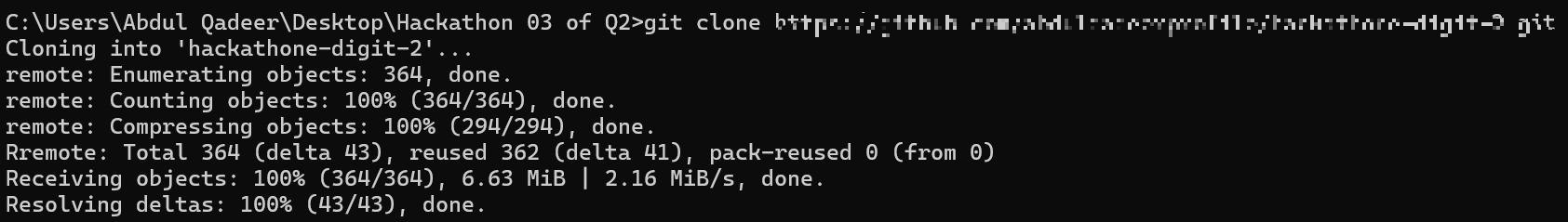
HACKATHON 3 (DAY 03)

API Integration and Data Migration

1. **Cloning the UI/UX Repository :**

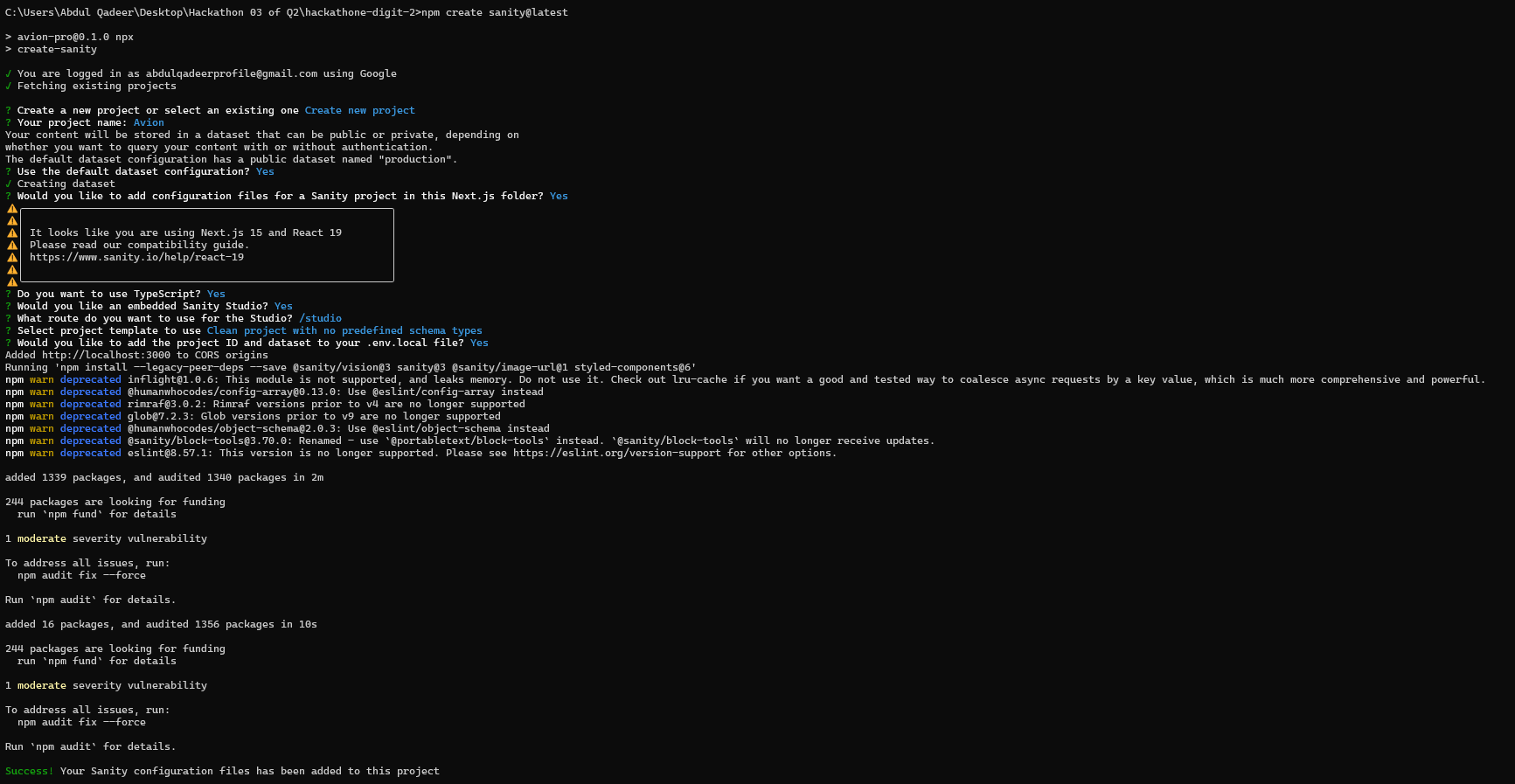
First of all I cloned my existing UI/UX repository which I have to work on by running **git clone [My Repo URL]**.



