
Hackathon

Day-03

Api integration & Data migration

By:uzma thursday 9 to 12

EXPLORER

UI_UX_HACKATHONE_PRO...

src

components

whywecho.tsx

sanity

lib

client.ts

image.ts

live.ts

schemaTypes

food.ts

index.ts

shef.ts

env.ts

structure.ts

.env.local

.eslintrc.json

gitignore

import-data.mjs

.env.local

gridshet.tsx

tsconfig.json

next.config.mjs

client.ts

shef.ts

1

2

3

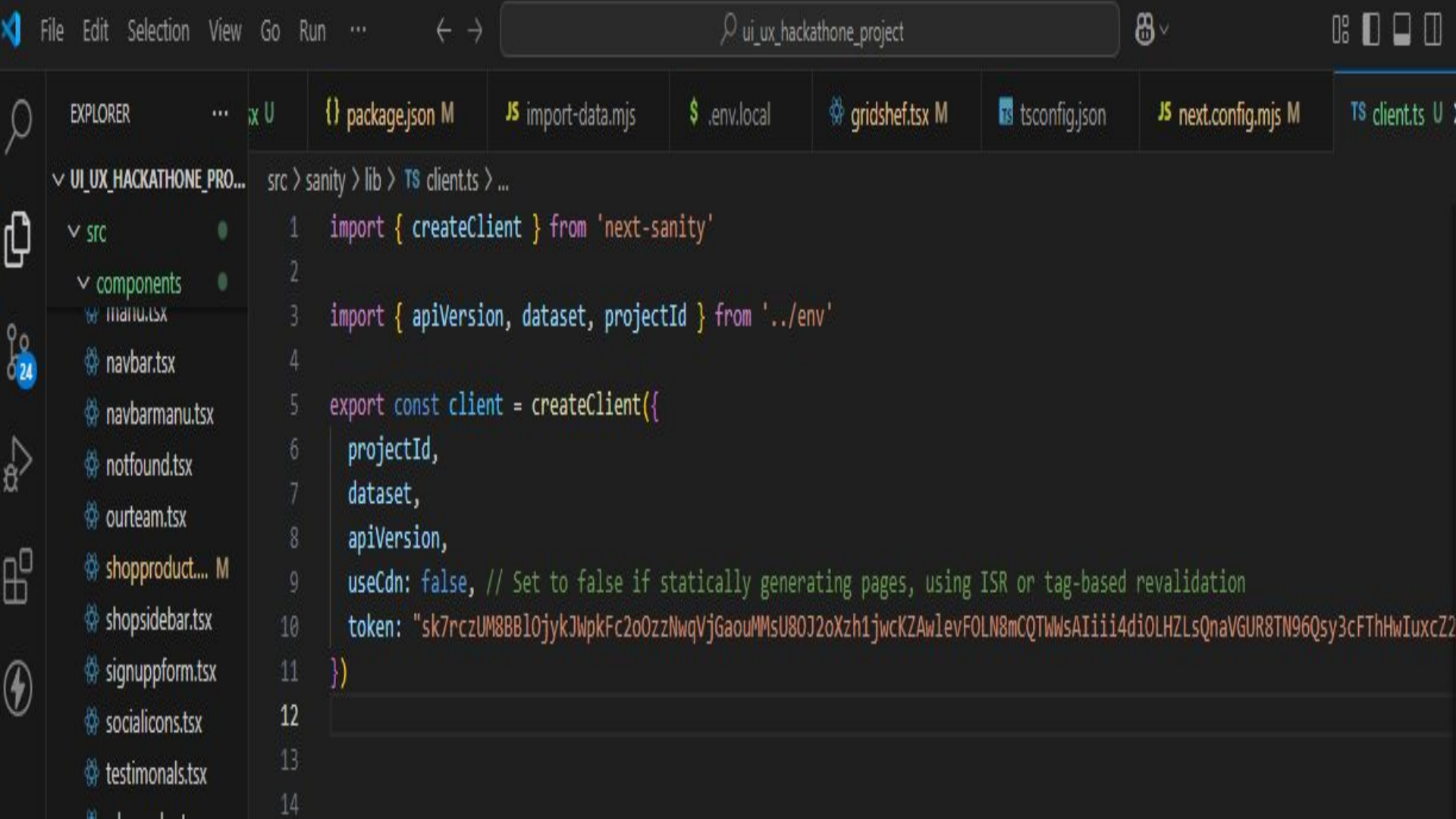
4

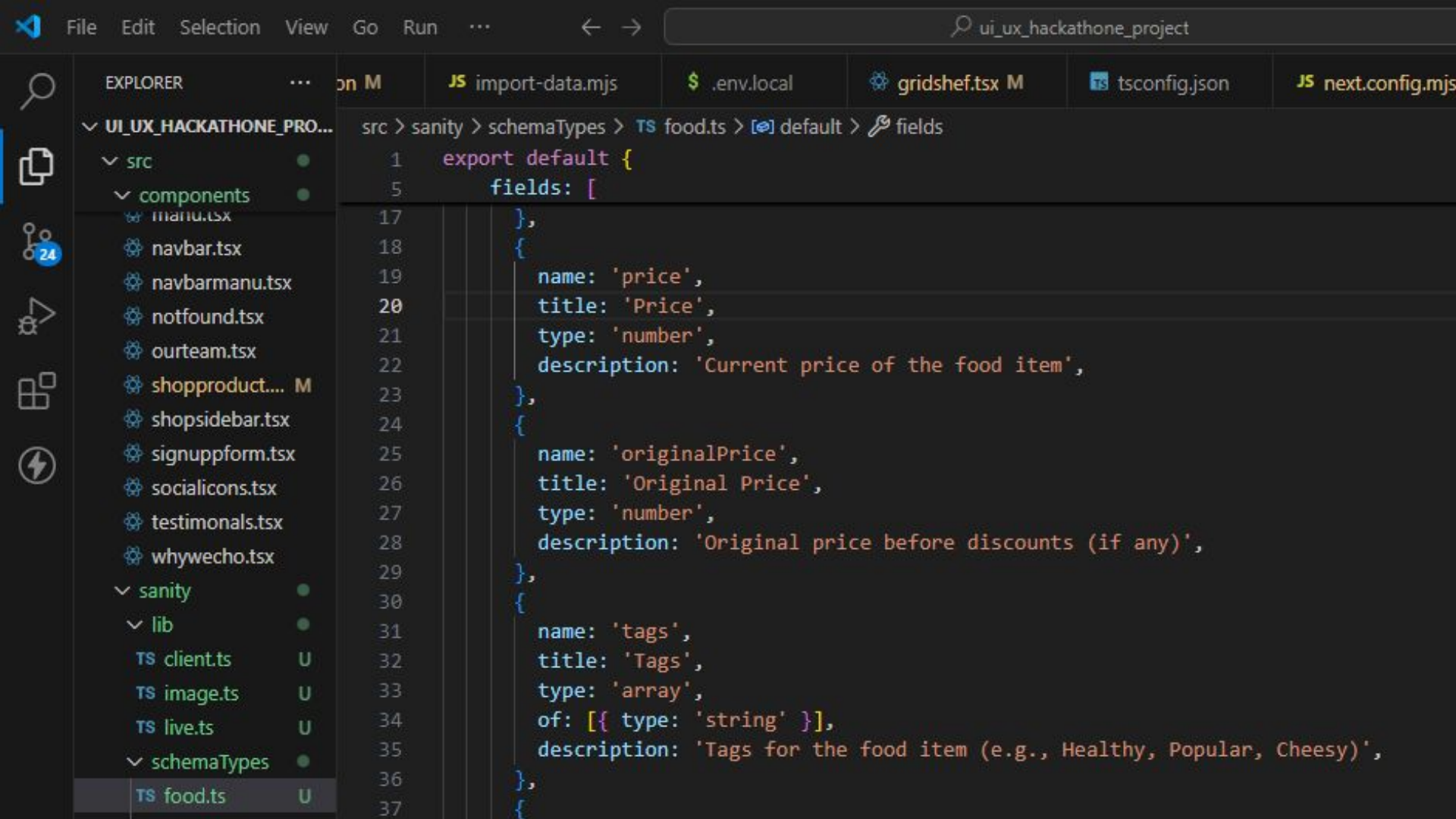
\$.env.local

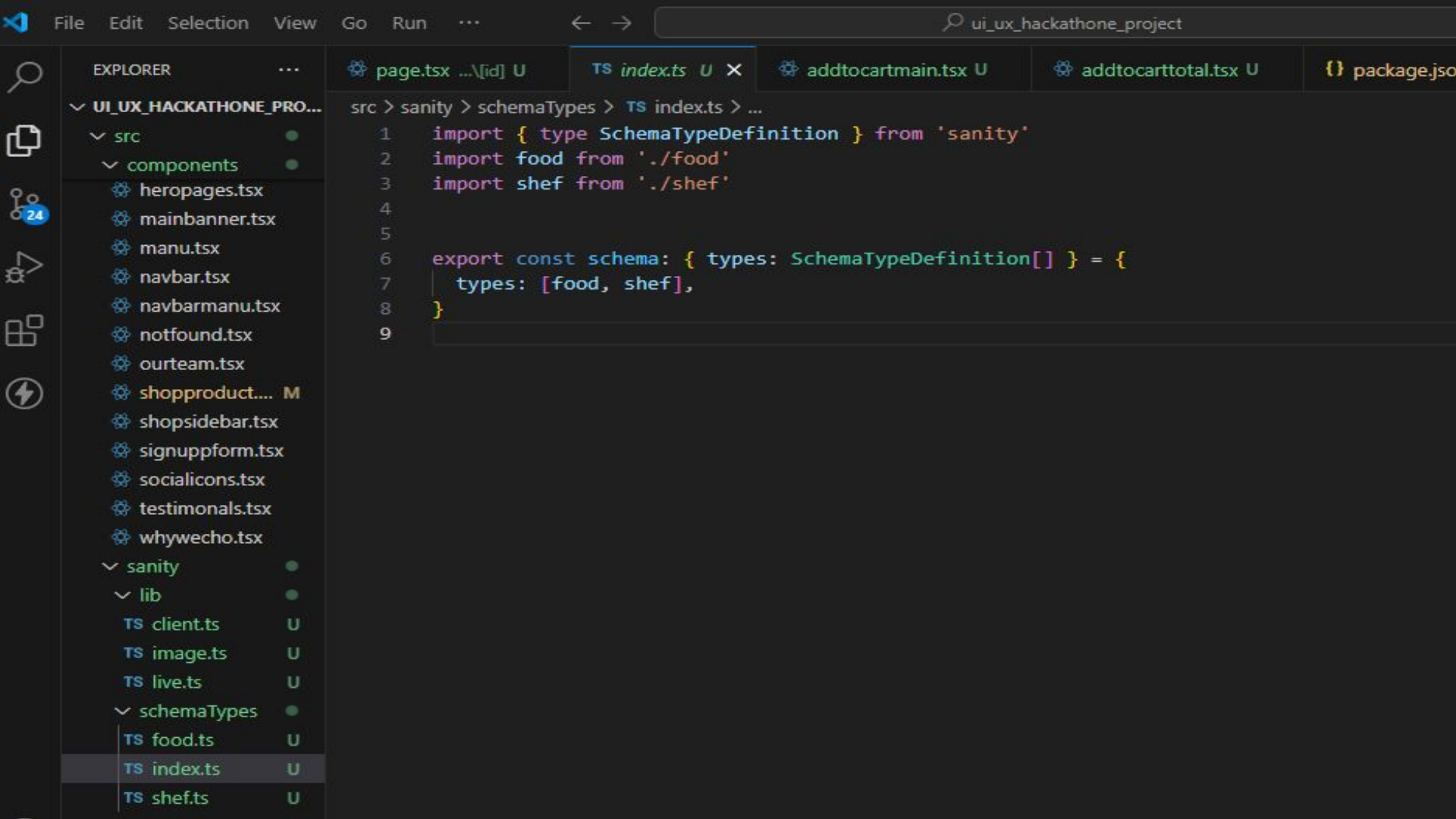
