch Listen

Execution: 8ms End-to-end: 488ms

Activate Windows  
Go to Settings to activate Windows.

Save result as JSON CSV