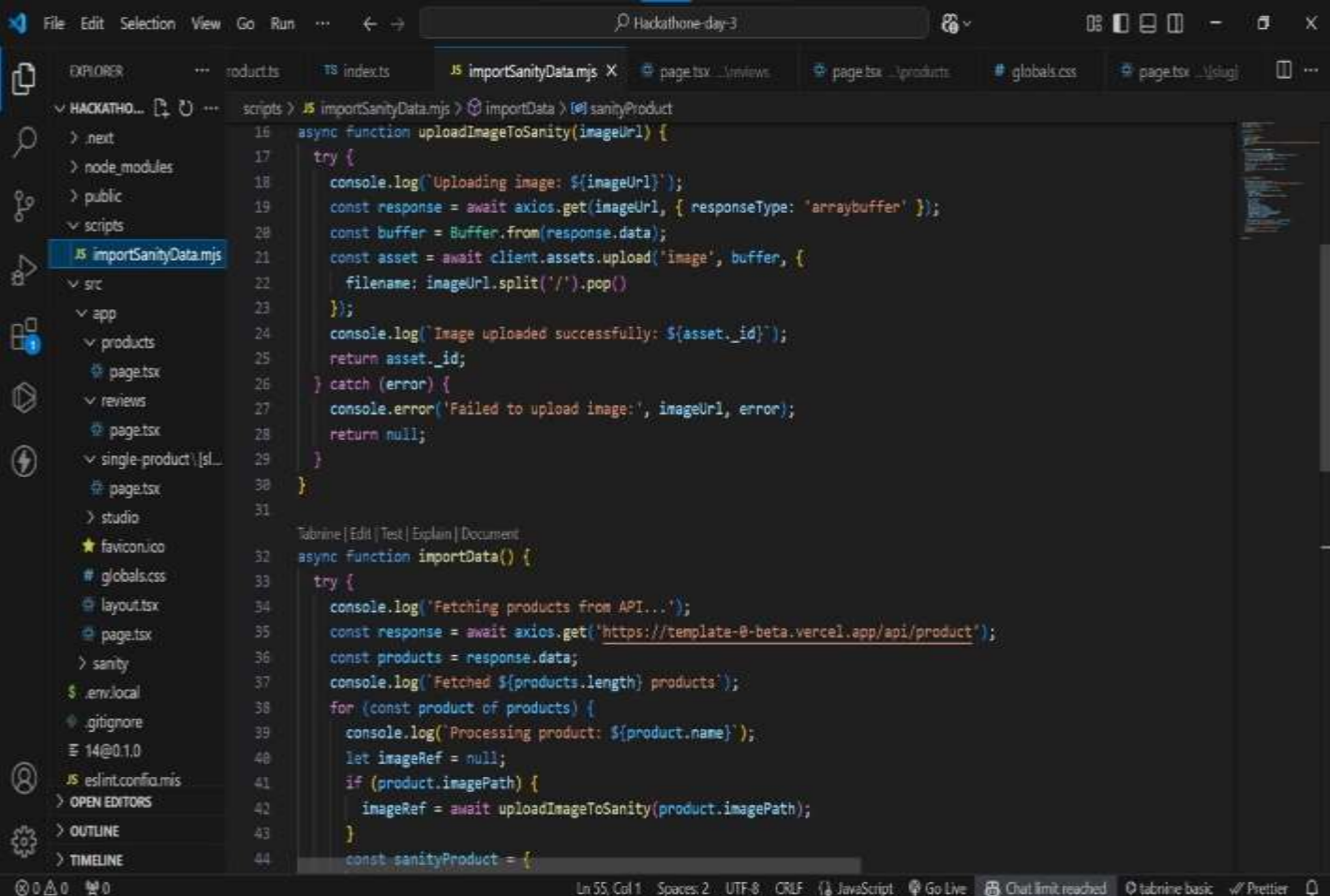


HACKATHON DAY THREE TASK

API INTEGRATION & DATA
MIGRATION

UNDERSTANDING THE API

- 1) Importing the API data
- 2) make a file of importData.mjs
- 3) Adjust schema in
sanity/schemaTypes
sanity/schemaTypes/product.ts



```
File Edit Selection View Go Run ... Hackathone-day-3
EXPLORER scripts > JS importSanityData.mjs X page.tsx ...views page.tsx ...products # globals.css page.tsx ...slug
HACKATHON > .next > node_modules > public > scripts JS importSanityData.mjs > src > app > products > page.tsx > reviews > page.tsx > single-product > |sl... > page.tsx > studio > favicon.ico # globals.css > layout.tsx > page.tsx > sanity > .env.local > .gitignore > 14@0.1.0 > JS eslint.config.mjs > OPEN EDITORS > OUTLINE > TIMELINE
Ln 55, Col 1 Spaces: 2 UTF-8 CRLF JavaScript Go Live Chat limit reached tabnine basic Prettier

16 async function uploadImageToSanity(imageUrl) {
17   try {
18     console.log('Uploading image: ${imageUrl}');
19     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
20     const buffer = Buffer.from(response.data);
21     const asset = await client.assets.upload('image', buffer, {
22       filename: imageUrl.split('/').pop()
23     });
24     console.log('Image uploaded successfully: ${asset._id}');
25     return asset._id;
26   } catch (error) {
27     console.error('Failed to upload image:', imageUrl, error);
28     return null;
29   }
30 }
31
32 async function importData() {
33   try {
34     console.log('Fetching products from API...');
35     const response = await axios.get('https://template-0-beta.vercel.app/api/product');
36     const products = response.data;
37     console.log('Fetched ${products.length} products');
38     for (const product of products) {
39       console.log('Processing product: ${product.name}');
40       let imageRef = null;
41       if (product.imagePath) {
42         imageRef = await uploadImageToSanity(product.imagePath);
43       }
44       const sanityProduct = {
```

ADDING SCHEMA

1) make schema in

Sanity/schemaTypes/Product.ts

2)import in index.ts

3)Add query in the api route

```
export const allProductsQuery =
  `*[_type=="products"]{
    _id,
    name,
    description,
    price,
    "imageUrl" : image.asset->url,
    category,
    discountPercent,
    "isNew": new,
    colors,
    sizes
  }`
```

```
TS products.ts X
src > sanity > schemaTypes > TS products.ts > default > fields
1  import { defineType } from "sanity"
2
3  export default defineType({
4    name: 'products',
5    title: 'Products',
6    type: 'document',
7    fields: [
8      {
9        name: 'name',
10       title: 'Name',
11       type: 'string',
12     },
13     {
14       name: 'price',
15       title: 'Price',
16       type: 'number',
17     },
18     {
19       name: 'description',
20       title: 'Description',
21       type: 'text',
22     },
23     {
24       name: 'image',
25       title: 'Image',
26       type: 'image',
27     },
28   ]
29 }
```