After cloning, run the **npm install** command to install all of the **node\_modules**.

1. **Installing Sanity :**

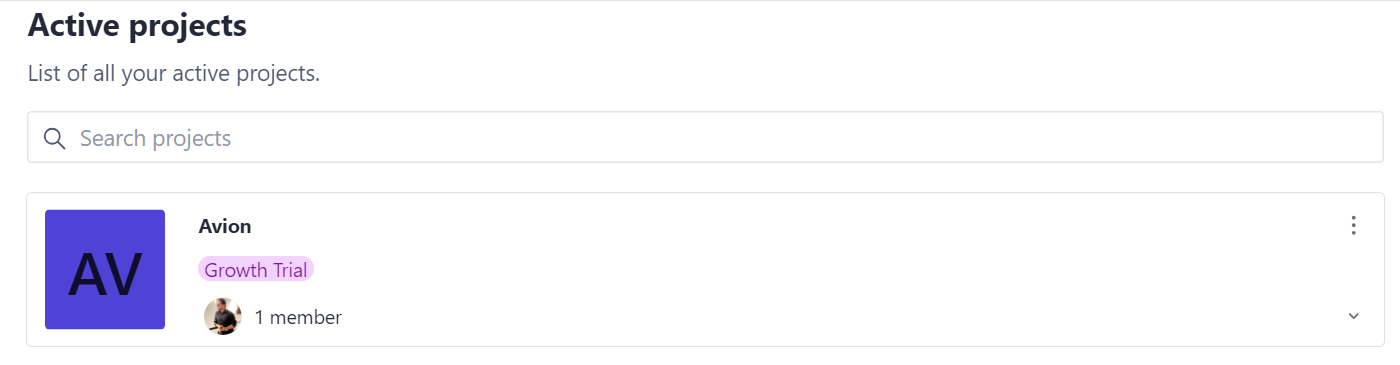
After cloning I ran **npm create sanity@latest** command to build the Sanity environment inside my UI directory.



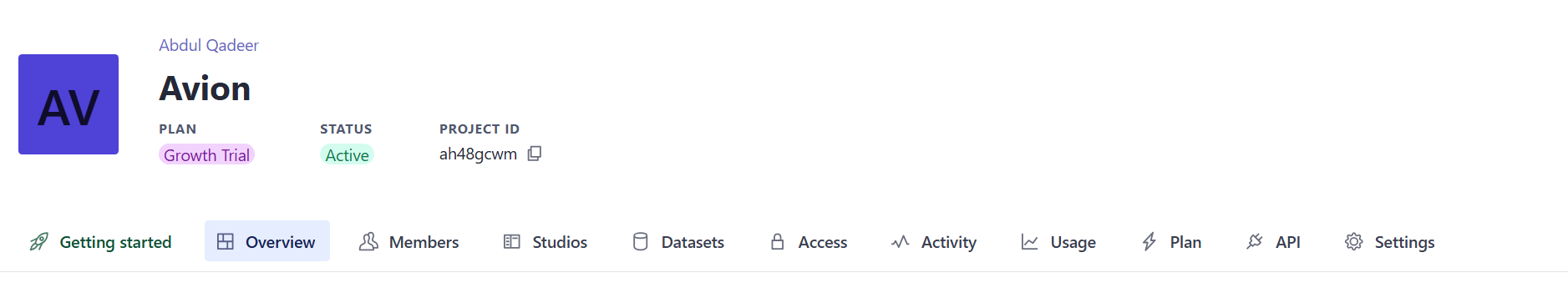
1. **Setting up Sanity environment :**

After installing Sanity in my project, I generated the **SANITY\_API\_TOKEN** from my Sanity Project dashboard through my browser. To generate that I did some steps.

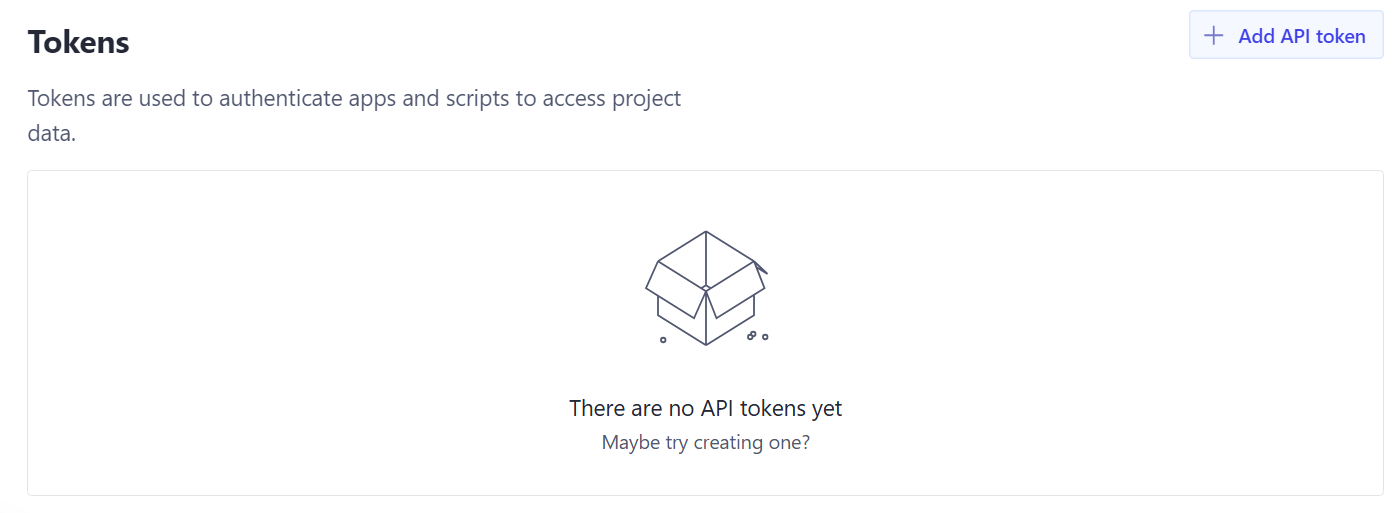
1. On my main dashboard I navigated to my project.



