

DAY-3 API INTEGRATION & DATA MIGRATION REPORT - [FOOD HUB]

STEPS TO PERFORM THE ABOVE STATED JOB SUCCESSFULLY.

1st - Installed the Sanity Studio in Next.js project with the following command.

npm create sanity@latest

1. I have created an account. Select a login provider from the list of options, and confirm with Enter. After creating an account in the browser, come back to the command line window.
2. It has asked me the following questions.
 - Would you like to add configuration files for a Sanity project in this Next.js folder? **Yes**
 - Would you like an embedded Sanity Studio? **Yes**
 - Would you like to use the Next.js app directory for routes? **Yes**
 - What route do you want to use for the Studio? **/studio**
 - **Select project template to use Clean project with no predefined schemas**
 - Would you like to add the project ID and dataset to your .env file? **Yes**
3. Wait a bit for the installation process to complete. When I get a Success! message, I was good to move on to the next step!! 🎉

```
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

D:\QUARTER 2 GIAIC\3 Hackathon\Food Hub>npm create sanity@latest
Need to install the following packages:
create-sanity@3.71.2
Ok to proceed? (y) y

> second-assignment@0.1.0 npx
> create-sanity

[?] You are logged in as muhammadrashid628@gmail.com using Google
[?] Fetching existing projects

[?] Create a new project or select an existing one Create new project
[?] Your project name: food-hub-day-3
Your content will be stored in a dataset that can be public or private, depending on
whether you want to query your content with or without authentication.
The default dataset configuration has a public dataset named "production".
[?] Use the default dataset configuration? Yes
[?] Creating dataset
[?] Would you like to add configuration files for a Sanity project in this Next.js folder? Yes

[?] It looks like you are using Next.js 15 and React 19
[?] Please read our compatibility guide.
[?] https://www.sanity.io/help/react-19

[?] Do you want to use TypeScript? Yes
[?] Would you like an embedded Sanity Studio? Yes
[?] What route do you want to use for the Studio? /studio
[?] Select project template to use Clean project with no predefined schema types
[?] Would you like to add the project ID and dataset to your .env.local file? Yes
Added http://localhost:3000 to CORS origins
```

```
added 8 packages, and audited 1294 packages in 13s

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

3 vulnerabilities (2 moderate, 1 high)

To address issues that do not require attention, run:
  npm audit fix

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

Run `npm audit` for details.

Success! Your Sanity configuration files has been added to this project

D:\QUARTER 2 GIAIC\3 Hackathon\Food Hub>
```

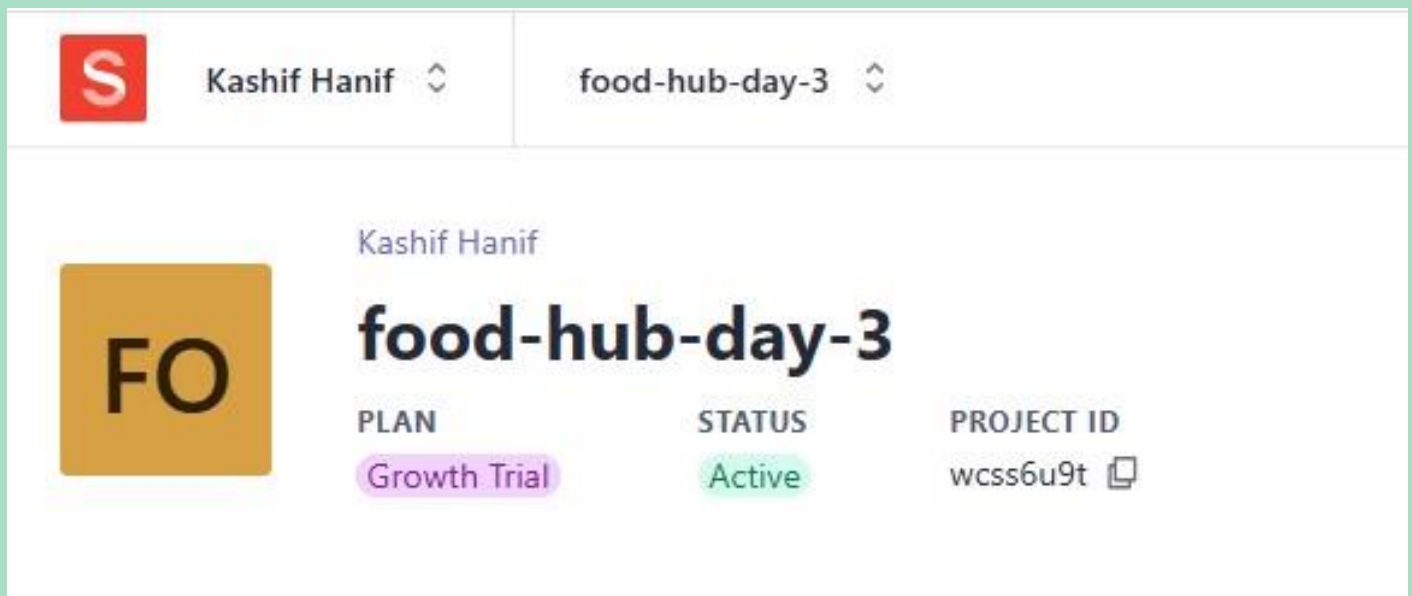
2nd -Generating the Token Number.

After the successful installation of sanity studio, visit the browser and open the

<https://www.sanity.io/manage?ref=homepage>

In sanity studio I have founded there my project and I have clicked on it to view my project detail.

Like... Project ID, Generating of Token in API Link.



