Hackathon 03
Day 03 Task

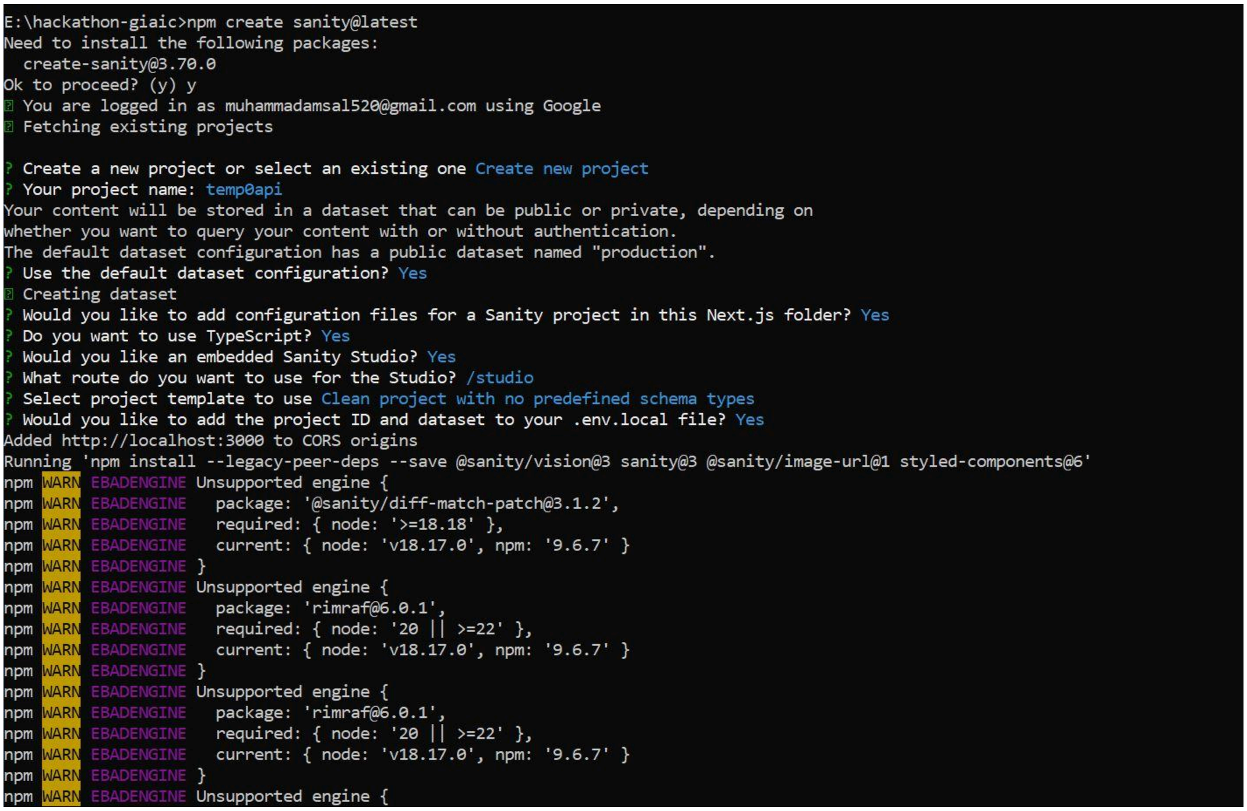
MUHAMMAD AMSAL   482160

This is my Day 3 GIAIC Hackathon task.