NEXT_PUBLIC_SANITY_PROJECT_ID="7wt2jwn4"

NEXT_PUBLIC_SANITY_DATASET="production"

NEXT_PUBLIC_SANITY_API_TOKEN="sk7rczUM8BB10jykJWpkFc2oOzzNwqVjGaouMMsU80J2oXzh1jwcKZAwIevFOLN8mCQTWwsAIiii4diOLHZLsQnaVGUR8TN96Qsy3cFThHwIuxcZ27VoaJsgSoYiWKSTkSKwa1PjSPfxssuvXTekKbuQkUcPRTMTKbZPcQmgTQpCzpHOH6QL"







EXPLORER

UI_UX_HACKATHONE_PRO...

src

components

heropages.tsx

mainbanner.tsx

manu.tsx

navbar.tsx

navbarmanu.tsx

notfound.tsx

ourteam.tsx

shopproduct.... M

shopsidebar.tsx

signupform.tsx

socialicons.tsx

testimonials.tsx

whywecho.tsx

sanity

lib

TS client.ts U

TS image.ts U

TS live.ts U

schemaTypes

TS food.ts U

TS index.ts U

TS shef.ts U

page.tsx ...[id] U

TS index.ts U X

addtocartmain.tsx U

addtocarttotal.tsx U

package.json

src > sanity > schemaTypes > TS index.ts > ...

```
1 import { type SchemaTypeDefinition } from 'sanity'
2 import food from './food'
3 import shef from './shef'
4
5
6 export const schema: { types: SchemaTypeDefinition[] } = {
7   types: [food, shef],
8 }
9
```