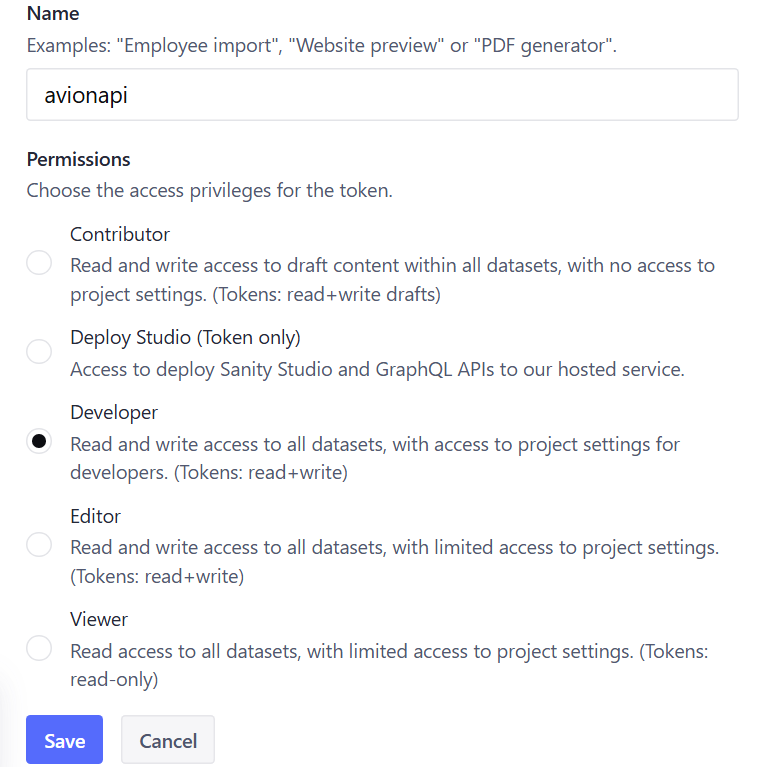
1. After opening my project dashboard I navigated to the **API** tab.



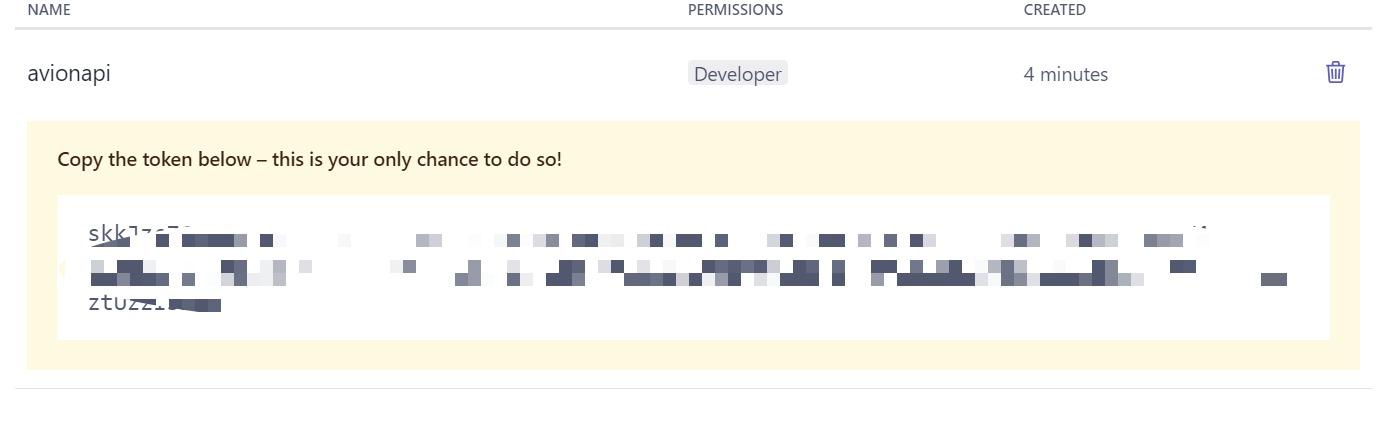
1. After navigating to the API tab I scrolled down a bit to find the **Token** section.



1. After I scrolled to the Token section I clicked **Add API token**, wrote the name my API should have and selected the **Developer** option and hitted the **Save** button.



