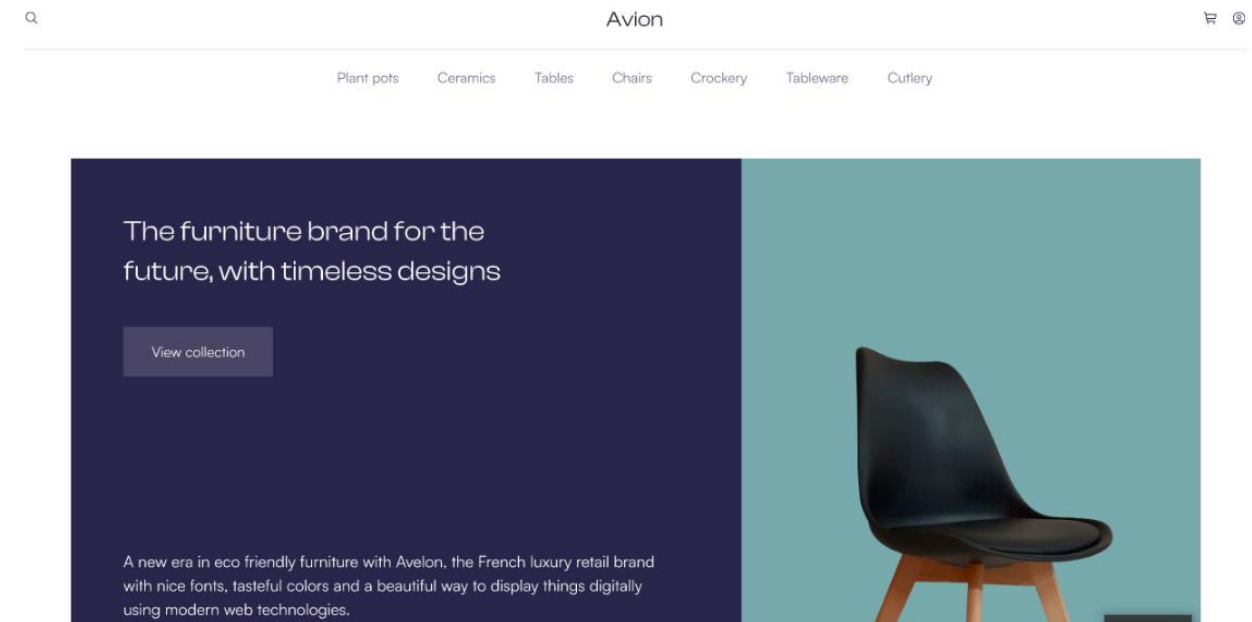# DOCUMENT TITLE: "DAY 3 - API INTEGRATION REPORT"