The screenshot displays the Sanity Studio interface. At the top, there's a header with the Sanity logo (a red 'S') on the left, the user name 'Kashif Hanif' with a dropdown arrow in the middle, and the project name 'food-hub-day-3' with a dropdown arrow on the right. Below the header, the main content area shows the project details for 'food-hub-day-3'. On the left is a square orange icon with the letters 'FO' in white. To the right of the icon, the project name 'food-hub-day-3' is displayed in a large, bold, black font. Below the project name, there are three columns of information: 'PLAN' with a value of 'Growth Trial' in a purple pill-shaped box, 'STATUS' with a value of 'Active' in a green pill-shaped box, and 'PROJECT ID' with the value 'wc5s6u9t' and a small copy icon to its right.

After this step I have click on given API link to generate the **Token Number** for my project use. I have given the API name **Third Day** to generate the Token number and **marked** on the **Developer** to read and write access to all datasets and in last press the save button.

Third Day

Permissions

Choose the access privileges for the token.

- Contributor
- ☐ Read and write access to draft content within all datasets, with no access to project settings. (Tokens: read+write drafts)
- ☐ Deploy Studio (Token only)
- ☐ Access to deploy Sanity Studio and GraphQL APIs to our hosted service.
- Developer
- ☒ Read and write access to all datasets, with access to project settings for developers. (Tokens: read+write)
- Editor
- ☐ Read and write access to all datasets, with limited access to project settings. (Tokens: read+write)
- Viewer
- ☐ Read access to all datasets, with limited access to project settings. (Tokens: read-only)

Save

Cancel

Activate Win
Go to Settings to

Token number Generated Successfully.

NAME	PERMISSIONS	CREATED	
Third Day	Developer	just now	
Copy the token below – this is your only chance to do so!			
<div>skUkVAYJ58C4RP12u21uNYWJKfcLMulq0AlYXZei9DpGoz2I3zin4zTPSSSN2aR1NGw77C67lKvmn1dHTDtcFTla9JATxC9Ng8D1BlVET7sp48cliYH82uuASxtWBgqR6WXX2ynlScieVM5lSaHTgdwV8Lz1u4nhUMReUAB1w56yBYK2yFjw</div>			

3rd- .env.local File.

I have added my project ID and Token number in **.env.local** file.

```
... $ .env.local x
$ .env.local
1 NEXT_PUBLIC_SANITY_PROJECT_ID="wc5s6u9t"
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 SANITY_API_TOKEN="skUkVAYJ58C4RP12u21uNYWJKfcLMulq0AlYXZei9DpGoz2I3zin4zTPSS5N2aR1NGw77C67lKvmn1dHTDtcFT1a9JATxC9Ng8D1B1v"
4
```

4rd- Define Schema.

Next step was of my project to integrate the API of **Sir Mubashir template-09** in sanity studio in his provided GitHub repository.

I have made the schema type in my project of **Food.ts** as the given in GitHub repository.

```
EXPLORER    ...    $ .env.local    TS food.ts U x
FOOD HUB
  src
  app
  FAQ
  fonts
  homePage
  studio
  favicon.ico
  # globals.css
  layout.tsx M
  page.tsx M
  components
  sanity
  lib
  schemaTypes
    TS food.ts U
    TS index.ts U
    TS env.ts U
    TS structure.ts U
  .env.local
  .eslintc.json
  .gitignore
  TS next-env.d.ts
  TS next.config.ts
  {} package-lock.json M
  {} package.json M
  OUTLINE
  TIMELINE
  APPLICATION BUILDER

src > sanity > schemaTypes > TS food.ts > [e] food
1 import { defineType } from "sanity";
2
3 export const food = defineType ({
4   name: 'food',
5   type: 'document',
6   title: 'Food',
7   fields: [
8     {
9       name: 'name',
10      type: 'string',
11      title: 'Food Name',
12    },
13    {
14      name: 'category',
15      type: 'string',
16      title: 'Category',
17      description:
18        'Category of the food item (e.g., Burger, Sandwich, Drink, etc.)',
19    },
20    {
21      name: 'price',
22      type: 'number',
23      title: 'Current Price',
24    },
25    {
26      name: 'originalPrice',
27      type: 'number',
28      title: 'Original Price',
29      description: 'Price before discount (if any)',
30    },
31    {
32      name: 'tags',
```

```
local TS food.ts U X
anity > schemaTypes > TS food.ts > [e] food
{
  name: 'tags',
  type: 'array',
  title: 'Tags',
  of: [{ type: 'string' }],
  options: {
    layout: 'tags',
  },
  description: 'Tags for categorization (e.g., Best Seller, Popular, New)',
},
{
  name: 'image',
  type: 'image',
  title: 'Food Image',
  options: {
    hotspot: true,
  },
},
{
  name: 'description',
  type: 'text',
  title: 'Description',
  description: 'Short description of the food item',
},
{
  name: 'available',
  type: 'boolean',
  title: 'Available',
  description: 'Availability status of the food item',
},
],
}
```

5th- Adding in Package.json File.

In Package.json file I have added 2 things.

- 1- **"type": "module",** (in line number 5)
- 2- **"import-data": "node script/importdata.mjs"** (in line number 11)


```
{
  "name": "second-assignment",
  "version": "0.1.0",
  "private": true,
  "type": "module",
  "scripts": {
    "dev": "next dev --turbo",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "import-data": "node script/importData.mjs"
  },
  "dependencies": {
    "@sanity/image-url": "^1.1.0",
    "@sanity/vision": "^3.71.2",
    "next": "15.0.1",
    "next-sanity": "^9.8.42",
    "react": "19.0.0-rc-69d4b800-20241021",
    "react-dom": "19.0.0-rc-69d4b800-20241021",
    "sanity": "^3.71.2",
    "styled-components": "^6.1.14"
  },
  "devDependencies": {
    "@types/node": "^20",
    "@types/react": "^18"
  }
}
```

6th- Script folder & importData.mjs File.