1. After I hitted the **Save** button my API Token appeared. I copied it and put it in my .env.local file in my root directory. Below the **NEXT\_PUBLIC\_SANITY\_DATASET** line I wrote **SANITY\_API\_TOKEN = “”** inside my quotation marks I pasted my API Token I copied.



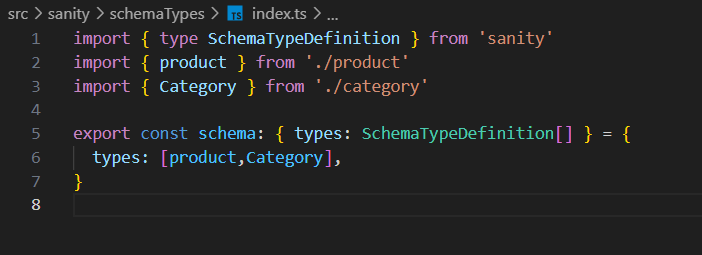


1. **Defining the Schemas :**

After setting up the Sanity environment completely for my project, inside the **app/sanity/schemaTypes** directory I created product.ts & category.ts files and in these files I defined all of my schemas.



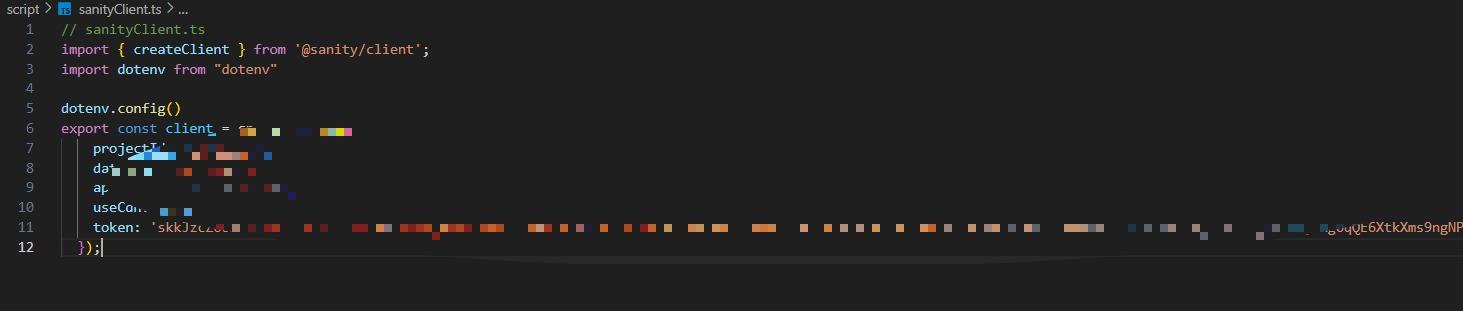
After that, I added my these schemaTypes into my index.ts located in **src/app/sanity/schemaTypes**