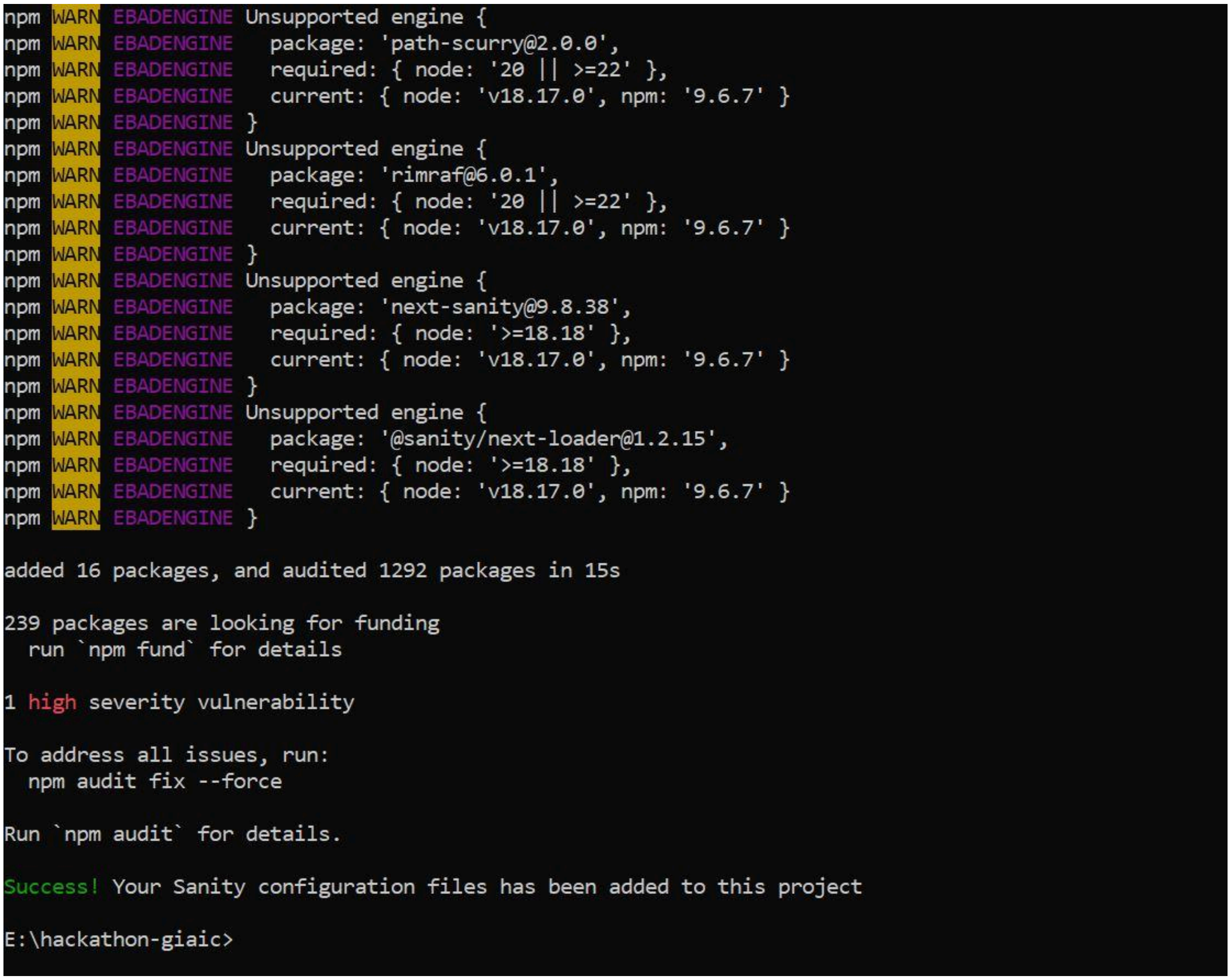
Sanity.io is a powerful headless CMS that allows you to structure content and serve it through an API. Here's how I set up my project:

- ## Install Sanity in the Project

First, I initialized a new Sanity project within my Next.js setup by running the following command:

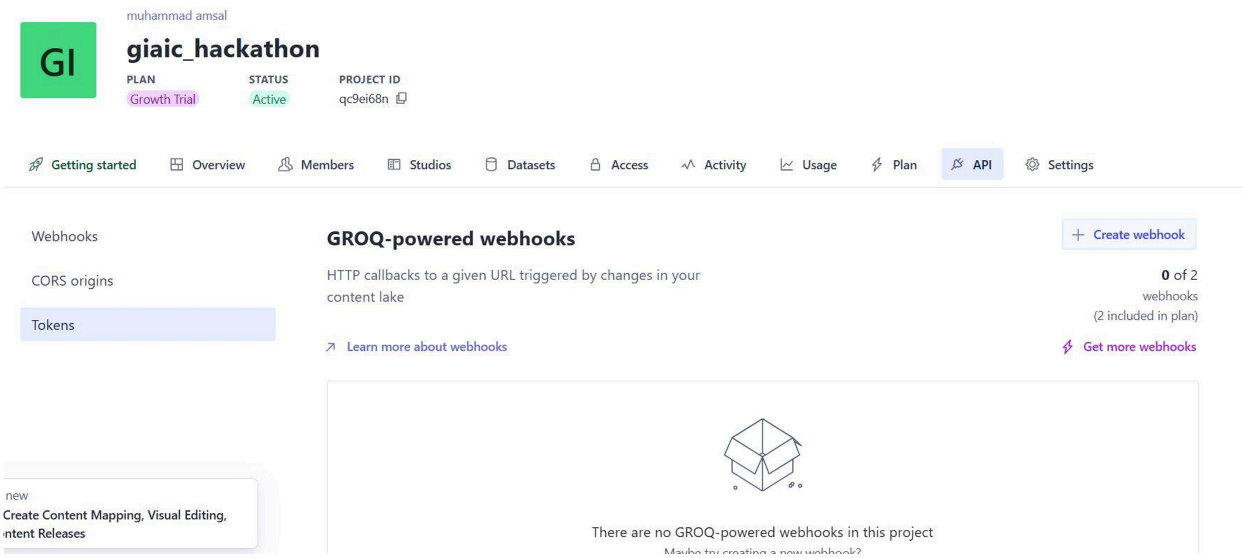


```
E:\hackathon-giaic>npm create sanity@latest
Need to install the following packages:
  create-sanity@3.70.0
Ok to proceed? (y) y
□ You are logged in as muhammadamsal520@gmail.com using Google
□ Fetching existing projects

? Create a new project or select an existing one Create new project
? Your project name: temp0api
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
? 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
Running 'npm install --legacy-peer-deps --save @sanity/vision@3 sanity@3 @sanity/image-url@1 styled-components@6'
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: '@sanity/diff-match-patch@3.1.2',
npm WARN EBADENGINE   required: { node: '>=18.18' },
npm WARN EBADENGINE   current: { node: 'v18.17.0', npm: '9.6.7' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'rimraf@6.0.1',
npm WARN EBADENGINE   required: { node: '20 || >=22' },
npm WARN EBADENGINE   current: { node: 'v18.17.0', npm: '9.6.7' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'rimraf@6.0.1',
npm WARN EBADENGINE   required: { node: '20 || >=22' },
npm WARN EBADENGINE   current: { node: 'v18.17.0', npm: '9.6.7' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
```



```
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'path-scurry@2.0.0',
npm WARN EBADENGINE   required: { node: '20 || >=22' },
npm WARN EBADENGINE   current: { node: 'v18.17.0', npm: '9.6.7' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'rimraf@6.0.1',
npm WARN EBADENGINE   required: { node: '20 || >=22' },
npm WARN EBADENGINE   current: { node: 'v18.17.0', npm: '9.6.7' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'next-sanity@9.8.38',
npm WARN EBADENGINE   required: { node: '>=18.18' },
npm WARN EBADENGINE   current: { node: 'v18.17.0', npm: '9.6.7' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: '@sanity/next-loader@1.2.15',
npm WARN EBADENGINE   required: { node: '>=18.18' },
npm WARN EBADENGINE   current: { node: 'v18.17.0', npm: '9.6.7' }
npm WARN EBADENGINE }

added 16 packages, and audited 1292 packages in 15s

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

1 high severity vulnerability

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

E:\hackathon-giaic>
```