I have made **Script Folder** before the src/app route & add the file **importData.mjs** in this to import the provided API of sir Mubashir of template-09 in sanity studio.

```
script > JS importData.mjs > client > apiVersion
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: "wc5s6u9t",
15   dataset: "production",
16   useCdn: true,
17   token: "skUkVAYJ58C4RPl2u21uNYWJKfcLMulq0ALYXZeI9DpGoz2I3zin4zTPSSSN2aR1NGw77C67lKvmn1dHTDtcFTla9JATxC9Ng8D1B1VE",
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   } catch (error) {
30     console.error('Error uploading image:', error);
31   }
32 }
```

EXPLORER...JS importData.mjs U X

FOOD HUB

script > JS importData.mjs > client > apiVersion

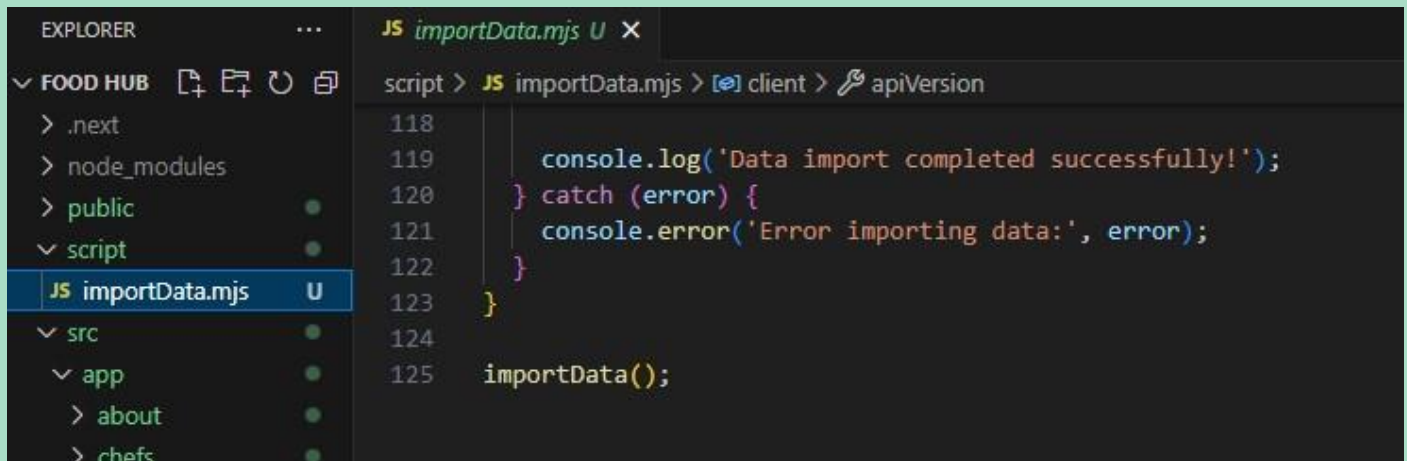
29 console.log(`Image uploaded successfully: \${asset._id}`);
30 return asset._id;
31 } catch (error) {
32 console.error('Failed to upload image:', imageUrl, error);
33 return null;
34 }
35 }
36
37 async function importData() {
38 try {
39 console.log('Fetching food, chef data from API...');
40
41 // 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 const [foodsResponse, chefsResponse] = await Promise.all(\$Promise);
51 const foods = foodsResponse.data;
52 const chefs = chefsResponse.data;
53
54 for (const food of foods) {
55 console.log(`Processing food: \${food.name}`);
56

EXPLORER...JS importData.mjs U X

FOOD HUB

script > JS importData.mjs > client > apiVersion

57 let imageRef = null;
58 if (food.image) {
59 imageRef = await uploadImageToSanity(food.image);
60 }
61
62 const sanityFood = {
63 _type: 'food',
64 name: food.name,
65 category: food.category || null,
66 price: food.price,
67 originalPrice: food.originalPrice || null,
68 tags: food.tags || [],
69 description: food.description || '',
70 available: food.available !== undefined ? food.available : true,
71 image: imageRef
72 ? {
73 _type: 'image',
74 asset: {
75 _type: 'reference',
76 _ref: imageRef,
77 },
78 }
79 : undefined,
80 };
81
82 console.log('Uploading food to Sanity:', sanityFood.name);
83 const result = await client.create(sanityFood);
84 console.log(`Food uploaded successfully: \${result._id}`);