****

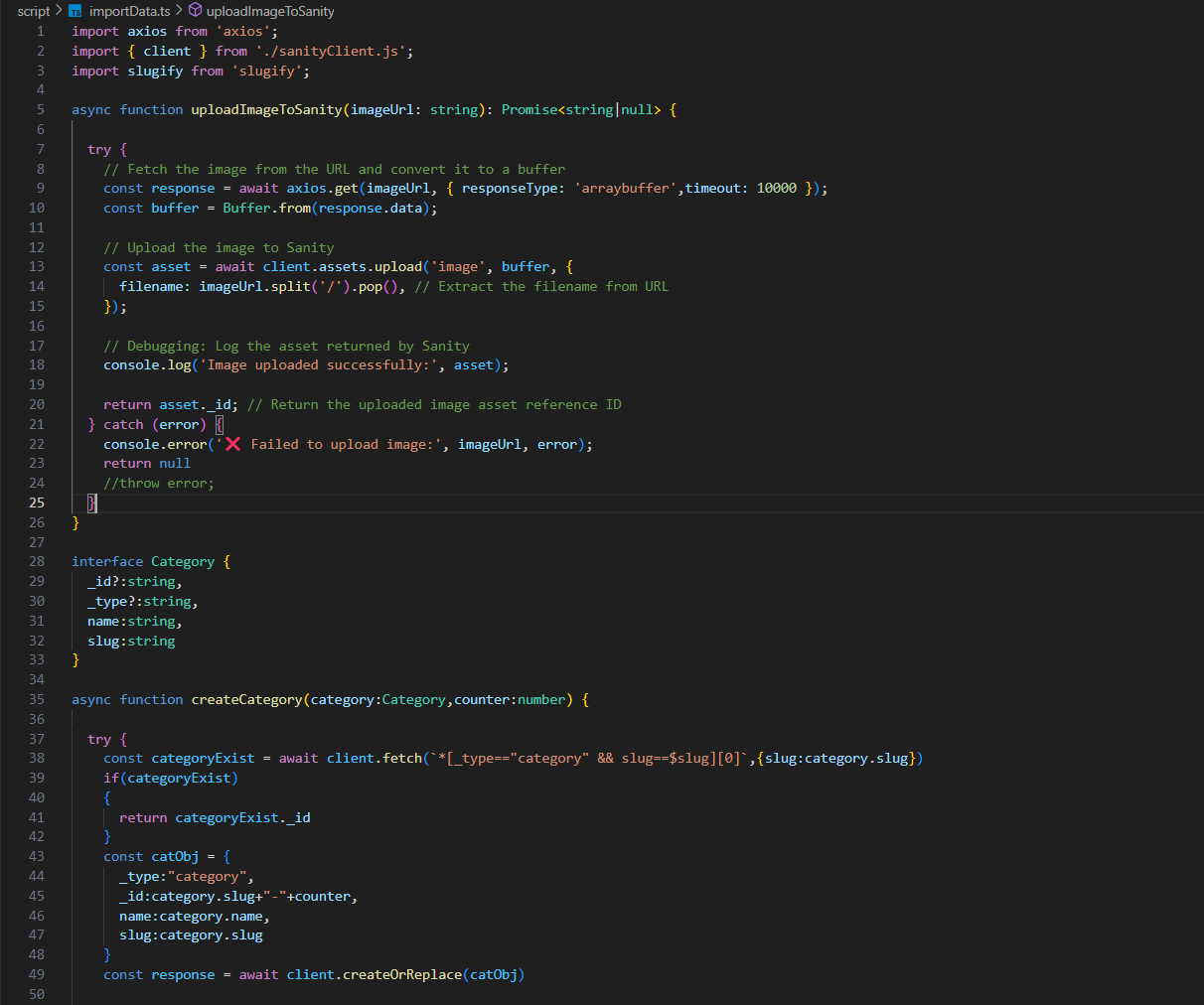
1. **Migrating the API data :**

After defining schemaTypes finally I started to migrate the API data. So, for that I did several steps.

1. I created a **script** folder in my root directory and in that folder I created **sanityClient.ts** to connect my frontend to my **Sanity CMS** backend. In that I defined all of my main info from which most of the info came from my **.env.local** file.

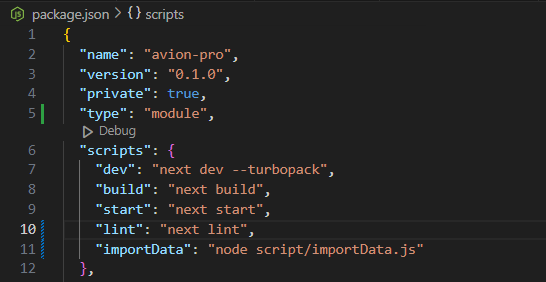


1. After connecting my frontend to Sanity CMS backend I created a file named **importData.ts** in the same directory **/scripts** where I imported my backend CMS data. To import the data properly you must have installed **Axios**,which is an API requesting tool.