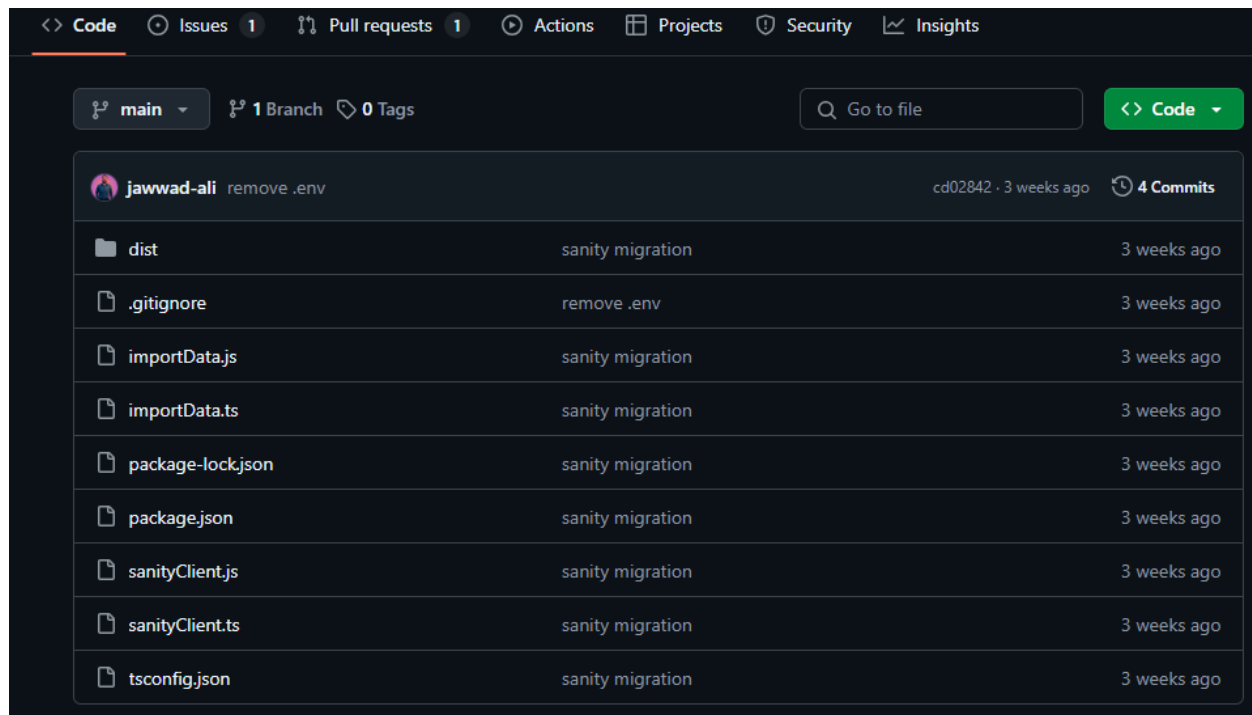## THE FURNITURE BRAND

SUBMIT:

✓ **A REPORT DOCUMENTING:**

Use Figma Template 02 (as last digit of roll number)
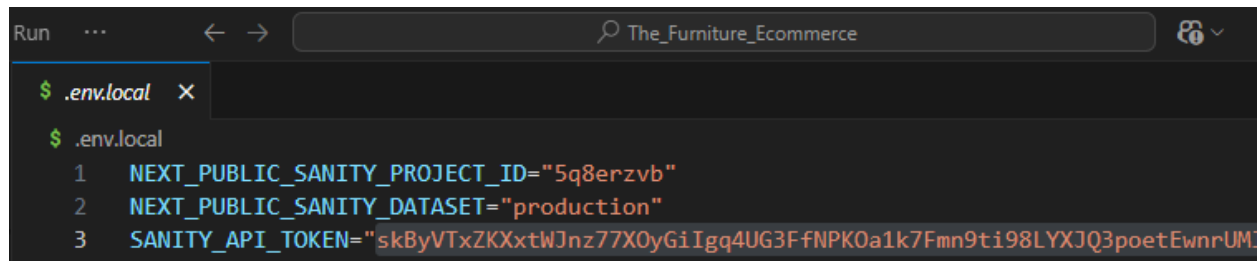

The Repository and Setting Up Environment Variables

1. We'll be using data from an external API—**FakestoreAPI**. You can import this API's data into **Sanity CMS**. Leveraging this feature of Sanity will significantly streamline your workflow.

2. Clone the project from: https://github.com/jawwad-ali/sanity-migration

4. Copy the URL:

5. Navigate to the sanity-migration folder and run the command **npm install.**

6. Create a .env file in your project and include the three variables shown in the attached picture.


```
$ .env.local
    1   NEXT_PUBLIC_SANITY_PROJECT_ID="5q8erzvb"
    2   NEXT_PUBLIC_SANITY_DATASET="production"
    3   SANITY_API_TOKEN="skByVTxZKXxtWJnz77XOyGiIgq4UG3FfNPKOa1k7Fmn9ti98LYXJQ3poetEwnrUM1
```

7. Make sure that you get your values from sanity dashboard and paste the values here
8. To get your token to include it in your .env file go to sanity dashboard > API > Token and add a new token. Click on **add API token** and Make sure you give a **Editor** or **Developer** role. Save your token in .env file.
9. Your **dataset** will always be **production.**
10. After these steps compile the project by running the command
11. tsc && node importData.js