Make a Project in Sanity



muhammad amsal

giaic_hackathon

PLAN: Growth Trial

STATUS: Active

PROJECT ID: qc9ei68n

GROQ-powered webhooks

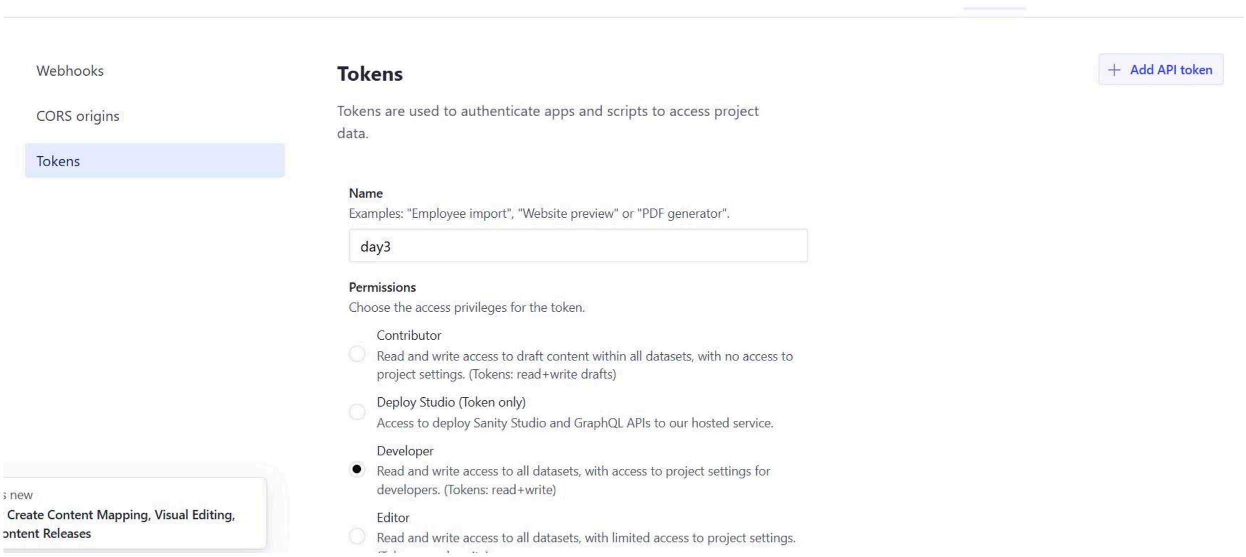
HTTP callbacks to a given URL triggered by changes in your content lake

0 of 2 webhooks (2 included in plan)

There are no GROQ-powered webhooks in this project

Maybe try creating a new webhook?

new
Create Content Mapping, Visual Editing,
ntent Releases



Tokens

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

Name
Examples: "Employee import", "Website preview" or "PDF generator".

day3

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)

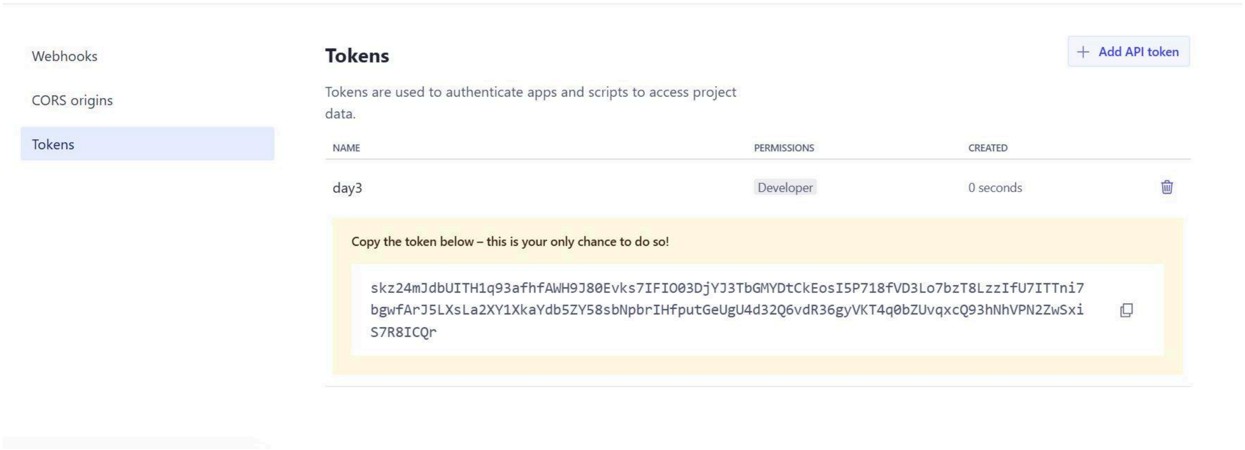
○ Deploy Studio (Token only)
Access to deploy Sanity Studio and GraphQL APIs to our hosted service.