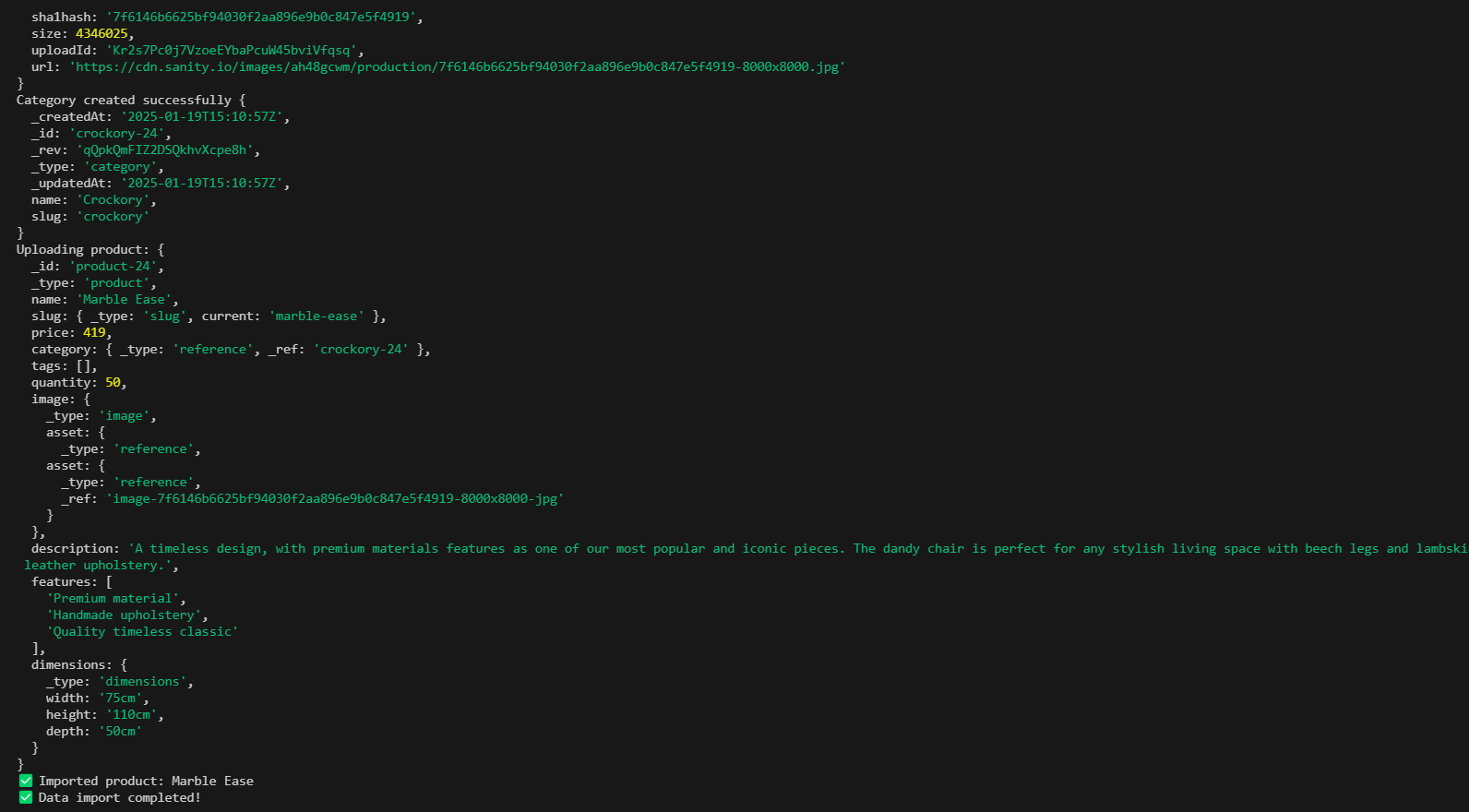


In this process of migrating the API data there is a command that you have to run that is **npm install @sanity/client axios dotenv** this command installs axios and dotenv for you sanity client.

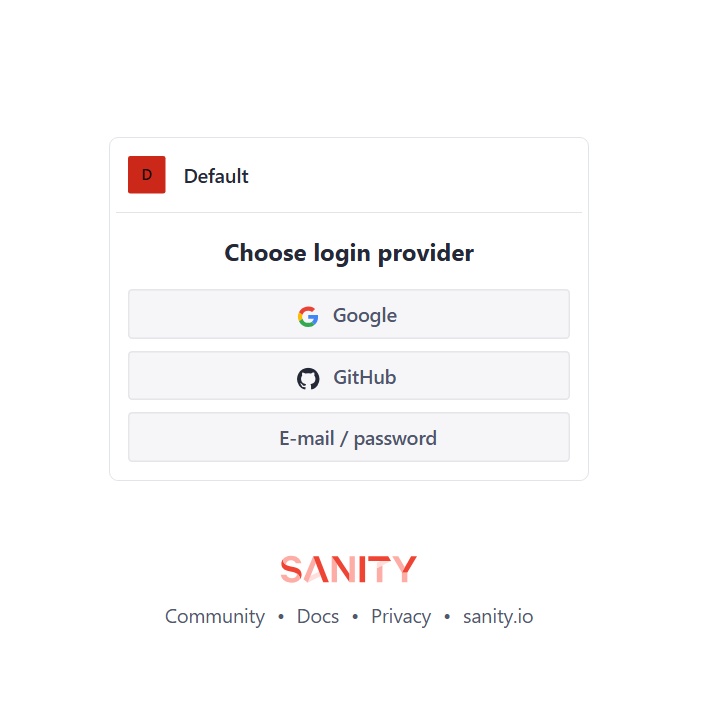
1. After doing these steps run the **npm install --save-dev typescript** command in your cmd to install typescript in your project then do **cd scripts** to move to the scripts directory then run **npx tsc importData.ts sanityClient.ts** to convert your typescript files into js files to run them then you have to write some lines in your **package.json** the one is below the 4th line of **package.json** that is **“type”: “module”** and the second one should be in the last of your scripts area where you can run your scripts that is **“importData”: “node script/importData.js”**



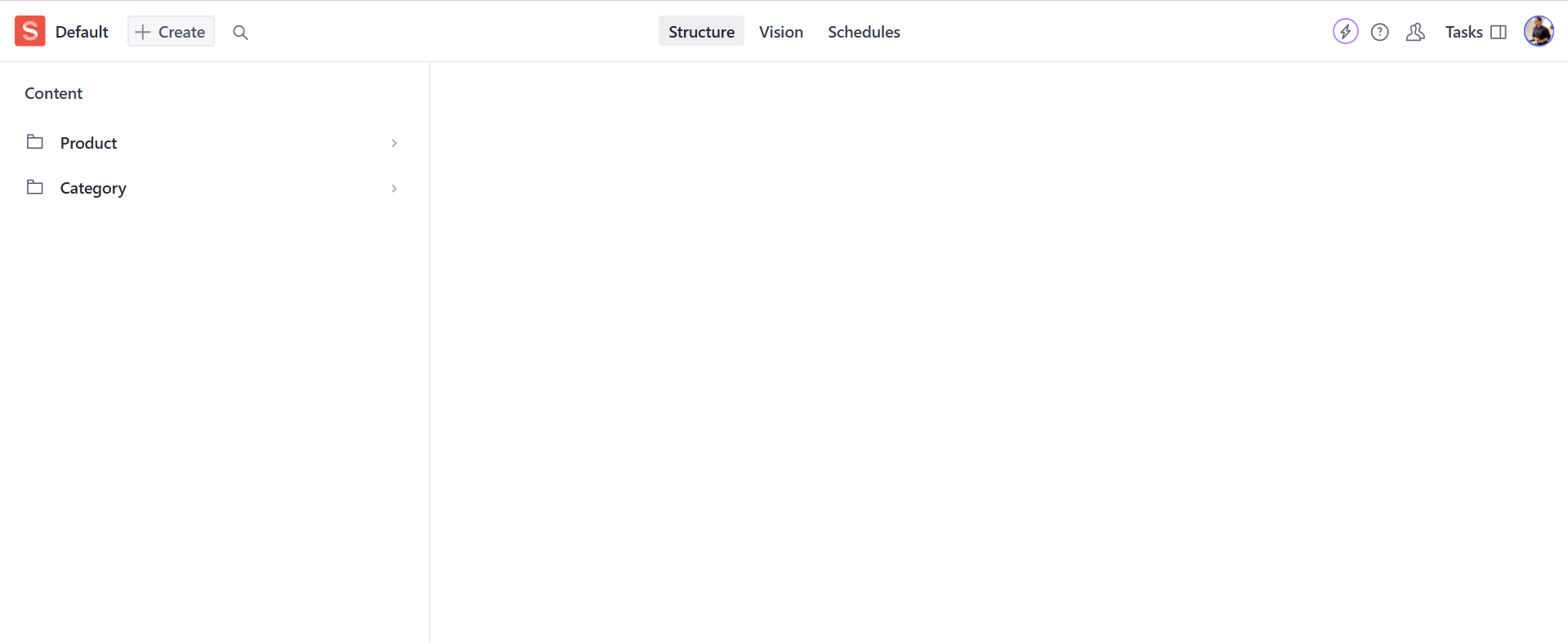
1. After doing these steps run the command **npm run importData** to import all of the data you migrated and **CONGRATS** your data is migrated successfully.



