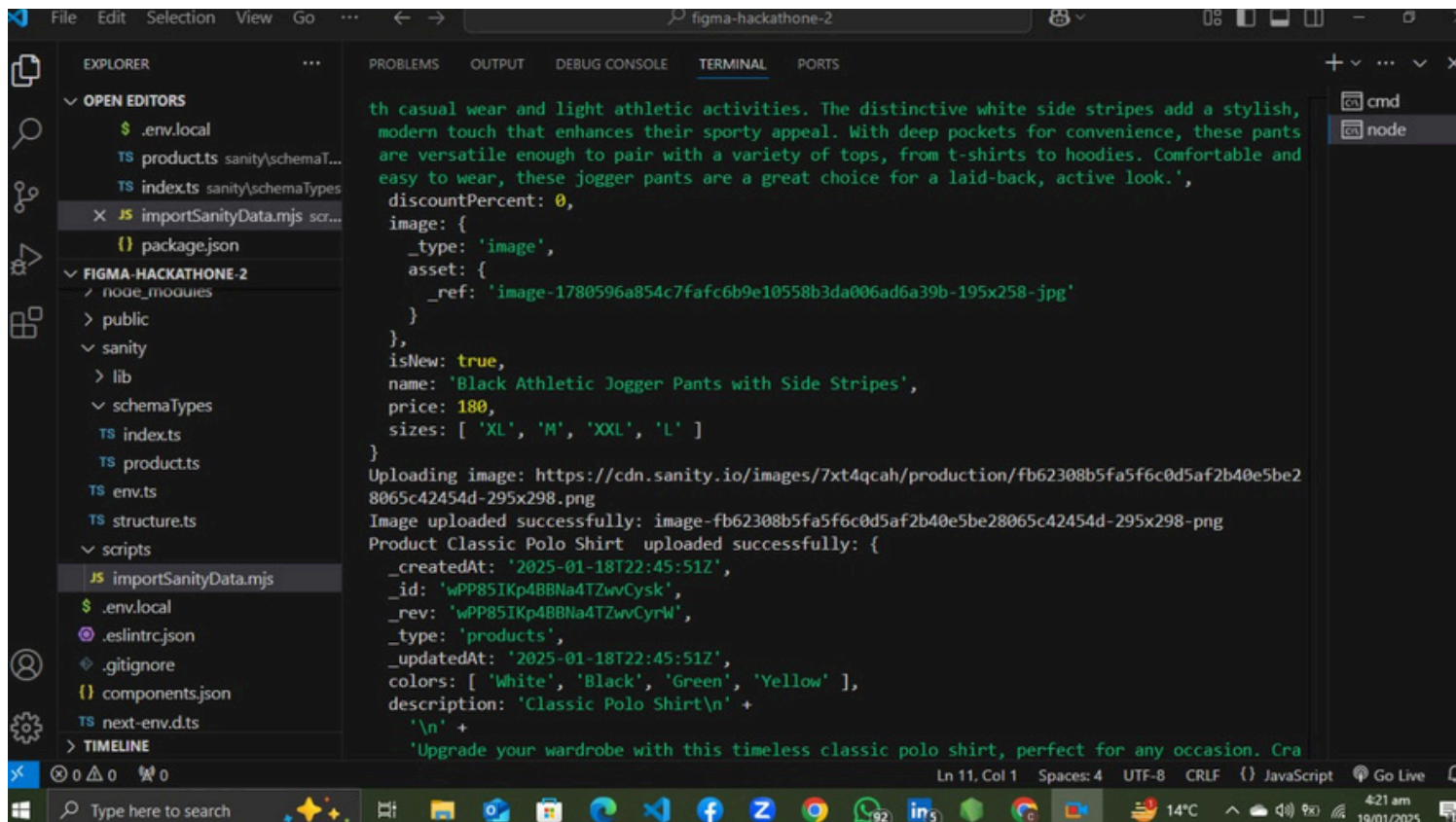


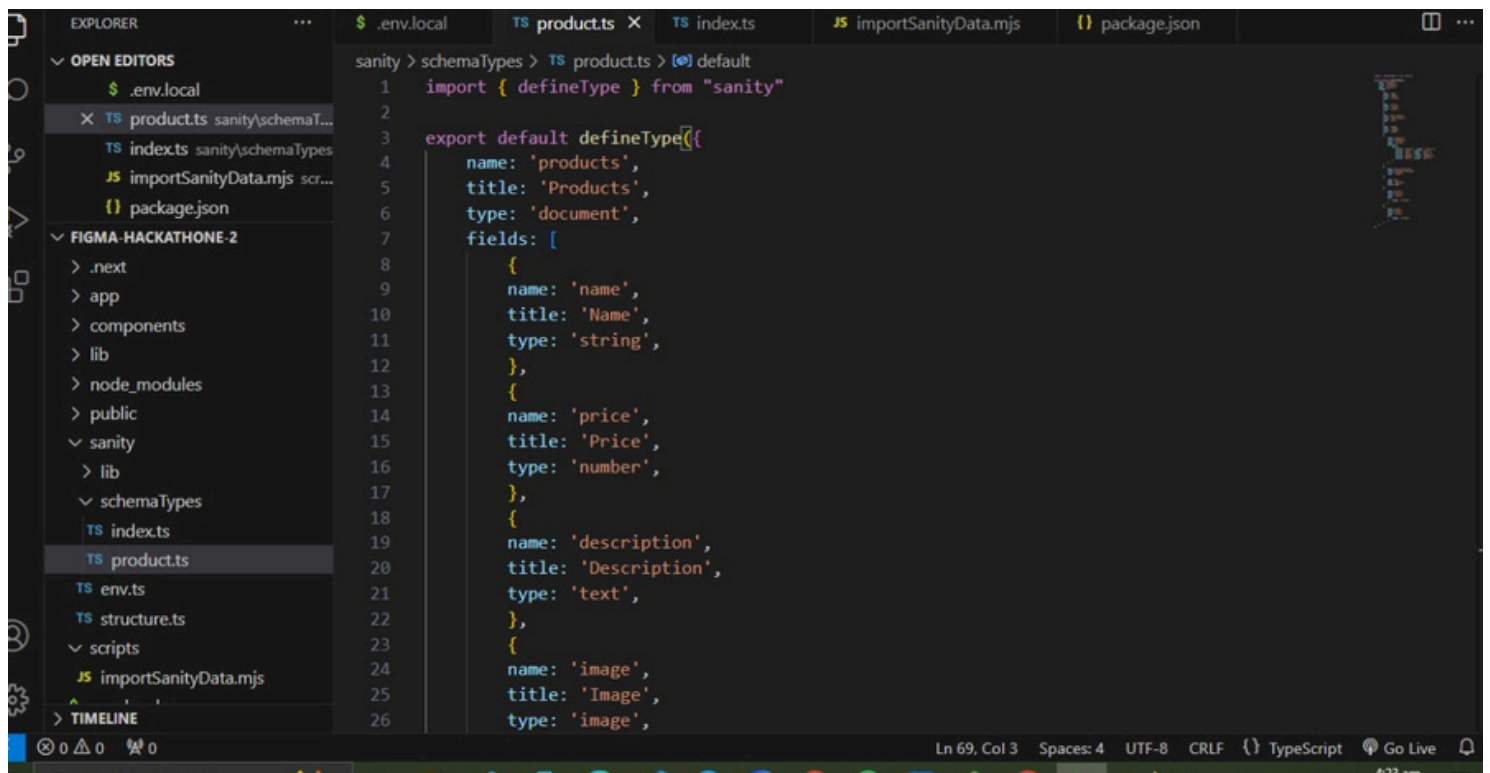
Day-3

API integration and Migration



The screenshot shows a VS Code editor with a terminal window open. The terminal displays the following output:

```
th casual wear and light athletic activities. The distinctive white side stripes add a stylish, modern touch that enhances their sporty appeal. With deep pockets for convenience, these pants are versatile enough to pair with a variety of tops, from t-shirts to hoodies. Comfortable and easy to wear, these jogger pants are a great choice for a laid-back, active look.',
discountPercent: 0,
image: {
  _type: 'image',
  asset: {
    _ref: 'image-1780596a854c7fafc6b9e10558b3da006ad6a39b-195x258-jpg'
  }
},
isNew: true,
name: 'Black Athletic Jogger Pants with Side Stripes',
price: 180,
sizes: [ 'XL', 'M', 'XXL', 'L' ]
}
Uploading image: https://cdn.sanity.io/images/7xt4qcah/production/fb62308b5fa5f6c0d5af2b40e5be28065c42454d-295x298.png
Image uploaded successfully: image-fb62308b5fa5f6c0d5af2b40e5be28065c42454d-295x298-png
Product Classic Polo Shirt uploaded successfully: {
  _createdAt: '2025-01-18T22:45:51Z',
  _id: 'wPP85IKp48BNa4TZwvCysk',
  _rev: 'wPP85IKp48BNa4TZwvCyrW',
  _type: 'products',
  _updatedAt: '2025-01-18T22:45:51Z',
  colors: [ 'White', 'Black', 'Green', 'Yellow' ],
  description: 'Classic Polo Shirt\n' +
    '\n' +
    'Upgrade your wardrobe with this timeless classic polo shirt, perfect for any occasion. Cra