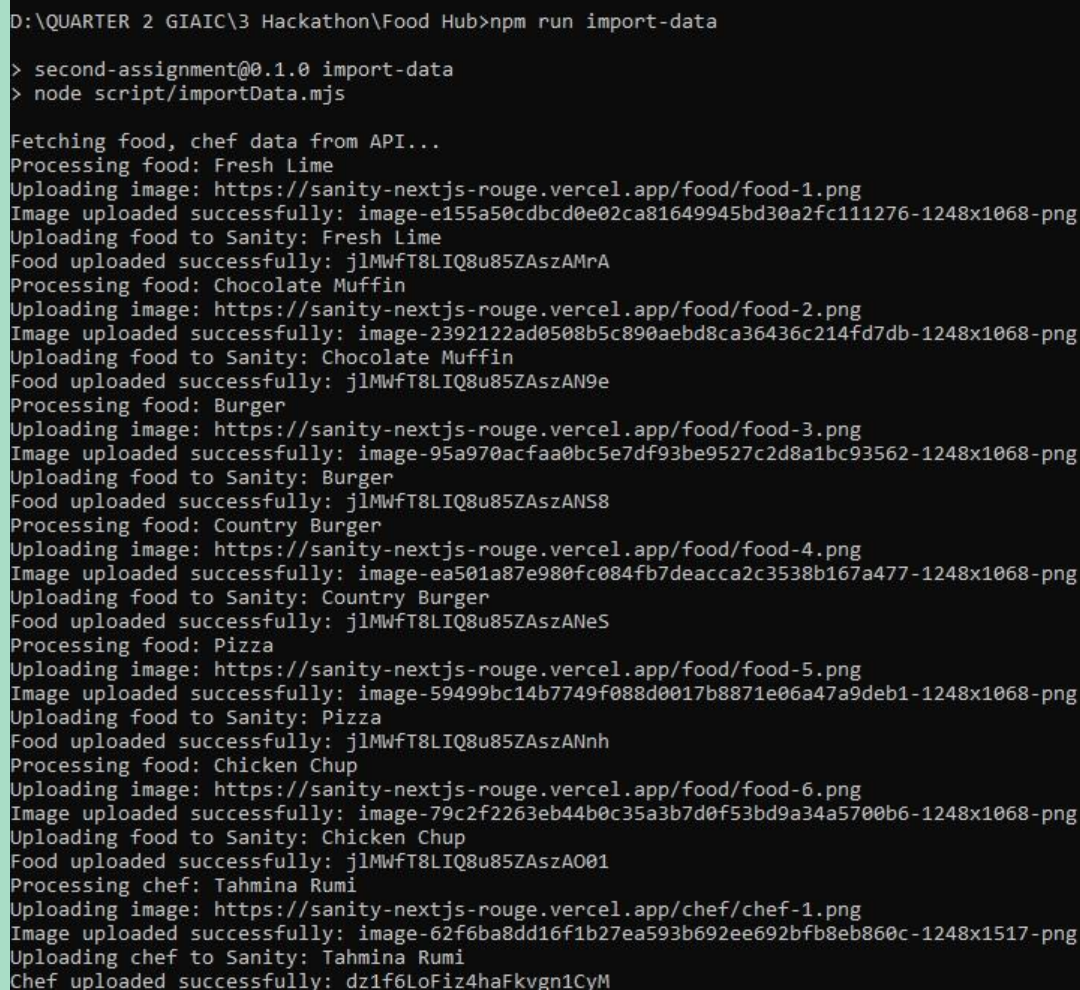
The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the project structure: FOOD HUB, .next, node_modules, public, script, JS importData.mjs (selected), src, app, about, and chefs. The main editor window shows the content of importData.mjs, which includes a script tag for client/apiVersion and a JavaScript function importData() that fetches data from an API, processes it, and uploads it to Sanity. The function uses console.log and console.error for logging and error handling.

```
script > JS importData.mjs > client > apiVersion
118
119     console.log('Data import completed successfully!');
120   } catch (error) {
121     console.error('Error importing data:', error);
122   }
123 }
124
125 importData();
```

7th- Run the command to Import the data.

npm run import-data

After running this command my API data has been imported successfully in sanity.



The terminal output shows the command 'npm run import-data' being executed in a directory named 'D:\QUARTER 2 GIAIC\3 Hackathon\Food Hub'. The output displays the process of fetching food and chef data from an API, processing it, and uploading it to Sanity. The data is organized into food items (Fresh Lime, Chocolate Muffin, Burger, Country Burger, Pizza, Chicken Chup) and a chef (Tahmina Rumi). Each item is processed and uploaded successfully, with the corresponding image URL and the Sanity ID (j1MWfT8LIQ8u85ZAszANrA, etc.) displayed.

```
D:\QUARTER 2 GIAIC\3 Hackathon\Food Hub>npm run import-data
> second-assignment@0.1.0 import-data
> node script/importData.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: j1MWfT8LIQ8u85ZAszANrA
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: j1MWfT8LIQ8u85ZAszAN9e
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: j1MWfT8LIQ8u85ZAszANS8
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: j1MWfT8LIQ8u85ZAszANeS
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: j1MWfT8LIQ8u85ZAszANnh
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: j1MWfT8LIQ8u85ZAszA001
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: dz1f6LoFiz4haFkvg1CyM
```



```

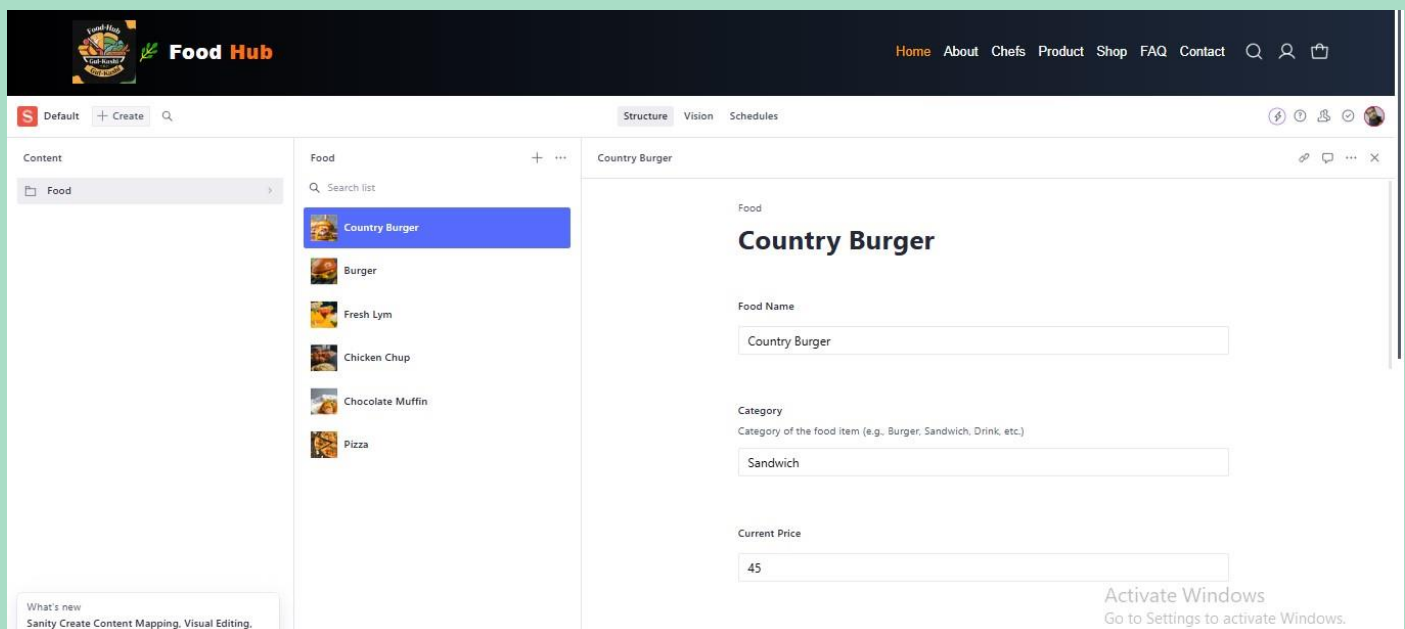
Processing chef: Jorina Begum
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-2.png
Image uploaded successfully: image-a8a4535b34a230733d2ef6eb5c0a4169f65226d5-1248x1517-png
Uploading chef to Sanity: Jorina Begum
Chef uploaded successfully: j1MwFT8LIQ8u85ZAszAOLa
Processing chef: M. Mohammad
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-3.png
Image uploaded successfully: image-9b9161acc32440ad4a6b853851b9c232b9c9c53e-1248x1517-png
Uploading chef to Sanity: M. Mohammad
Chef uploaded successfully: dz1f6LoFiz4haFkvgn1DOY
Processing chef: Munna Kathy
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-4.png
Image uploaded successfully: image-03eb4eacebd8ca11b707cfc569b87894b4e9bd57-1248x1517-png
Uploading chef to Sanity: Munna Kathy
Chef uploaded successfully: 0lAzcbUv2dscuQkndQo3IL
Processing chef: Bisnu Devgon
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-5.png
Image uploaded successfully: image-7576fb850ddb0f7d4cefab457f848c09a816186d-1248x1517-png
Uploading chef to Sanity: Bisnu Devgon
Chef uploaded successfully: dz1f6LoFiz4haFkvgn1DsU
Processing chef: William Rumi
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-6.png
Image uploaded successfully: image-ef1c3b9ecfd9bc1aad0a931c6b4c564d6939e4f8-1248x1517-png
Uploading chef to Sanity: William Rumi
Chef uploaded successfully: j1MwFT8LIQ8u85ZAszAOwY
Data import completed successfully!

D:\QUARTER 2 GIAIC\3 Hackathon\Food Hub>_