● 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.

s new
Create Content Mapping, Visual Editing,
ontent Releases

Generate a Token



Tokens

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

NAME	PERMISSIONS	CREATED
day3	Developer	0 seconds

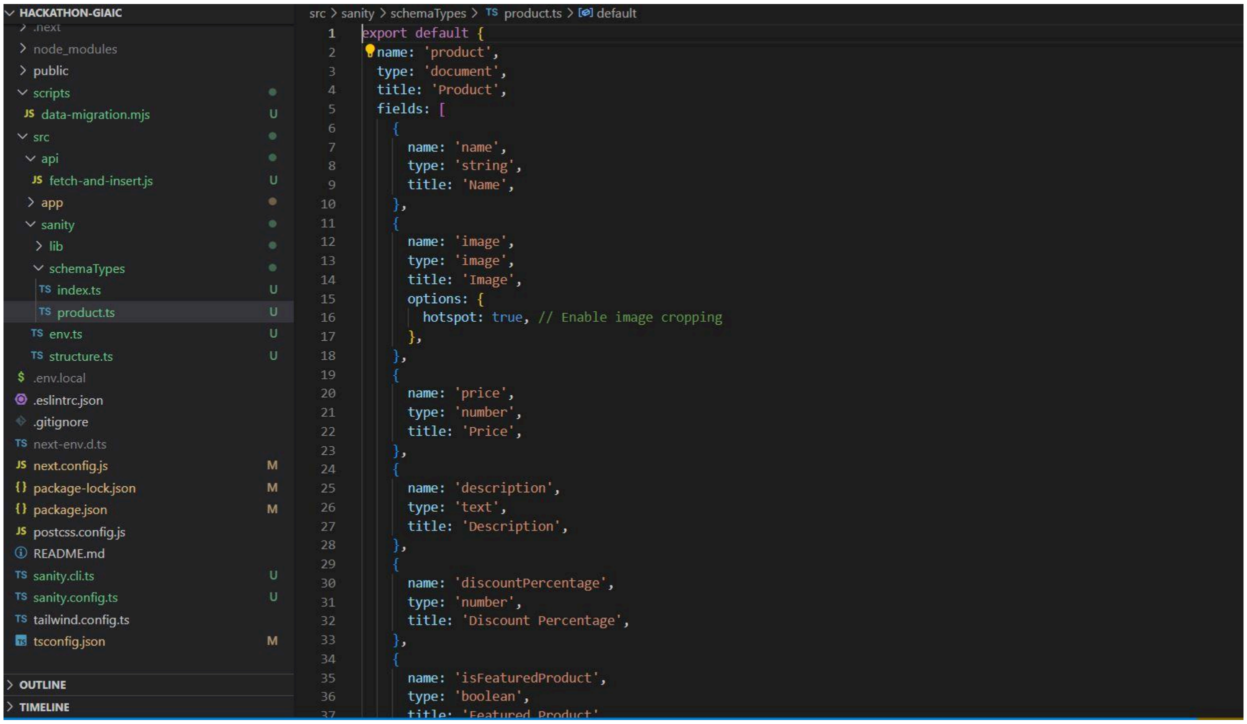
Copy the token below – this is your only chance to do so!

```
skz24mJdbUITH1q93afhfAWH9J80Evks7IFIO03DjYJ3TbGMYDtCkEosI5P718fVD3Lo7bzT8LzzIfU7ITTni7
bgwfArJ5LXsLa2XY1XkaYdb5ZY58sbNpbrIHfputGeUgU4d32Q6vdR36gyVKT4q0bZUvqxcQ93hNhVPN2ZwSxi
S7R8ICQr
```

