

## Hackathon-3

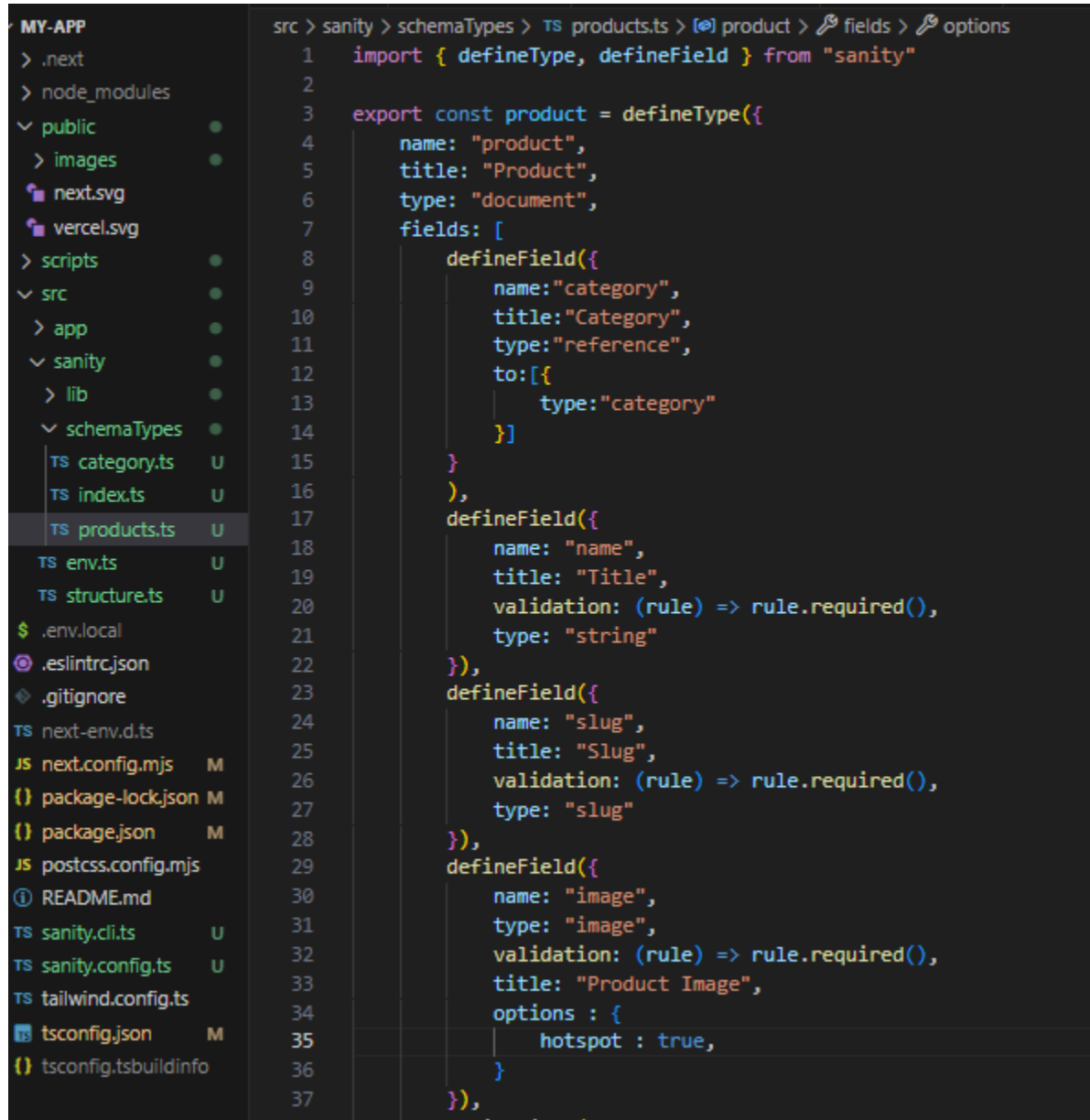
Day-3

### Folder Layouts in Project

▼ MY-APP	src > app
> .next	1
> node_modules	2
▼ public ●	3
> images ●	4
📁 next.svg	5
📁 vercel.svg	6
> scripts ●	7
▼ src ●	8
> app ●	9
> sanity ●	10
\$.env.local	11
🔗 .eslintrc.json	12
🔗 .gitignore	13
TS next-env.d.ts	14
JS next.config.mjs M	15
{ } package-lock.json M	16
{ } package.json M	17
JS postcss.config.mjs	18
📄 README.md	19
TS sanity.cli.ts U	20
TS sanity.config.ts U	21
TS tailwind.config.ts	22
TS tsconfig.json M	23
{ } tsconfig.tsbuildinfo	24
	25
	26
	27
	28
	29

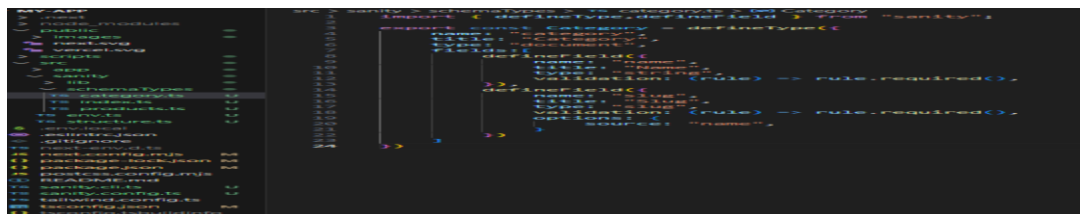
Successfully setup Sanity in Project :