```

8th- Open the Sanity Studio & see the Import Data.

After the successfully completion of above command I have founded my API Food Data on sanity studio.



9th- Checking of GROQ QUERY in Sanity Vision.

To display & fetching the import data on my frontend, I have checked first in Sanity Vision that it is fetching & working properly or not.

The screenshot shows the Sanity Vision interface with a GROQ query and its results. The query is:

```
QUERY
*[_type == "food"] {
  name,
  category,
  price,
  description,
  originalPrice,
  "imageUrl": image.asset->url,
  tags
}
```

The results are:

```
RESULT
[6] 6 items
0: { 7 properties
  name: Country Burger
  category: Sandwich
  price: 45
  description: Classic country-style burger served with fries.
  originalPrice: 50
  imageUrl: https://cdn.sanity.io/images/wcss6u9t/production/e155a50cdbc0e02ca81649945bd30a2fc111276-1248x1068.png
  tags: [1] 1 item
    0: Recommended
1: { 7 properties
  category: Sandwich
  price: 21
  description: Juicy beef burger with fresh lettuce, tomatoes, and cheese.
  originalPrice: 45
  imageUrl: https://cdn.sanity.io/images/wcss6u9t/production/2392122ad0508b5c890aebd8ca36436c214fd7db-1248x1068.png
  tags: [1] 1 item
    0: Popular
  name: Burger
2: { 7 properties
  description: Refreshing fresh lime drink made with natural ingredients.
  originalPrice: 45
  imageUrl: https://cdn.sanity.io/images/wcss6u9t/production/95a970acfaa0bc5e7df93be9527c2d8a1bc93562-1248x1068.png
  tags: [2] 2 items
```

At the bottom, there are buttons for "Fetch" and "Listen", and a status bar showing "Execution: 11ms End-to-end: 279ms".

The screenshot shows the Sanity Vision interface with a GROQ query and its results. The query is:

```
QUERY
1 *[_type == "food"] {
2   name,
3   category,
4   price,
5   description,
6   originalPrice,
7   "imageUrl": image.asset->url,
8   tags
9 }
```

The results are:

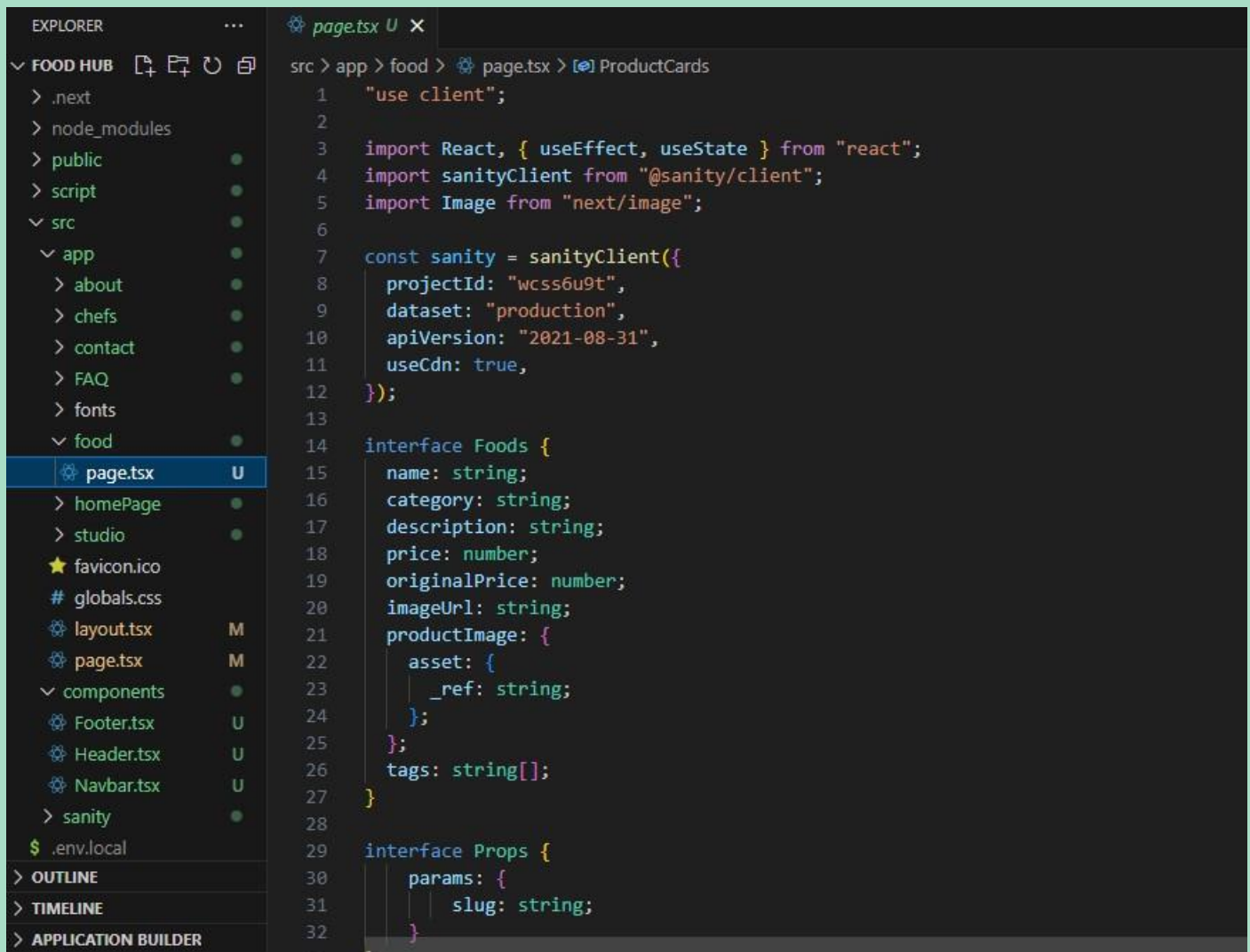
```
RESULT
[6] 6 items
0: { 7 properties
1: { 7 properties
2: { 7 properties
3: { 7 properties
4: { 7 properties
5: { 7 properties
```

At the bottom, there are buttons for "Fetch" and "Listen", and a status bar showing "Execution: 11ms End-to-end: 279ms".

I have found that my GROQ Query is working properly and fetching the all data which is given in provided API.

10th- Migrate the data on Frontend.

To migrate the data from the Sanity studio to Frontend of my project. I have made a Food folder in src/app route and fetch the data with the help of **Map method**, **useState** & **useEffect** hooks.



```
EXPLORER
FOOD HUB
  .next
  node_modules
  public
  script
  src
    app
      about
      chefs
      contact
      FAQ
      fonts
      food
        page.tsx
      homePage
      studio
      favicon.ico
      globals.css
      layout.tsx
      page.tsx
    components
      Footer.tsx
      Header.tsx
      Navbar.tsx
    sanity
  .env.local
  OUTLINE
  TIMELINE
  APPLICATION BUILDER

src > app > food > page.tsx > ProductCards
1  "use client";
2
3  import React, { useEffect, useState } from "react";
4  import sanityClient from "@sanity/client";
5  import Image from "next/image";
6
7  const sanity = sanityClient({
8    projectId: "wc5s6u9t",
9    dataset: "production",
10    apiVersion: "2021-08-31",
11    useCdn: true,
12  });
13
14  interface Foods {
15    name: string;
16    category: string;
17    description: string;
18    price: number;
19    originalPrice: number;
20    imageUrl: string;
21    productImage: {
22      asset: {
23        _ref: string;
24      };
25    };
26    tags: string[];
27  }
28
29  interface Props {
30    params: {
31      slug: string;
32    }
33  }
```