Now, you can check the data you migrated by running your website at localhost:3000 and don’t forget to put the /studio at the end as it is important to open your sanity studio. If you are opening your Sanity studio first time on your project it will look something like this :

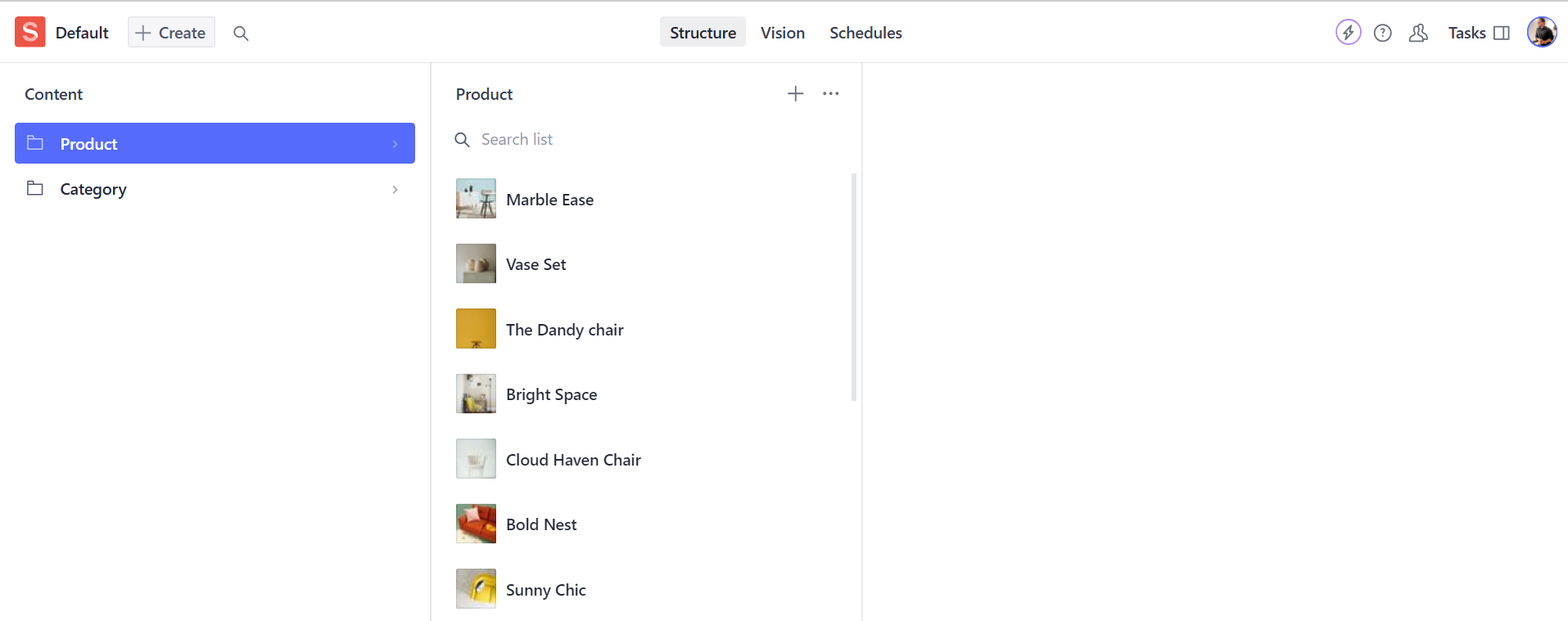


If you are using any of these methods in signing in your Sanity dashboard choose and login with that, after that you will see this type of interface :