EXPLORER

UI_UX_HACKATHONE_PRO...

src

components

manu.tsx

navbar.tsx

navbarmanu.tsx

notfound.tsx

ourteam.tsx

shopproduct.... M

shopsidebar.tsx

signuppform.tsx

socialicons.tsx

testimonials.tsx

whywecho.tsx

sanity

lib

TS client.ts

TS image.ts

JS import-data.mjs

\$.env.local

gridshf.tsx M

tsconfig.json

src > sanity > schemaTypes > TS shf.ts > default > fields > options

```
1 export default {
2   name: 'chef',
3   title: 'Chef',
4   type: 'document',
5   fields: [
6     {
7       name: 'name',
8       title: 'Name',
9       type: 'string',
10    },
11    {
12      name: 'image',
13      title: 'Image',
14      type: 'image',
15      options: {
16        hotspot: true,
17      },
18    },
19    {
20      name: 'speciality'
```

File Edit Selection View Go Run ... ui_ux_hackathone_project

EXPLORER

UI_UX_HACKATHONE_PRO...

- src
 - components
 - banner.tsx
 - blog.tsx
 - blogpost.tsx
 - chouseus.tsx
 - cta.tsx
 - errorsignin.tsx
 - foodcatagory.tsx
 - footer1.tsx
 - footer2.tsx
 - footer3.tsx
 - gridshef.tsx M
 - header2.tsx
 - heading.tsx
 - hero.tsx
 - hero2manu.tsx
 - hero3manu.tsx
 - hero4manu.tsx

src > components > gridshef.tsx > ...

```
1  "use client"
2
3  import { client } from '@sanity/lib/client';
4  import Image from 'next/image';
5  import { useEffect, useState } from 'react';
6
7  export default function Gridshef() {
8    const [chefs, setChefs] = useState<any[]>([]);
9
10   // Function to fetch chef data
11   const getChefsData = async () => {
12     try {
13       // Fetch the chef data from Sanity using groq query
14       const chefsData = await client.fetch(`*[_type == "chef"]{
15         _id,
16         name,
17         position,
18         experience,
19         specialty,
20         description,
21         available,
22         "imageUrl": image.asset->url
```

- ✓ components
- ✎ gridshef.tsx M
- ✎ header2.tsx
- ✎ heading.tsx
- ✎ hero.tsx
- ✎ hero2manu.tsx
- ✎ hero3manu.tsx
- ✎ hero4manu.tsx
- ✎ hero5manu.tsx
- ✎ heropages.tsx
- ✎ mainbanner.tsx
- ✎ manu.tsx
- ✎ navbar.tsx
- ✎ navbarmanu.tsx
- ✎ notfound.tsx
- ✎ ourteam.tsx
- ✎ shopproduct.... M
- ✎ shopsidebar.tsx
- ✎ signuppform.tsx
- ✎ socialicons.tsx
- ✎ testimonials.tsx
- ✎ whywecho.tsx
- ✓ sanity
- ✓ lib
- TS_client.ts

OUTLINE

MFINE

```
21 const [loading, setLoading] = useState<boolean>(true);
22
23 const query = `*[_type == "food"] {
24   name,
25   price,
26   originalPrice,
27   "image": image.asset->url,
28   description,
29   tags,
30   category,
31   available,
32   _id
33 }`;
34
35 const fetchProducts = async () => {
36   try {
37     const data: Product[] = await client.fetch(query);
38     setProducts(data.slice(0, 20)); // Get only the first 20 items
39     setLoading(false);
40   } catch (error) {
41     console.error("Error fetching products:", error);
42   }
43 };
44
45 useEffect(() => {
46   fetchProducts();
47 }, []);
48
49 if (loading) {
50   return <p className="text-center mt-10">Loading products...</p>;
51 }
```


File

Edit

Selection

View

Go

Run

...

←

→

sanity-nextjs-master

EXPLORER

...