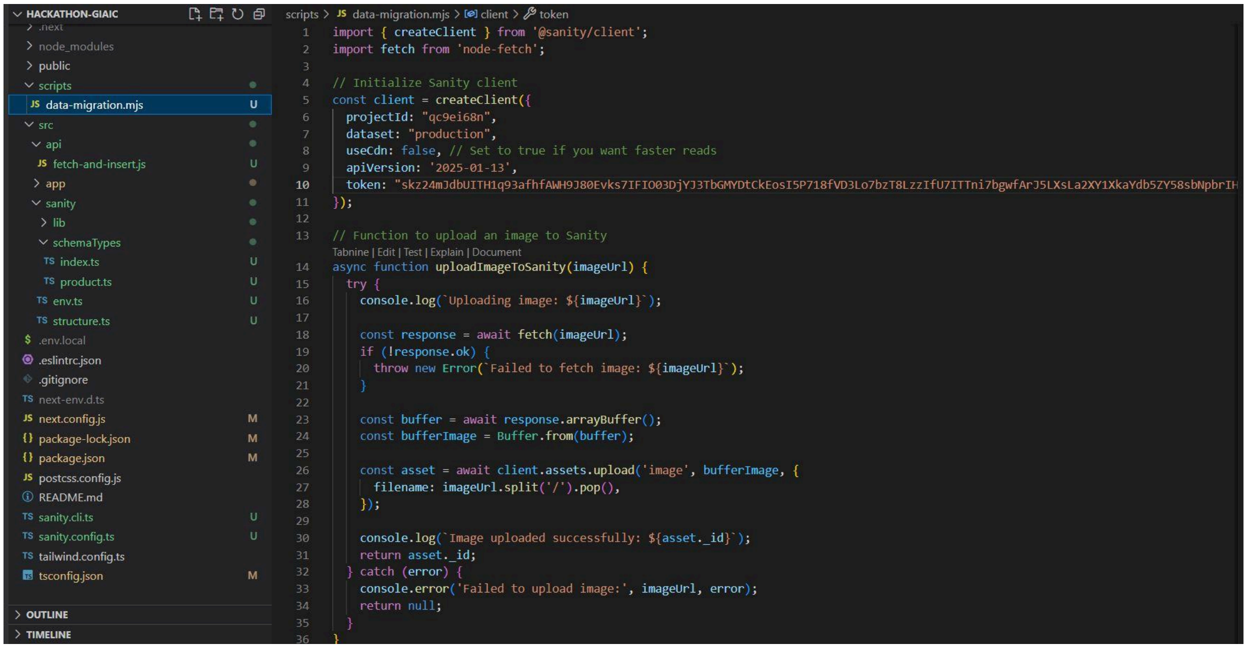
- **MIGRATING DATA TO SANITY**

Once the project was initialized, I proceeded to migrate existing content into Sanity. I created schemas to define the content structure:

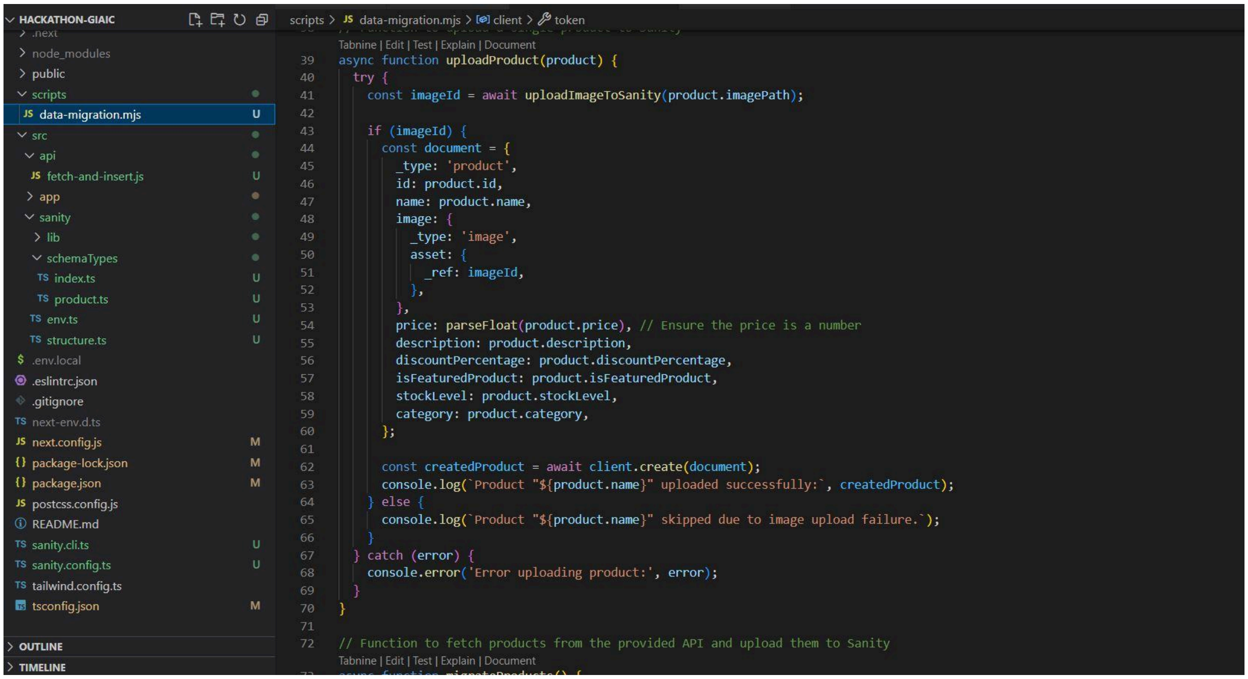
Create a Schema



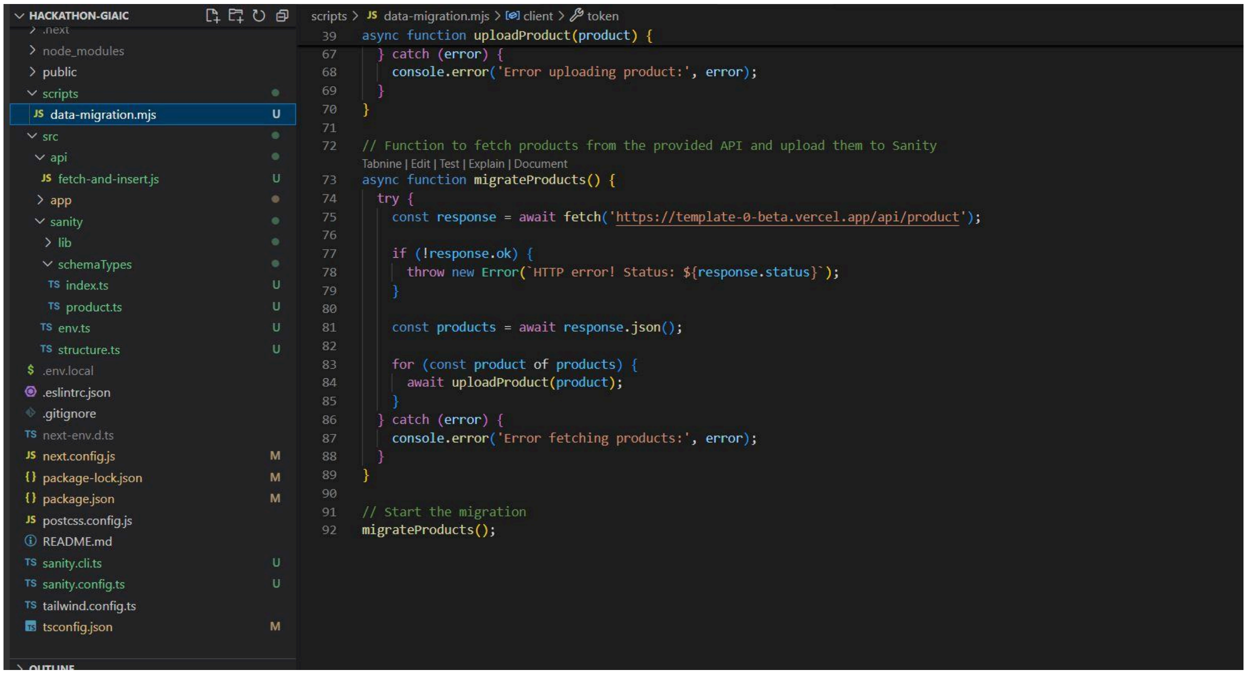
```
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Name',
    },
    {
      name: 'image',
      type: 'image',
      title: 'Image',
      options: {
        hotspot: true, // Enable image cropping
      },
    },
    {
      name: 'price',
      type: 'number',
      title: 'Price',
    },
    {
      name: 'description',
      type: 'text',
      title: 'Description',
    },
    {
      name: 'discountPercentage',
      type: 'number',
      title: 'Discount Percentage',
    },
    {
      name: 'isFeaturedProduct',
      type: 'boolean',
      title: 'Featured Product',
```



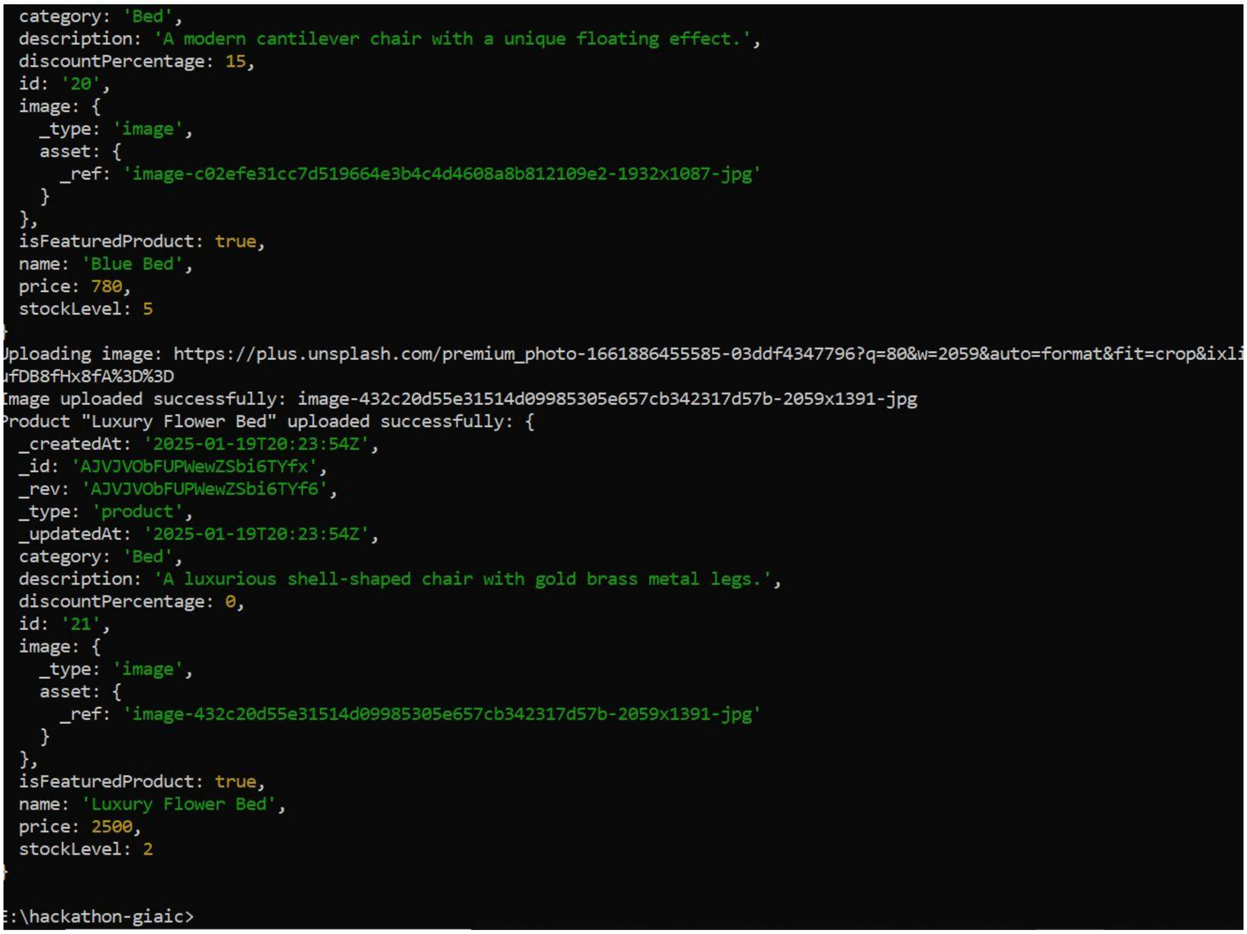
```
import { createClient } from '@sanity/client';
import fetch from 'node-fetch';

// Initialize Sanity client
const client = createClient({
  projectId: "qc9ei68n",
  dataset: "production",
  useCdn: false, // Set to true if you want faster reads
  apiVersion: '2025-01-13',
  token: "skz24mJdbUITH1q93afhfAWH9J80Evks7IFIO03DjYJ3TbGMYDtCkEosI5P718fVD3Lo7bzT8LzzIfU7ITTni7bgwfArJ5LXsLa2XY1XkaYdb5ZY58sbNpbrIH
});

// Function to upload an image to Sanity
Tabnine | Edit | Test | Explain | Document
async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    const response = await fetch(imageUrl);
    if (!response.ok) {
      throw new Error(`Failed to fetch image: ${imageUrl}`);
    }

    const buffer = await response.arrayBuffer();
    const bufferImage = Buffer.from(buffer);

    const asset = await client.assets.upload('image', bufferImage, {
      filename: imageUrl.split('/').pop(),
    });

    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}
```