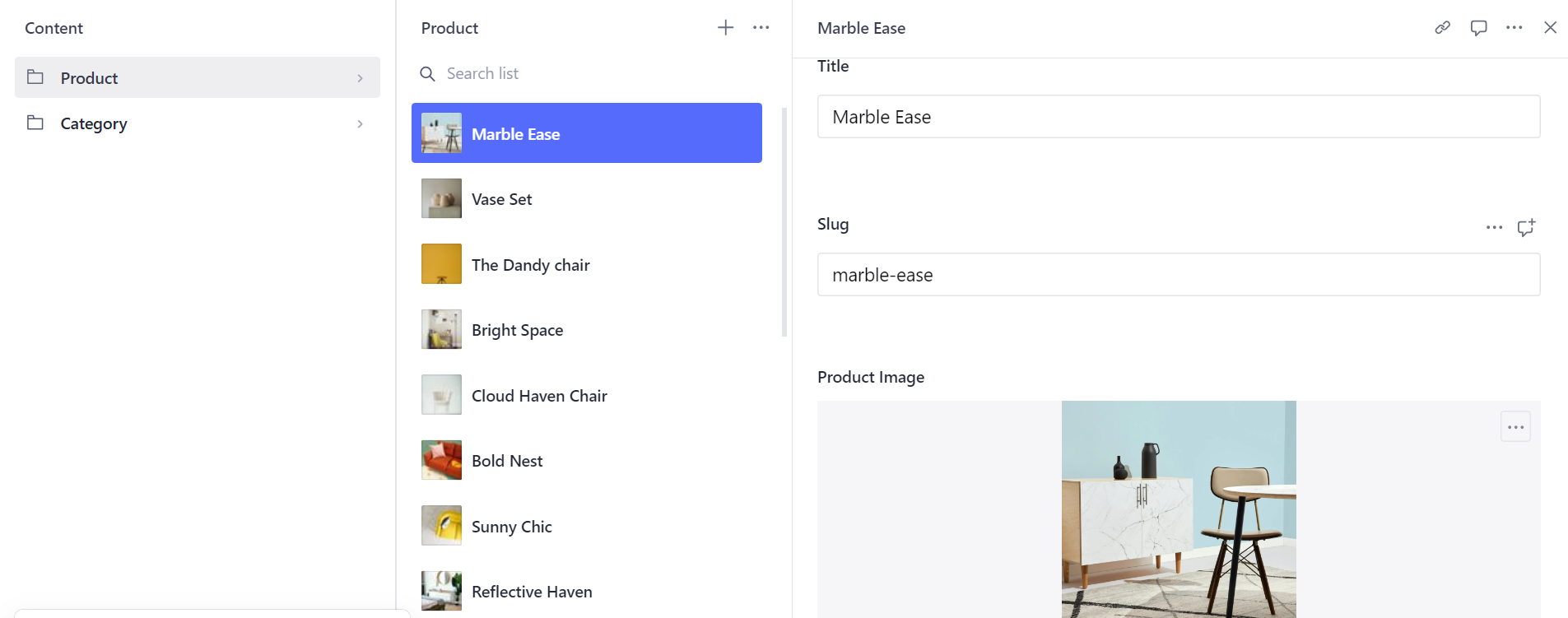


In my case, as I defined two types of schemas so the two folders appearing for your case it will be different or maybe same.

When you open one of these folders you see a list of your API items that you migrated.



When you click one of these you see that your Sanity fields are populated with your defined schema types and you API data you migrated.



1. **Displaying the data on frontend :**

After my API integration and data migration work, now I move to the most critical part which is displaying the elements of the CMS to my frontend. To do that, first of all I opened my file where I have to display my CMS elements to my frontend. Then I imported the sanityClient and also defined my all sanityClient elements from my .env.local to apply the Sanity environment in my that file.

import sanityClient from "@sanity/client";

const sanity = sanityClient({

projectId: "MY PROJECT ID",

dataset: "MY DATASET",

apiVersion: "MY API VERSION",

useCdn: true,

});

After that I defined all of my types of elements in my API.

interface Category {

name: string;

slug: string;

}

interface Dimensions {

width: string;

height: string;

depth: string;

}

interface Products {

name: string;

description: string;

image: string;

\_id: string;

features: string[];

dimensions: Dimensions;

category: Category;

price: number;

tags: string[];

}

After that I created a async function to fetch my all elements present in my Sanity CMS.

async function fetchProducts() {

const products = await sanity.fetch(`

\*[\_type == "product"]{

\_id,

name,

"image": image.asset->url,

price,

description,

features,

dimensions,

category,

tags

}

`);

return products;

}

After that I converted my main const of my file to async function and then I created the products variable, gave that variable a type and then fetched the whole elements present in my Sanity CMS.

export default async function ProductListing() {

const products: Products[] = await fetchProducts();

And after that I removed my hard coded methods from my file parts and replaced to my dynamic writed code and repeated the same process for every place and in every file where implementing API is necessary.

**BEFORE :**

<div className="mt-4 flex justify-between">

<div>

<h3 className="text-sm font-normal text-gray-900">

Lucy lamp

</h3>

<p className="mt-1 text-sm text-gray-500">€200</p>

</div>

</div>

**AFTER :**

<div className="mt-4 flex justify-between">

<div>

<h3 className="text-sm font-normal text-gray-900">

{product.name}

</h3>

<p className="mt-1 text-sm text-gray-500">€{product.price}</p>

</div>

</div>

After that my API is also showing on my frontend.