EXPLORER

FOOD HUB

> .next

> node_modules

> public

> script

> src

> app

> about

> chefs

> contact

> FAQ

> fonts

> food

page.tsx

> homePage

> studio

★ favicon.ico

globals.css

layout.tsx

page.tsx

> components

Footer.tsx

Header.tsx

Navbar.tsx

> sanity

\$.env.local

OUTLINE

TIMELINE

APPLICATION BUILDER

page.tsx U X

src > app > food > page.tsx > ProductCards

```
28
29 interface Props {
30   params: {
31     slug: string;
32   }
33 }
34
35 const ProductCards: React.FC = () => {
36   const [foods, setFoods] = useState<Foods[]>([]);
37   const [cart, setCart] = useState<Foods[]>([]);
38
39   const fetchProducts = async () => {
40     try {
41       const query = `
42         *[_type == "food"] {
43           name,
44           category,
45           price,
46           description,
47           originalPrice,
48           "imageUrl": image.asset->url,
49           tags
50         }
51       `;
52
53       const data = await sanity.fetch(query);
54       setFoods(data);
55       console.log(data)
56     } catch (err) {
57       console.error("Error Fetching Products:", err);
58     }
59   };
60 }
```

```
EXPLORER  ...  page.tsx U X
FOOD HUB
  > .next
  > node_modules
  > public
  > script
  > src
    > app
      > about
      > chefs
      > contact
      > FAQ
      > fonts
      > food
        > page.tsx U
        > homePage
        > studio
        > favicon.ico
        > globals.css
        > layout.tsx M
        > page.tsx M
        > components
        > Footer.tsx U
        > Header.tsx U
        > Navbar.tsx U
        > sanity
        > .env.local
    > OUTLINE
    > TIMELINE
    > APPLICATION BUILDER

src > app > food > page.tsx > ProductCards
62  const addToCart = (food: Foods) => {
63    setCart((prevCart) => [...prevCart, food]);
64    alert(`${food.name} has been added to your cart!`);
65  };
66
67  useEffect(() => {
68    fetchProducts();
69  }, []);
70
71  return (
72    <div className="p-4 bg-black min-h-screen">
73      <h1 className="text-center text-yellow-400 mt-4 mb-4 text-4xl font-extrabold tracking-wide uppercase">
74        <q> Our Menu </q></h1>
75      <br />
76      <h2 className="text-center text-red-500 mt-4 mb-8 text-2xl font-bold">
77        Best Food for your Life
78      </h2>
79
80      <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
81        {foods.map((food) => (
82          <div
83            key={food.name}
84            className="bg-gray-900 text-white shadow-xl rounded-lg overflow-hidden transition-transform duration-300"
85          >
86            <div className="overflow-hidden">
87              <Image
88                src={food.imageUrl}
89                alt={food.name}
90                width={500}
91                height={500}
92                className="w-full h-48 object-cover rounded-t-md transition-transform duration-300 hover:scale-110"
93              />
94            </div>
95          </div>
96        ))}
97      </div>
98    </div>
99  );
100
```

```
EXPLORER  ...  page.tsx U X
FOOD HUB
  > .next
  > node_modules
  > public
  > script
  > src
    > app
      > about
      > chefs
      > contact
      > FAQ
      > fonts
      > food
        > page.tsx U
        > homePage
        > studio
        > favicon.ico
        > globals.css
        > layout.tsx M
        > page.tsx M
        > components
        > Footer.tsx U
        > Header.tsx U
        > Navbar.tsx U
        > sanity
        > .env.local
    > OUTLINE
    > TIMELINE
    > APPLICATION BUILDER

src > app > food > page.tsx > ProductCards
96  <div className="p-4">
97    <h2 className="text-xl font-bold text-red-400">{food.name}</h2>
98    <p className="text-gray-300 mt-2 text-sm">{food.description}</p>
99    <div className="flex justify-between items-center mt-4">
100      <div className="">
101        <p className="text-yellow-400 font-bold">
102          Price: ${food.price}
103        </p>
104        <p className="text-green-500 text-sm">
105          Original Price: ${food.originalPrice}
106        </p>
107      </div>
108    </div>
109
110    <div className="mt-2 flex flex-wrap gap-2">
111      {food.tags.map((tag, index) => (
112        <span
113          key={index}
114          className="text-xs bg-gray-700 text-white rounded-full px-3 py-1"
115        >
116          {tag}
117        </span>
118      ))}
119    </div>
120
121    {/* Add to cart Functionality */}
122    <button
123      className="mt-4 w-full bg-red-600 text-white py-2 rounded-md hover:bg-red-700 transition-all font-l"
124      onClick={() => addToCart(food)}
125    >
126      Add to Cart
127    </button>
128  </div>
129
```

EXPLORER

FOOD HUB

.next

node_modules

public

script

src

app

about

chefs

contact

FAQ

fonts

food

page.tsx

homePage

studio

favicon.ico

globals.css

layout.tsx

page.tsx

components

Footer.tsx

Header.tsx

Navbar.tsx

sanity

.env.local

OUTLINE

TIMELINE

APPLICATION BUILDER

page.tsx U

src > app > food > page.tsx > ProductCards

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

</div>

</div>

))}

</div>

/* Cart Summary */

<div className="mt-8 bg-gray-800 p-6 rounded-lg shadow-md">

<h2 className="text-lg font-bold text-red-500">Cart Summary</h2>

{cart.length > 0 ? (

<ul className="space-y-4">

{cart.map((item, index) => (

<li

key={index}

className="flex justify-between items-center bg-gray-700 shadow-sm p-4 rounded-md"

>

<div>

<p className="font-medium text-white">{item.name}</p>

<p className="text-sm text-yellow-400">

{item.price}

</p>

<Image

src={item?.imageUrl}

alt={item.name}

width={50}

height={50}

className="rounded-md"