▼ SANITY-NEXTJS-MASTER

▼ src

▼ sanity

▼ lib

TS live.ts

▼ schemaTypes

TS chefs.ts

TS foods.ts

TS index.ts

TS env.ts

TS structure.ts

▼ scripts

JS import-data.mjs

\$.env.local

.gitignore

JS eslint.config.mjs

TS next.config.ts

{ } package-lock.json

{ } package.json

JS postcss.config.mjs

① README.md

TS sanity.cli.ts

TS sanity.config.ts

TS tailwind.config.ts

tsconfig.json

{ } tsconfig.tsbuildinfo

> OUTLINE

> TIMELINE

TS client.ts

\$.env.local

JS import-data.mjs X

TS live.ts

src > scripts > JS import-data.mjs > client > token

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: '7wt2jwn4',

15

dataset: 'production',

16

useCdn: true,

17

token:

"sk7rczUM8BB10jykJWpkFc2o0zzNwqVjGaouMMsU80J2oXzh1jwcKZAwlevFOLN8mCQTWwsAIiii4diOLHZLsQnaVGUR8T

gSoYiWKSTkSKwalPjSPfxssuvXTeKkbuQkUcPRMTkKbZPcQmgTQpCzpHOH6QL",

apiVersion: '2021-08-31',

18

});

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

File

Edit

Selection

View

Go

Run

...

sanity-nextjs-master

EXPLORER

...

▼ SANITY-NEXTJS-MASTER

▼ src

▼ sanity

▼ lib

TS live.ts

▼ schemaTypes

TS chefs.ts

TS foods.ts

TS index.ts

TS env.ts

TS structure.ts

▼ scripts

JS import-data.mjs

\$.env.local

.gitignore

JS eslint.config.mjs

TS next.config.ts

{ } package-lock.json

{ } package.json

JS postcss.config.mjs

① README.md

TS sanity.cli.ts

TS sanity.config.ts

TS tailwind.config.ts

tsconfig.json

{ } tsconfig.tsbuildinfo

> OUTLINE

> TIMELINE

TS client.ts

\$.env.local

JS import-data.mjs X

TS live.ts

src > scripts > JS import-data.mjs > importData

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

async function importData() {

try {

console.log('Fetching food, chef data from API...');

// API endpoint containing data

const \$Promise = [];

\$Promise.push(

axios.get('https://sanity-nextjs-rouge.vercel.app/api/foods')

);

\$Promise.push(

axios.get('https://sanity-nextjs-rouge.vercel.app/api/chefs')

);

const [foodsResponse, chefsResponse] = await Promise.all(\$Promise);

const foods = foodsResponse.data;

const chefs = chefsResponse.data;

console.log("🍴",chefs)

for (const food of foods) {

console.log(`Processing food: \${food.name}`);

let imageRef = null;

if (food.image) {

imageRef = await uploadImageToSanity(food.image);

}

const sanityFood = {

_type: 'food',

name: food.name,



+ Create



Structure

Visi

Content



Food



Chef



Food



Search list



Chicken Chup



Pizza



Country Burger



Burger



Chocolate Muffin



Fresh Lime



+ Create



Structure

Vision

Sched

Content



Food



Chef



Chef



Search list



William Rumi



Bisnu Devgon



Munna Kathy



M. Mohammad



Jorina Begum



Tahmina Rumi

+ Create

Q

StructureVisionSchedules

DATASET

API VERSION

CUSTOM API VERSION

PERSPECTIVE ?

QUERY URL [COPY TO CLIPBOARD]

production

Other

v2025-01-14

raw

https://7wt2jwn4.api.sanity.io/v2025-01-14/data/quer

QUERY

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

PARAMS

```
{
}
```

RESULT

```
[...] 6 items
  0: {...} 9 properties
    description: Refreshing fresh lime drink made with natural ingredients.
    category: Drink
    name: Fresh Lime
    price: 38
    originalPrice: 45
    image: https://cdn.sanity.io/images/7wt2jwn4/production/e155a50cdbcd0e02ca81649945bd30a2fc111276-1248x10
    tags: [...] 2 items
      0: Healthy
      1: Popular
    available: true
    _id: YTjpX7gIRs7VphrWQPhdpg
  1: {...} 9 properties
    description: Soft and rich chocolate muffin topped with chocolate chips.
    _id: YTjpX7gIRs7VphrWQPhdul
    originalPrice: 30
```




Fresh Lime

Price: \$38

Discount: 15.55555555555555%

Category: Drink

[View Details](#)



Chocolate Muffin

Price: \$28

Discount: 6.666666666666667%

Category: Dessert

[View Details](#)





M. Mohammad

Grill Master

Grilled Dishes

10 years of experience

Renowned for creating perfectly grilled meats and vegetables.



Munna Kathy

Culinary Instructor

Asian Fusion

15 years of experience

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



Bisnu Devgon

Executive Chef

Global Cuisine

20 years of experience

Expert in international cuisines and menu planning.



Tahmina Rumi

Head Chef

Italian Cuisine

12 years of experience

Expert in crafting authentic Italian dishes and pastries.





Burger

Juicy beef burger with fresh lettuce, tomatoes, and cheese.

\$21.00

Discount: 53.33%

Category: Sandwich

Add to Cart