Hackathon Day 03

API-INTERGRATION AND MIGRATION

Hackathon Day 3, where we migrated data, created schemas, built APIs, imported data into Sanity CMS, and displayed the data on a front-end application using Next.js.

1. Data Migration

Objective:

To migrate data from an API source and prepare it for use in our application..

Steps:

- 1. Write a script to fetch the data from the API.
- 2.Example: https://sanity-nextjs-rouge.vercel.app/api/foods'

Write a script to fetch the data from the API..

. Use libraries like axios or node-fetch in your JavaScript/Node.js script.

Example:

```
- Harry Hunchitton Importmenta() (
                                        consule.log "fetching food, thef data from MT...");
                                        // Wi cropalit containing data
                                       comt strucks: - (1)
                                       $Proxise.push
 2 dits
                                        axios_get['https://wesity-mextjs-rouge_vercel_app/epi/foods')
                                       Stronise push
                                         exice.get('nttpu://wardty-mestjs-rouge.vercel.app/api/shefs')
  TV privis
                                       court [fondsResponse, rhefsResponse] - must Promise.all($Promise);
                                       commt foods = foodsResponse.data;
commt chefa = chefaResponse.data;
                                       for (comit foot of foods)
  .g tignore
                                         conside.log("Frocessing food: 1/food.name)");
& esint.mmig.rrjs.
                                          list imageRef + null;
is necticenfielts
(1) package-lock/son
                                            imageRef = minit uploadImageToSan[ty(food.image];
# paylosconfigures
                                         omst smityhout = [
III READWEARD
to sanituality
                                           name: food name,
                                           category: food.category | | wall,
to be invivided to the last
                                           price: food.price.
                                           originalPrice: food.originalPrice | mil.
                                            tegn: food.tegn | [].
                                            description modulescription || ++, available: food.available | +- underfront | food.available | +- underfront | food.available | +- underfront |
                                             insuct insuchet
                                                   _tyse: Himse's
                                                     _type: 'meference',
CUTUM
                                                       _rer imageher,
```

Clean and transform the data (if necessary) to match the desired format. Save the transformed data into a format compatible with Sanity (e.g., JSON).

2. Creating the Schema

To define the structure of the data in Sanity CMS.

Steps:

1.Install Sanity CLI and initialize a new Sanity project: npm install -g @sanity/cli

Define a schema for the data in the schemas folder of your Sanity project

Example Sehema:

```
SOURCE CONTROL ....
                            (1) food/sor X
SAMETY-MEX_ C: CT () () food(non ) __
                                        "mame": "Fresh Line",
 > public
                                        "category": "Drine",
                                        "price": NLO,
  5 app
                                        "originalPrice": 45.8,
                                  "tags": "Teatiny", "Dopoler"),
"leage": "https://wenity-mextjs-rouge.vertel.app/food/food-1.prg",
"description": "Refreshing fresh lime drink made with natural ingredients.",
  w data
  (1) chefuson
  II food,ison
                                        available : too
  ~ sanity
   > 46
                             "name": "(hornlate Putfin",
"cetegory": "Dessert",
"price": 28.8,
"originalPrice": 38.8,
"tegs": ["Sell", "hasert"],
  > schemaTypes
  III emili
  m import-data.mis M
                          TO 18
                                      "image": "https://sanity-nextjs-rouge.vercel.app/food/food-2.prg",
                                        "description": "Soft and rich chocolate muffin topped with chocolate chips.",
 gilgnore
                                        "available": true
 A estint config mis
                                      "name": "Burger",
                                      "category": "Sandwich",
[] package-lockmen
                                       "price": 21.0,
                            "originalPrice": 45.8,
() perkege juan
 # pestessennigues
                                        "tegs": ["Popular"];
                                      timage": "attps://samity.nextis.rouge.vercei.spo/food/food 5.prg":
(II) README.md
                          10 (10)
                                      "description": "Dudry beef burger with fresh lettuce, tomatoes, and cheese.",
at its sanity of its
                                        "available": frue
taliwind.config.ts
tsconfig.json.
                                        "name": "Country Surger",
                                        "category": "Sandwich",
                                         "price": 45.0,
                                         "originalPrice": 50.8,
                                         "tegs": ["Recommended"];
                                         "image": "https://sanity-next/s-rouge.vercel.app/food/food-4.prg",
OUTLINE
                                         "description": "Classic country-style burger served with fries."
 THUS IN
```

3. Building the API

Objective:

Steps:

Create a new API route in your Next.js application under the pages/api directory.

```
SOLIRCA CONTROL
SAMITY-NDOOS
                        art ) app ) component ) III foodfatch (bx ) III Product/legs
                              import React from 'react';
                              import ( client ) from '8/sminy/linyclient';
                              import image From 'nest/image';
> public
                         5 (105) FroductPage = 450°C () =0 -
                              f type - food ||
  3 40
                                 _id;
                                grice;
discountPercentage,
  a dudlo
                                  category,
sescription,
systemics
 taxicon ins
 # globals.css
                                    "imageled": image.ecost-ourl
 3" data
 or confly
  3. schemallypes
                                   this class@ee*o 6 Bbg gray 100 min h screen?
 III wrote
                                   (01 classiones*text 4x1 font-bold text-center Ditext gray 890 no 87)
 Till ofreclure/fo
                                   only classifies grid grid-only-1 swaptio-only-2 tapgrid-only-2 gap-9
                                       (query-men((front any, indext number) -> (
                                     classWate-" I be white shadow-ad nounded-lg swerflow-hidden hower:shadow-lg transition-shadow duration-100"
15 salimbourriours
                                           case classWomer to full h-[NAGys] flox justify-center items-center | bg-gray-100*
                                             secs(root.fragetri | '/placetobler.prg') // the slacerabler large it largetri is whatley
(1) package ison
                                              alt=(food.mame || 'Food Item')

    postoscorrigares

                                              width=(500)
                                               height=(500)
                                               classwee-"object-contain"
OUTLINE
```

4. Importing Data into Sanity CMS

- 1.Make a project on sanity .
- 2.Create a .env.local file in the root of the project directory:
- 3.Open .env.local and add the following environment variables:

```
NEXT_PUBLIC_SANITY_PROJECT_ID="{your-sanity-project-id}"

NEXT_PUBLIC_SANITY_DATASET="production"

SANITY_API_TOKEN="{your-sanity-api-token}"
```

Eample

.env.local file

```
** Swilly Mr. | T. P. C. P. S. Ancided

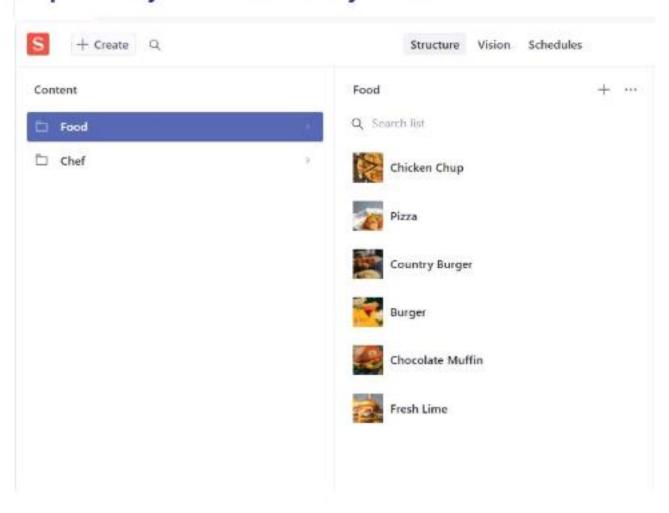
** Ancided | Next Printing Swilly Property Instruction**

** Depth | Swilly Mr. There's Appropriate Control of Con
```

Import Data into sanity cammand:

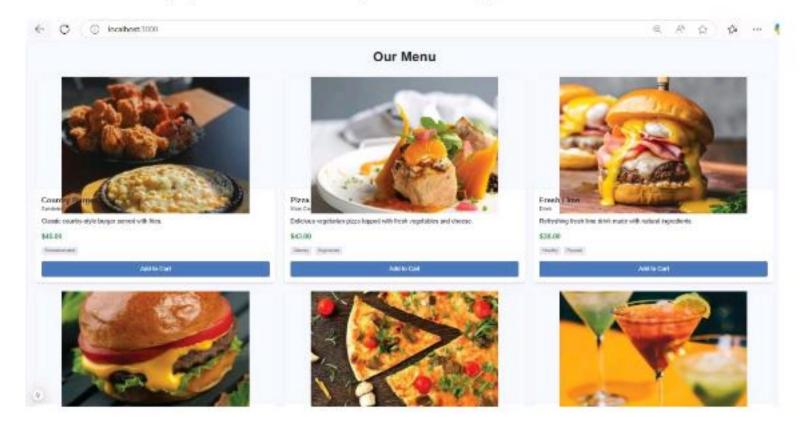
npm run import-data on sanity .
and npm run dev.
http://localhost:3000/studio

Step 5: Verify the Data in Sanity Studio



Step 6:Displaying Data on the Front-End

To fetch and display the data in a Next.js front-end application.



7. Conclusion

By following these steps, we successfully:

- · Migrated data from an API.
- · Created a schema in Sanity CMS.
- · Built a custom API to fetch data.
- · Imported data into Sanity.
- Displayed the data on the front-end using Next.js.

This workflow can be adapted for similar projects to streamline the process of data migration. API creation, and front-end integration.

THANK YOU

1212107707