</div>

))}

</div>

): (

159

160

161

162

163

164

165

166

167

168

169

170

) : (

<p className="text-gray-400 text-center">

Your Cart is empty. Please add Products

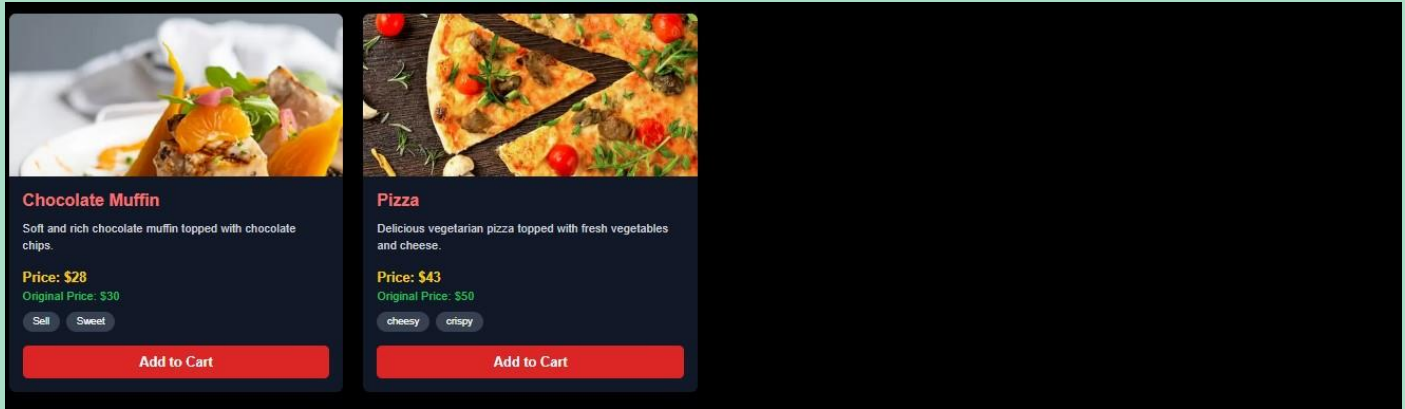
</p>

</div>

11th- Run the command "npm run dev".

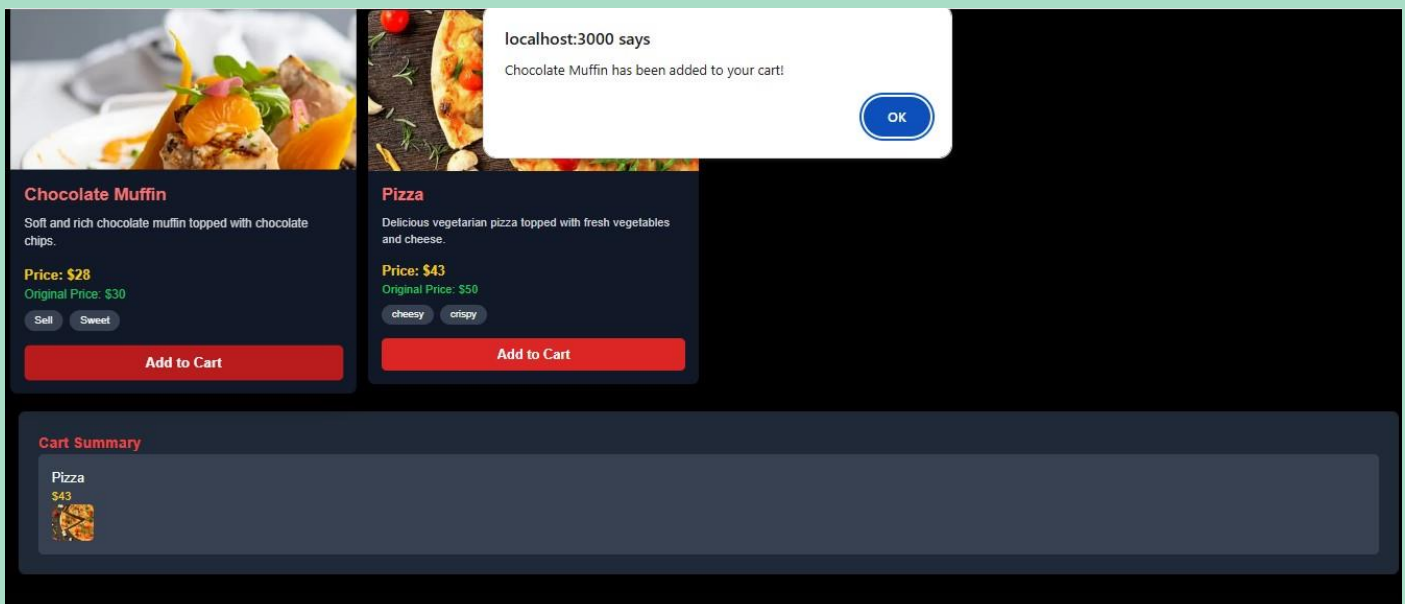
After running this command I have clicked on localhost: 3000 to view the migration of sanity data on my frontend project and I have founded that my all data has been fetched successfully.

The screenshot shows a web application interface for a food menu. At the top, there's a header with the title "OUR MENU" in yellow and the subtitle "Best Food for your Life" in red. Below the header, there are four food items displayed in a grid. Each item has a food image, a title, a description, a price, an original price, and an "Add to Cart" button. The items are: Country Burger (Classic country-style burger served with fries, Price: \$45, Original Price: \$50, Recommended), Burger (Juicy beef burger with fresh lettuce, tomatoes, and cheese, Price: \$21, Original Price: \$45, Popular), Fresh Lym (Refreshing fresh lime drink made with natural ingredients, Price: \$38, Original Price: \$45, Popular, Healthy), and Chicken Chup (Classic country-style burger served with fries, Price: \$12, Original Price: \$15, Sell, Crispy).



12th- Add to Cart Summary.

On click of every Add to cart button of items, they will be starting to add in my cart summary.



SUMMARY.

The Hackathon Day-3 task was to Integrate of provided API and Data Migration on Frontend of Next.js project. Which I have done successfully with following short summary.

- 1- **Installation of Sanity Studio.**
- 2- **Generating the Token number and save it in .env.local file with the project ID.**
- 3- **Define schema type of Food and made script folder & file to integrate the API data in Sanity and also run the command after this "npm run import-data".**
- 4- **Checking the GROQ Query in sanity vision to migrate the data on frontend.**
- 5- **Made a Food folder in src/app route & fetch the data with the help of map method, useEstate & useEffect then run the command npm run dev for localhost: 3000.**
- 6- **Finally & Al-Hamdulillah My 3rd Day task has completed successfully and I have fetched the data on my Next.js project Frontend.**