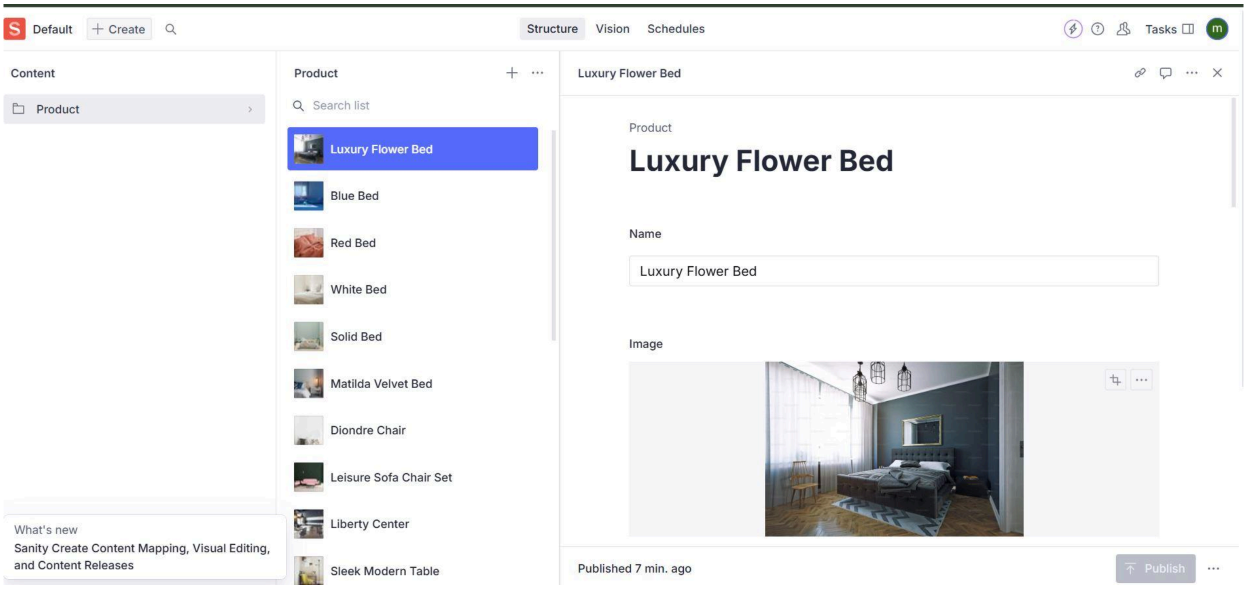
```
// Function to upload a single product to Sanity
Tabnine | Edit | Test | Explain | Document
async function uploadProduct(product) {
  try {
    const imageId = await uploadImageToSanity(product.imagePath);

    if (imageId) {
      const document = {
        _type: 'product',
        id: product.id,
        name: product.name,
        image: {
          _type: 'image',
          asset: {
            _ref: imageId,
          },
        },
        price: parseFloat(product.price), // Ensure the price is a number
        description: product.description,
        discountPercentage: product.discountPercentage,
        isFeaturedProduct: product.isFeaturedProduct,
        stockLevel: product.stockLevel,
        category: product.category,
      };

      const createdProduct = await client.create(document);
      console.log(`Product "${product.name}" uploaded successfully:`, createdProduct);
    } else {
      console.log(`Product "${product.name}" skipped due to image upload failure.`);
    }
  } catch (error) {
    console.error('Error uploading product:', error);
  }
}

// Function to fetch products from the provided API and upload them to Sanity
Tabnine | Edit | Test | Explain | Document
async function migrateProducts() {
```



```
async function uploadProduct(product) {
  } catch (error) {
    console.error('Error uploading product:', error);
  }
}

// Function to fetch products from the provided API and upload them to Sanity
async function migrateProducts() {
  try {
    const response = await fetch('https://template-0-beta.vercel.app/api/product');

    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }

    const products = await response.json();

    for (const product of products) {
      await uploadProduct(product);
    }
  } catch (error) {
    console.error('Error fetching products:', error);
  }
}

// Start the migration
migrateProducts();
```



```
  category: 'Bed',
  description: 'A modern cantilever chair with a unique floating effect.',
  discountPercentage: 15,
  id: '20',
  image: {
    _type: 'image',
    asset: {
      _ref: 'image-c02efe31cc7d519664e3b4c4d4608a8b812109e2-1932x1087-jpg'
    }
  },
  isFeaturedProduct: true,
  name: 'Blue Bed',
  price: 780,
  stockLevel: 5
}
Uploading image: https://plus.unsplash.com/premium_photo-1661886455585-03ddf4347796?q=80&w=2059&auto=format&fit=crop&ixli
ufDB8fHx8fA%3D%3D
Image uploaded successfully: image-432c20d55e31514d09985305e657cb342317d57b-2059x1391-jpg
Product "Luxury Flower Bed" uploaded successfully: {
  _createdAt: '2025-01-19T20:23:54Z',
  _id: 'AJVJVObFUPWewZSbi6TYfx',
  _rev: 'AJVJVObFUPWewZSbi6TYf6',
  _type: 'product',
  _updatedAt: '2025-01-19T20:23:54Z',
  category: 'Bed',
  description: 'A luxurious shell-shaped chair with gold brass metal legs.',
  discountPercentage: 0,
  id: '21',
  image: {
    _type: 'image',
    asset: {
      _ref: 'image-432c20d55e31514d09985305e657cb342317d57b-2059x1391-jpg'
    }
  },
  isFeaturedProduct: true,
  name: 'Luxury Flower Bed',
  price: 2500,
  stockLevel: 2
}

E:\hackathon-giaic>
```