## Sanity Product Schemas Defined:



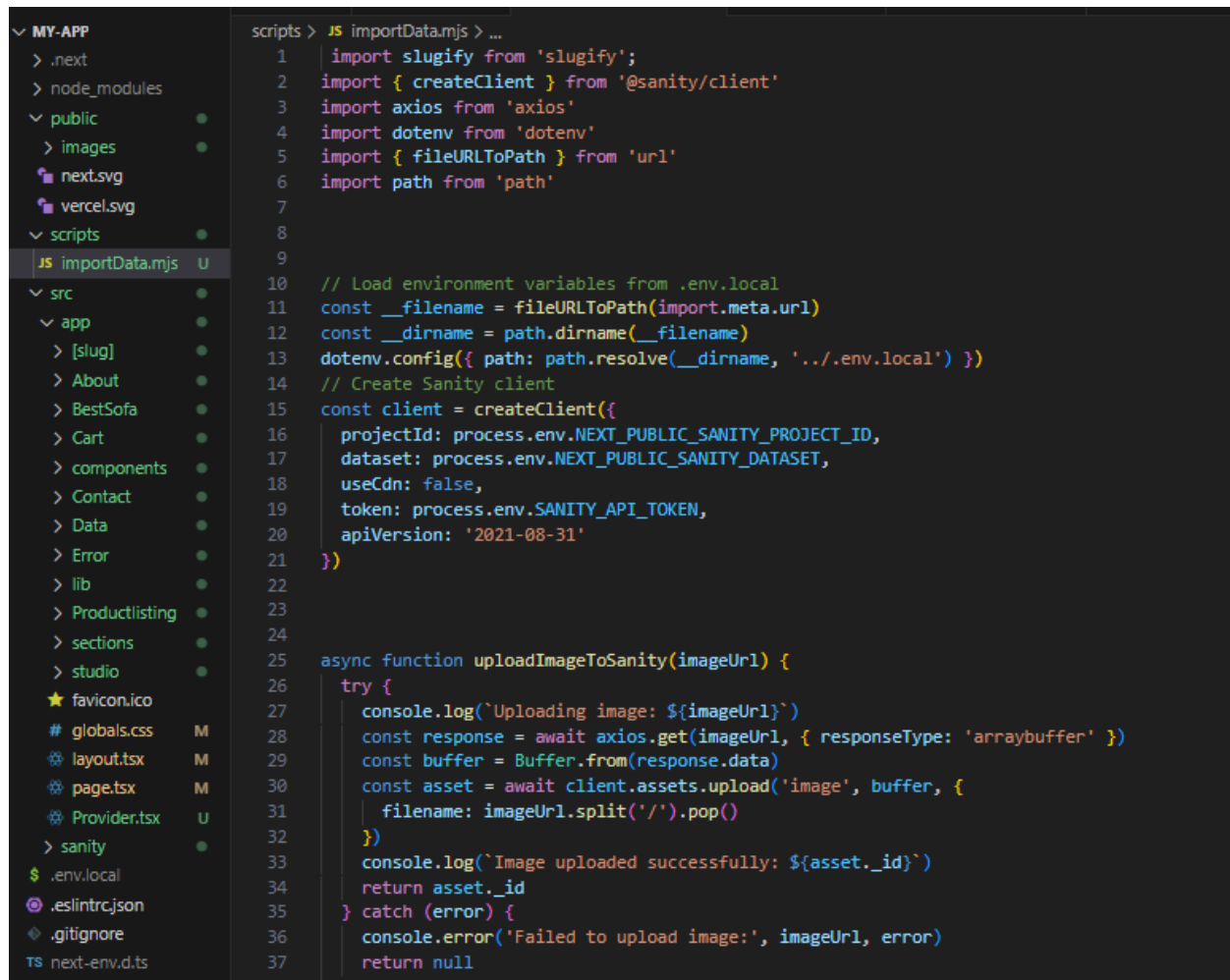
```
src > sanity > schemaTypes > TS products.ts > fields > options
1  import { defineType, defineField } from "sanity"
2
3  export const product = defineType({
4    name: "product",
5    title: "Product",
6    type: "document",
7    fields: [
8      defineField({
9        name: "category",
10       title: "Category",
11       type: "reference",
12       to: [{
13         type: "category"
14       }]
15     }
16   ),
17   defineField({
18     name: "name",
19     title: "Title",
20     validation: (rule) => rule.required(),
21     type: "string"
22   }),
23   defineField({
24     name: "slug",
25     title: "Slug",
26     validation: (rule) => rule.required(),
27     type: "slug"
28   }),
29   defineField({
30     name: "image",
31     type: "image",
32     validation: (rule) => rule.required(),
33     title: "Product Image",
34     options : {
35       hotspot : true,
36     }
37   }),
38 ]
39 })
```