```



The screenshot shows a VS Code editor with a TypeScript file open. The file defines a product schema using the Sanity schema type system. The code is as follows:

```
sanity > schemaTypes > TS products > [0] default
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',
```

EXPLORER

OPEN EDITORS

\$.env.local

TS product.ts sanity/schemaT...

TS index.ts sanity/schemaTypes

JS importSanityData.mjs scr...

package.json

FIGMA-HACKATHONE-2

.next

app

components

lib

node_modules

public

sanity

lib

schemaTypes

TS index.ts

TS product.ts

env.ts

structure.ts

scripts

importSanityData.mjs

TIMELINE

scripts > JS importSanityData.mjs > ...

1import { createClient } from '@sanity/client';

2import fetch from 'node-fetch';

3

4const client = createClient({

5 projectId: "mpwxc15w",

6 dataset: "production",

7 useCdn: true,

8 apiVersion: '2025-01-13',

9 token: "skTOL8h0STsqpDiiQL3DoHvp1rM89RiQYwJpqAR11T9z2m83F1Zkcvj2SudX8h9QV1RuWRv3qeYpncp9sE

10});

11

12async function uploadImageToSanity(imageUrl) {

13 try {

14 console.log(`Uploading image: \${imageUrl}`);

15

16 const response = await fetch(imageUrl);

17 if (!response.ok) {

18 throw new Error(`Failed to fetch image: \${imageUrl}`);

19 }

20

21 const buffer = await response.arrayBuffer();

22 const bufferImage = Buffer.from(buffer);

23

24 const asset = await client.assets.upload('image', bufferImage, {

25 filename: imageUrl.split('/').pop(),

26 });