Product

Luxury Flower Bed

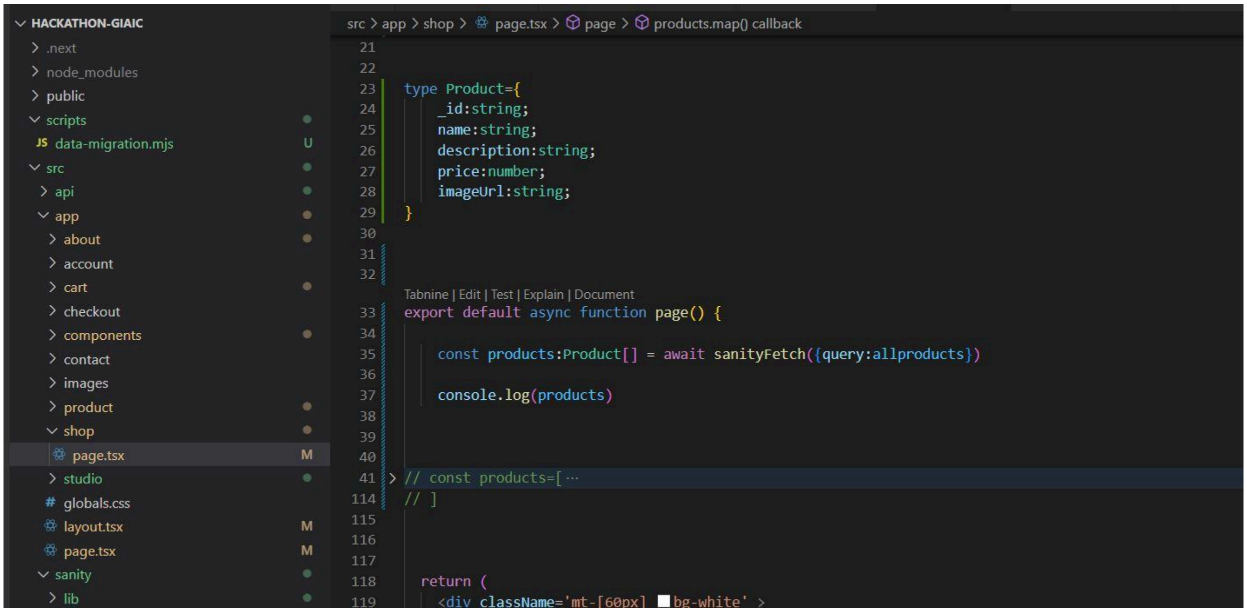
Name

Luxury Flower Bed

Image

Published 7 min. ago

● FETCHING DATA FROM SANITY IN NEXT JS



```
type Product={
    _id:string;
    name:string;
    description:string;
    price:number;
    imageUrl:string;
}



export default async function page() {

    const products:Product[] = await sanityFetch({query:allproducts})

    console.log(products)



// const products=[ ···
// ]



  return (
    <div className='mt-[60px] bg-white' >
```