## Category Schema: optional



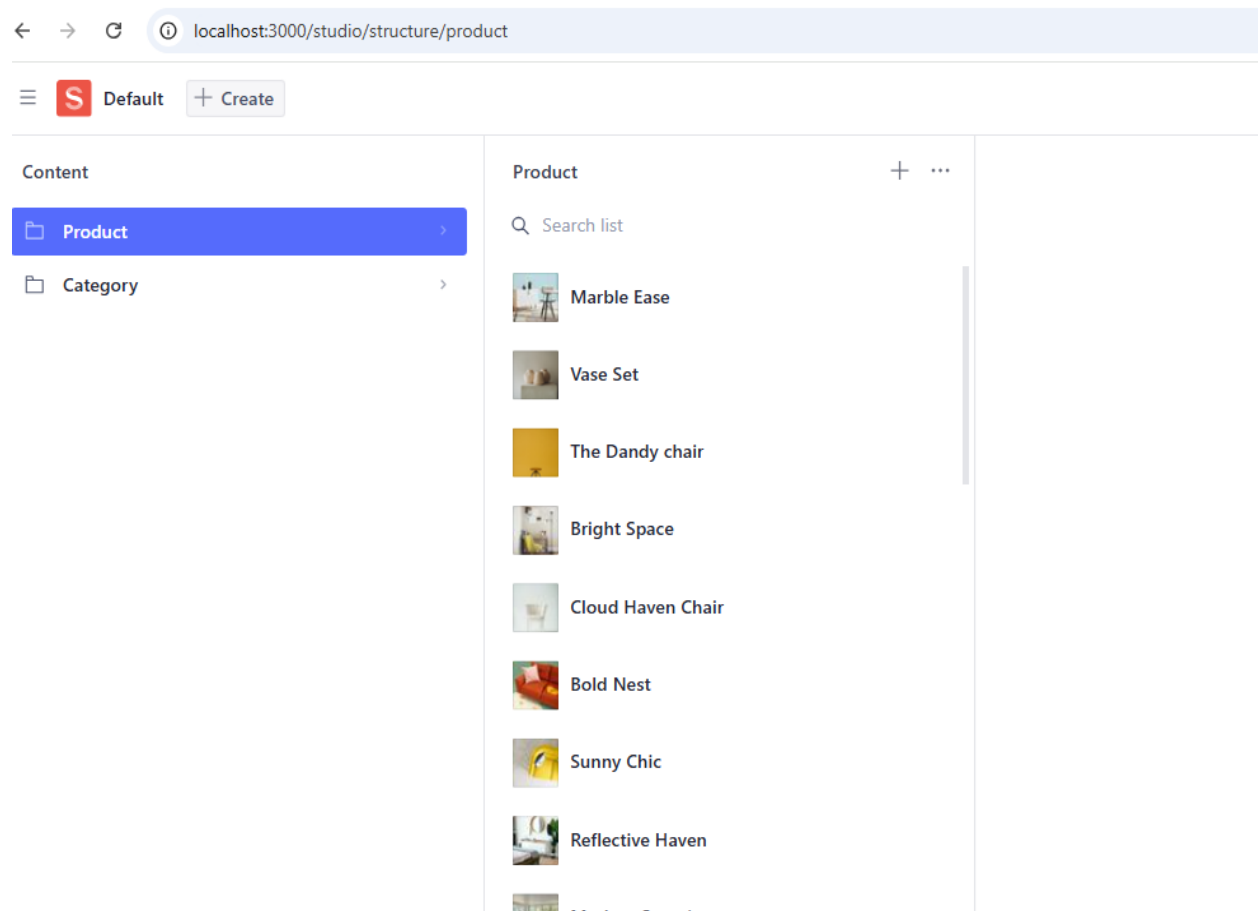
## Migration data to sanity :

I created scripts folder then I create a file of .mjs then I done succesed migration:



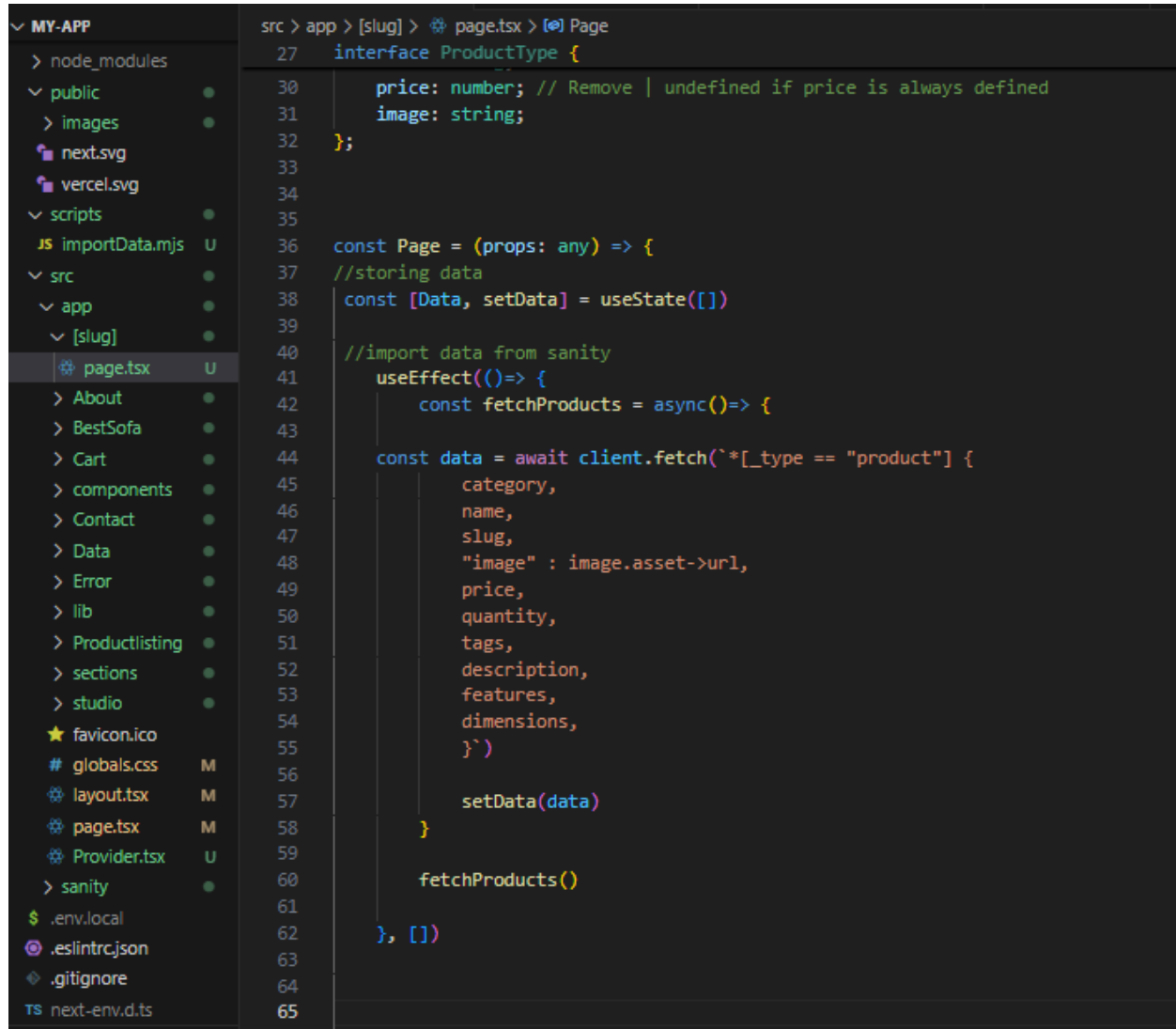
```
scripts > JS importData.mjs > ...
1  import slugify from 'slugify';
2  import { createClient } from '@sanity/client'
3  import axios from 'axios'
4  import dotenv from 'dotenv'
5  import { fileURLToPath } from 'url'
6  import path from 'path'
7
8
9
10 // Load environment variables from .env.local
11 const __filename = fileURLToPath(import.meta.url)
12 const __dirname = path.dirname(__filename)
13 dotenv.config({ path: path.resolve(__dirname, '../.env.local') })
14 // Create Sanity client
15 const client = createClient({
16   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
17   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
18   useCdn: false,
19   token: process.env.SANITY_API_TOKEN,
20   apiVersion: '2021-08-31'
21 })
22
23
24
25 async function uploadImageToSanity(imageUrl) {
26   try {
27     console.log(`Uploading image: ${imageUrl}`)
28     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' })
29     const buffer = Buffer.from(response.data)
30     const asset = await client.assets.upload('image', buffer, {
31       filename: imageUrl.split('/').pop()
32     })
33     console.log(`Image uploaded successfully: ${asset._id}`)
34     return asset._id
35   } catch (error) {
36     console.error('Failed to upload image:', imageUrl, error)
37     return null
38   }
39 }
```