Part II: Define your Schema

1. In your **Practice Hackathon Project setup sanity:**
   learn-nextjs/step09_headless_cms/sanity/step00_sanity_setup at main · panaverse/learn-nextjs
2. After setup make a new file in sanity > schemaTypes named product.ts and copy the following code
3. export default {
4.   name: 'product',
5.   type: 'document',
6.   title: 'Product',
7.   fields: [
8.   {
9.   name: 'name',
10.   type: 'string',
11.   title: 'Product Name',
12.   },
13.   {
14.   name: 'description',
15.   type: 'string',
16.   title: 'Description'
17.   },
18.   {
19.   name: 'price',
20.   type: 'number',
21.   title: 'Product Price',
22.   },
23.   {
24.   name: 'discountPercentage',
25.   type: 'number',
26.   title: 'Discount Percentage',
27.   },
28.   {
29.   name: 'priceWithoutDiscount',
30.   type: 'number',
31.   title: 'Price Without Discount',
32.   description: 'Original price before discount'
33.   },
34.   {
35.   name:'rating',
36.   type:'number',
37.   title:'Rating',
38.   description:'Rating of the product'
39.   },
40.   {
41.   name: 'ratingCount',
42.   type: 'number',
43.   title: 'Rating Count',
44.   description: 'Number of ratings'
45.   },
46.   {
47.   name: 'tags',

```
48.         type: 'array',
49.         title: 'Tags',
50.         of: [{ type: 'string' }],
51.         options: {
52.             layout: 'tags'
53.         },
54.         description: 'Add tags like "new arrival", "bestseller", etc.'
55.     },
56.     {
57.         name: 'sizes',
58.         type: 'array',
59.         title: 'Sizes',
60.         of: [{ type: 'string' }],
61.         options: {
62.             layout: 'tags'
63.         },
64.         description: 'Add sizes like S , M , L , XL , XXL'
65.     },
66.     {
67.         name: 'image',
68.         type: 'image',
69.         title: 'Product Image',
70.         options: {
71.             hotspot: true // Enables cropping and focal point selection
72.         }
73.     }
74.     ]
75. };
76.
```

Part III: Fetch Data from CMS

How will you fetch data according to the sections:

✓ **API INTEGRATION PROCESS**


```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                                    ⌄ cmd  +  ⌄  ▢  🗑  ···  ∧
Uploading product: {
  _id: 'mockapi-product-undefined',
  _type: 'product',
  name: 'Vase Set',
  category: { name: 'Plant Pots', slug: 'plant-pots' },
  price: 150,
  quantityAvailable: undefined,
  isAvailable: undefined,
  createdAt: undefined,
  images: []
}
✅ Imported product: Vase Set
Uploading product: {
  _id: 'mockapi-product-undefined',
  _type: 'product',
  name: 'Marble Ease',
  category: { name: 'Crockory', slug: 'crockory' },
  price: 419,
  quantityAvailable: undefined,
  isAvailable: undefined,
  createdAt: undefined,
  images: []
}
✅ Imported product: Marble Ease
✅ Data import completed!

F:\GIAIC\import-api\import-api-data-in-sanity\dist>
```

✓ **ADJUSTMENTS MADE TO SCHEMAS.**


```
TS category.ts ✕

src > sanity > schemaTypes > TS category.ts > [∅] categorySchema
  1    import { defineType,defineField } from "sanity";
  2
  3    export const categorySchema = defineType({
  4        name: "category",
  5        title: "Category",
  6        type: "document",
  7        fields:[
  8            defineField({
  9                name: "name",
 10                title: "Name",
 11                type: "string",
 12                validation: (rule) => rule.required(),
 13            }),
 14            defineField({
 15                name: "slug",
 16                title: "Slug",
 17                type: "slug",
 18                validation: (rule) => rule.required(),
 19                options: {
 20                    source: "name",
 21                }
 22            })
 23        ]
 24    })
```

✓ **API INTEGRATION AND DATA MIGRATION**

```
  images: []
}
✅ Imported product: Marble Ease
✅ Data import completed!

F:\GIAIC\import-api\import-api-data-in-sanity\dist>
```
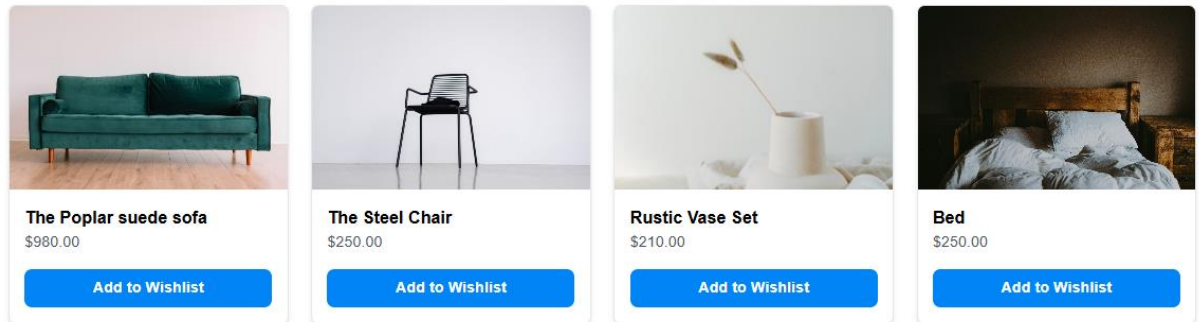
✓ **MIGRATION STEPS**