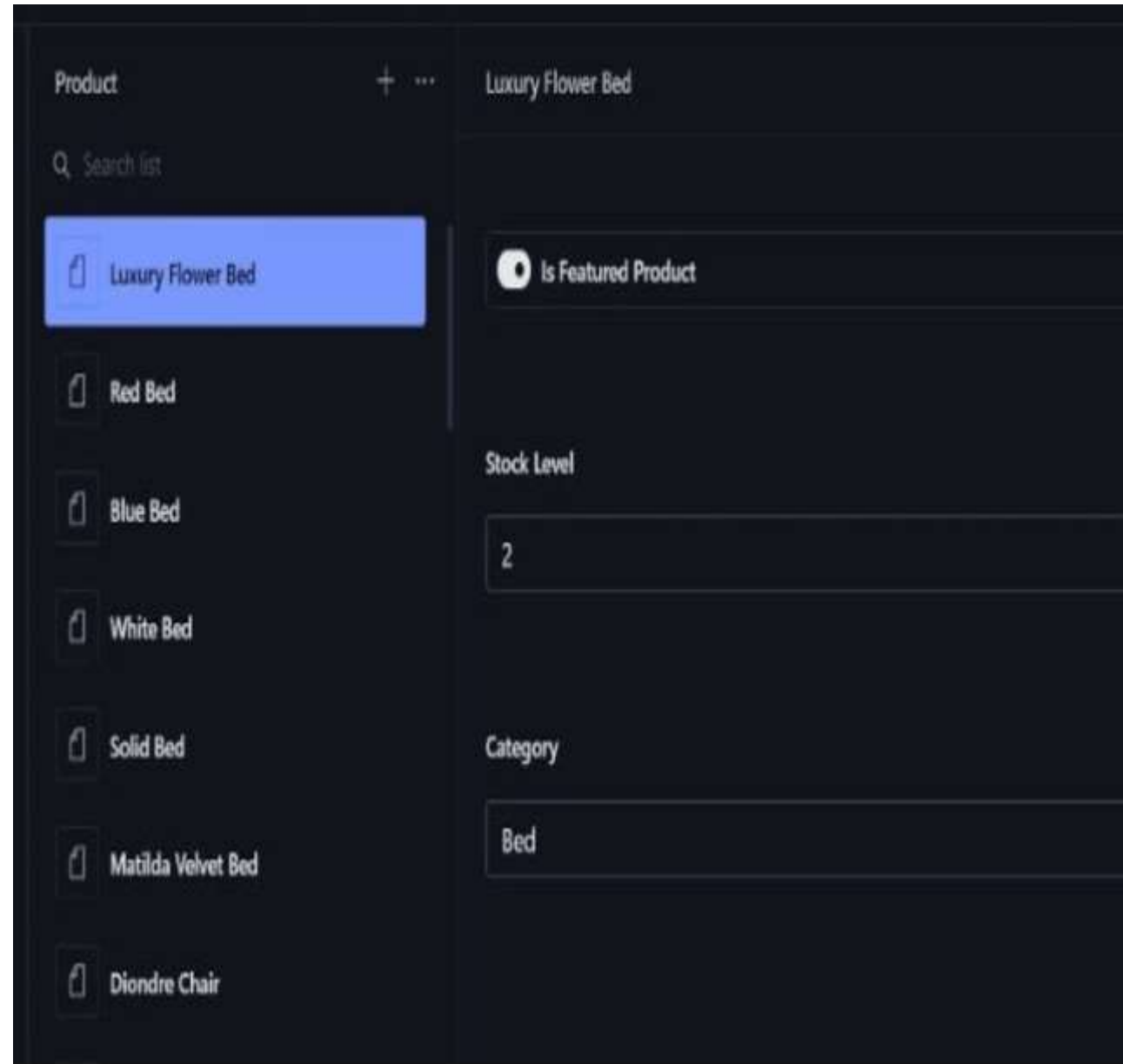
DATA IN SANITY

1)now the DATA is stored in
SANITY

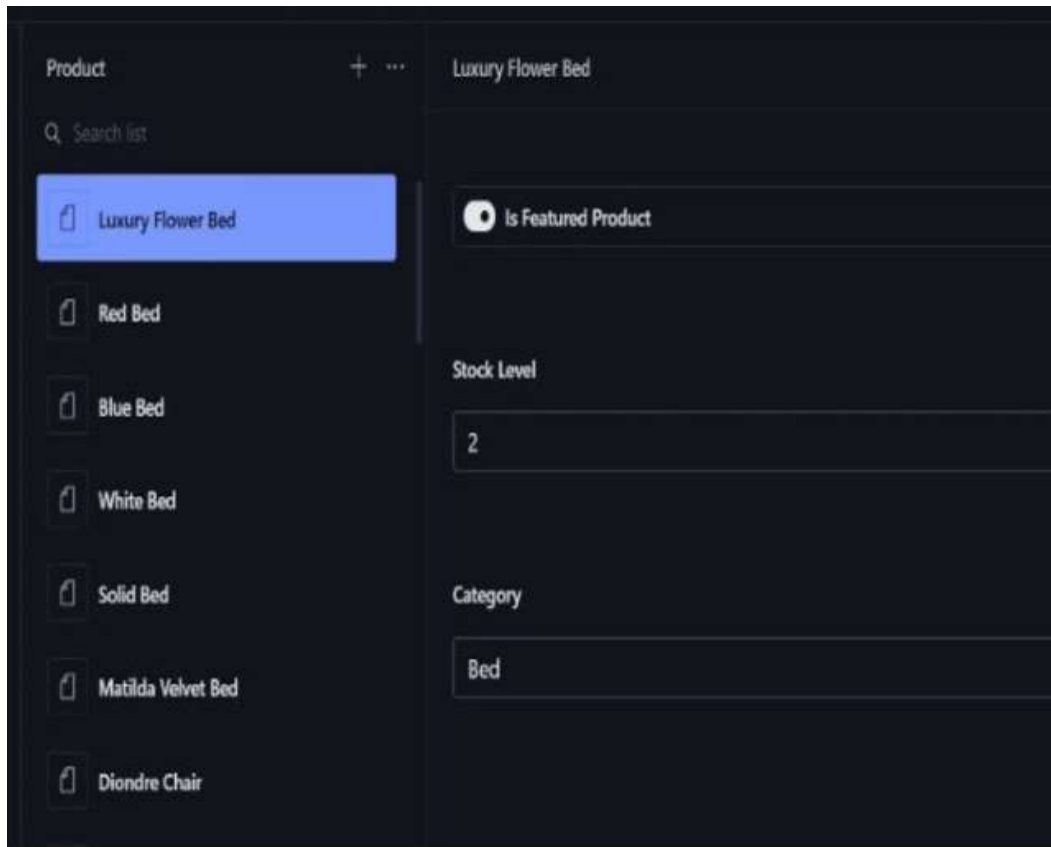
2)make tokens, id through sanity
Dashboard and add in .env.local

3)After running node importData.mjs
The DATA is importing to sanity you can
see In terminal.

4)The data is successfully is stored in
SANITY



SANITY OUTPUT



FRONTEND OUTPUT