## Successfully Imported Data in Sanity Dashboard Studio



*Imported Data in sanity Dashboard from API through the Next JS .*

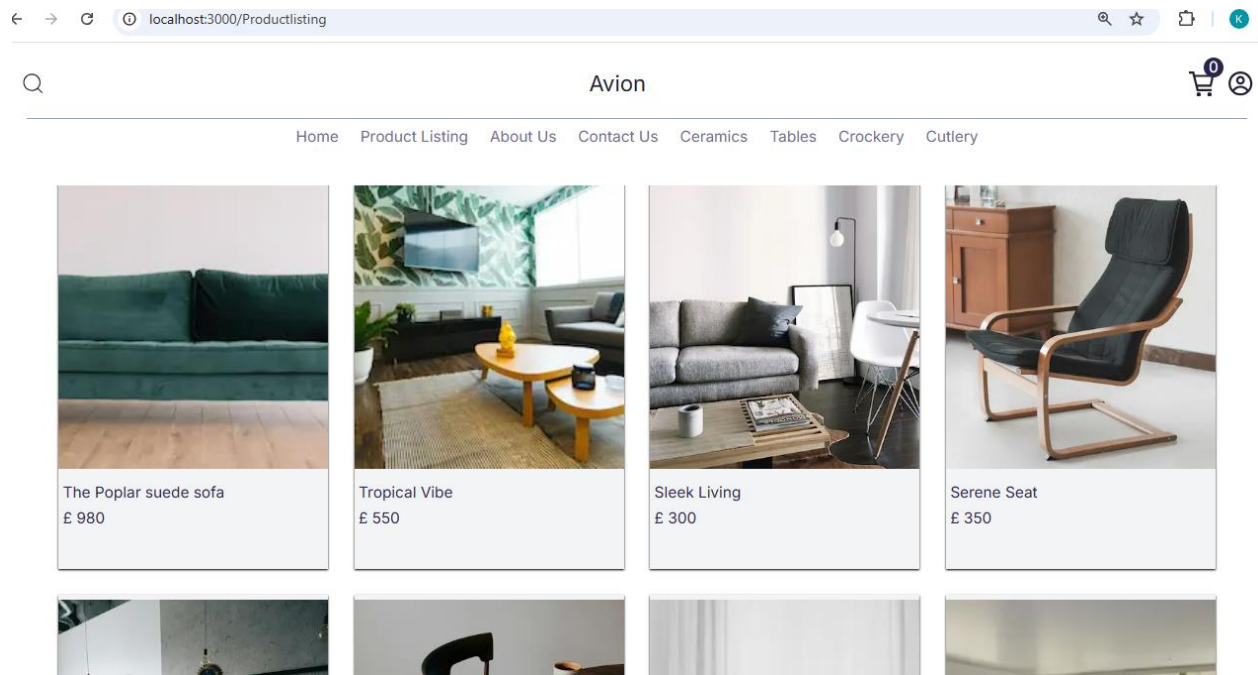
## Fetching Data from Sanity into NextJS Project .



```
src > app > [slug] > page.tsx > Page
27  interface ProductType {
30      price: number; // Remove | undefined if price is always defined
31      image: string;
32  };
33
34
35
36  const Page = (props: any) => {
37      //storing data
38      const [Data, setData] = useState([])
39
40      //import data from sanity
41      useEffect(()=> {
42          const fetchProducts = async()=> {
43
44              const data = await client.fetch(`*[_type == "product"] {
45                  category,
46                  name,
47                  slug,
48                  "image" : image.asset->url,
49                  price,
50                  quantity,
51                  tags,
52                  description,
53                  features,
54                  dimensions,
55              }`)
56
57              setData(data)
58          }
59
60          fetchProducts()
61
62      }, [])
63
64
65
```

***Successfully Fetched data through the Grooq query from sanity CMS into NextJS projects..***

## Data Successfully Displayed In frontend



**Succesfully diplayed data in next js project from sanity cms:**