First clone from url provided by the Sir Ali Jawwad.
Second update data scheme
Set up urser ID, Data set and Copy Generated Taken in .env local file.
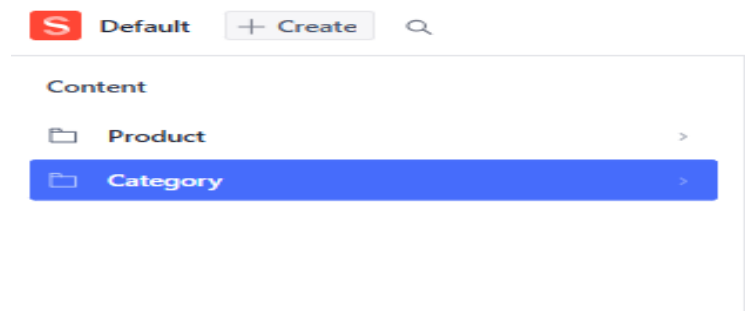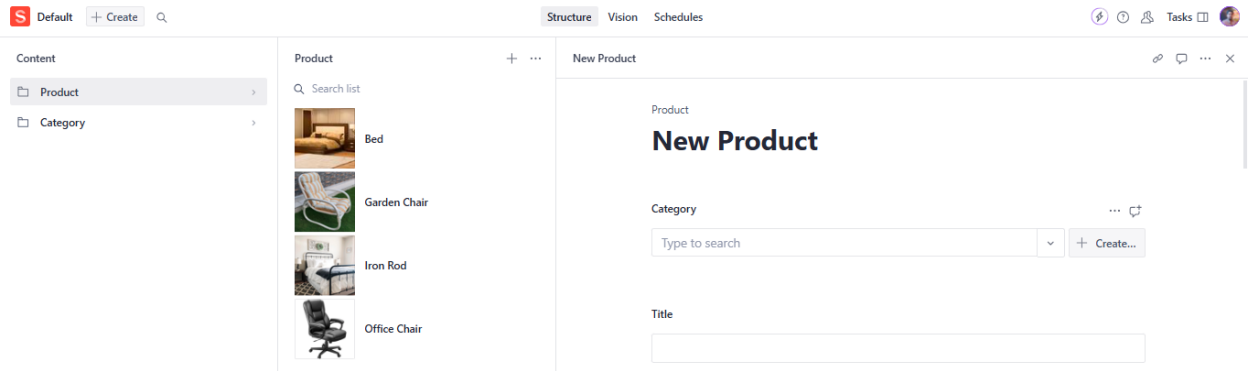The Run tsc && node importData.js

Product List

The Poplar suede sofa	The Steel Chair	Rustic Vase Set	Bed
$980.00	$250.00	$210.00	$250.00
Add to Wishlist	Add to Wishlist	Add to Wishlist	Add to Wishlist

✓ **TOOLS USED.**

VS Code
Sanity

✓ **POPULATED SANITY CMS FIELDS.**

✓ **CODE SNIPPETS FOR API INTEGRATION AND MIGRATION SCRIPTS.**

```
1   tsc && node importData.js
```

2. After setup make a new file in sanity > schemaTypes named product.ts and copy the following code

3.
```
1   export default {
2       name: 'product',
3       type: 'document',
4       title: 'Product',
5       fields: [
6           {
7               name: 'name',
8               type: 'string',
9               title: 'Product Name',
10          },
11          {
12              name: 'description',
13              type: 'string',
14              title: 'Description'
```

```
1   const query = await client.fetch(
2           `*[_type == "product" && "featured" in tags]{
3               _id,
4             name,
5             price,
6             discountPercentage,
7             tags,
8             "imageUrl": image.asset->url
9           }`
10      );
```