

LIVINGSTYLE



HACKATHON DAY

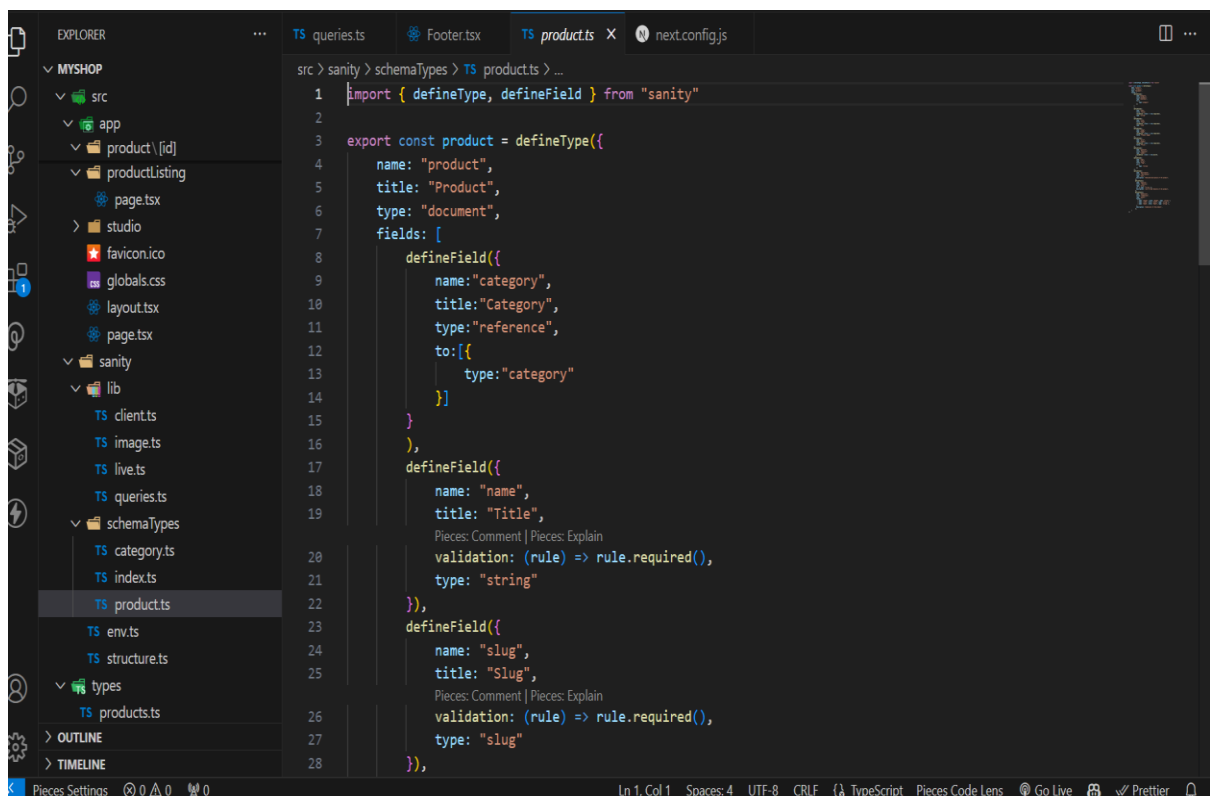
3

Day 3 - API Integration and Data Migration Report LivingStyle

Objective:

The goal for Day 3 was to integrate API data into Sanity CMS for the LivingStyle project, enabling dynamic content updates for the marketplace. Instead of manually entering data, the API integration provided a more efficient and scalable solution.

Schema Design



The screenshot shows a VS Code editor with the Explorer sidebar on the left displaying the project structure. The main editor window shows the file `src > sanity > schemaTypes > TS products > ...` with the following TypeScript code:

```
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    defineField({
17      name: "name",
18      title: "Title",
19      validation: (rule) => rule.required(),
20      type: "string"
21    }),
22    defineField({
23      name: "slug",
24      title: "Slug",
25      validation: (rule) => rule.required(),
26      type: "slug"
27    })
28  ]
29 })
```

The Explorer sidebar shows the following structure:

- MYSHOP
 - src
 - app
 - product\ [id]
 - productListing
 - page.tsx
 - studio
 - favicon.ico
 - globals.css
 - layout.tsx
 - page.tsx
 - sanity
 - lib
 - TS clients.ts
 - TS image.ts
 - TS live.ts
 - TS queries.ts
 - schemaTypes
 - TS category.ts
 - TS index.ts
 - TS product.ts
 - env.ts
 - structure.ts
 - types
 - TS products.ts

The status bar at the bottom indicates: Pieces Settings, 0 0 0, Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, TypeScript, Pieces Code Lens, Go Live, Prettier.

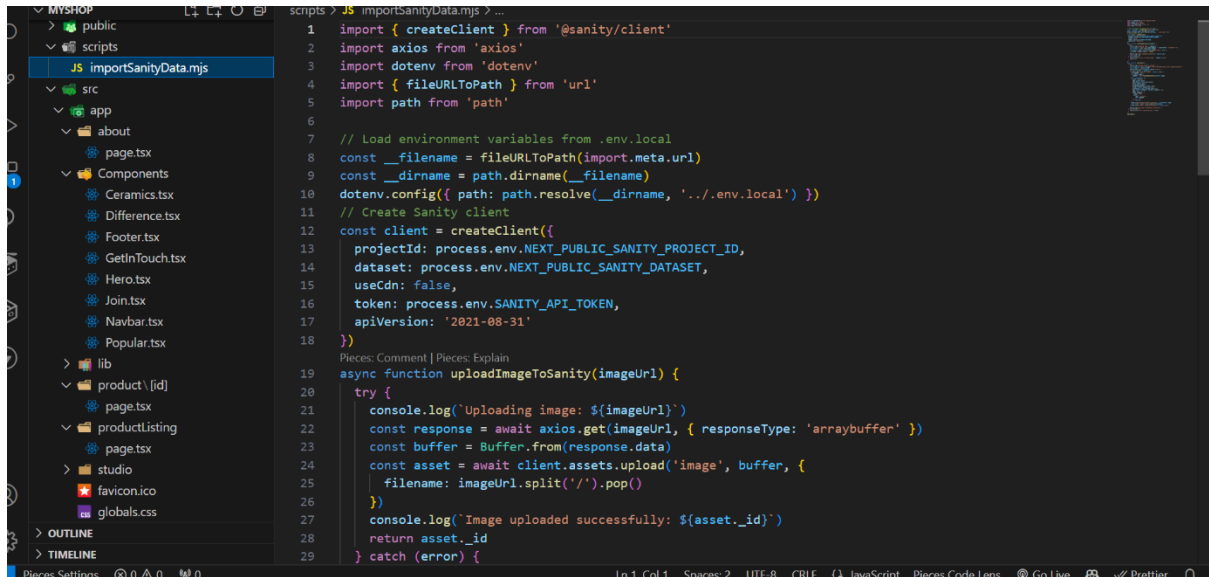
API Integration and Data Migration:

API Data Fetching:

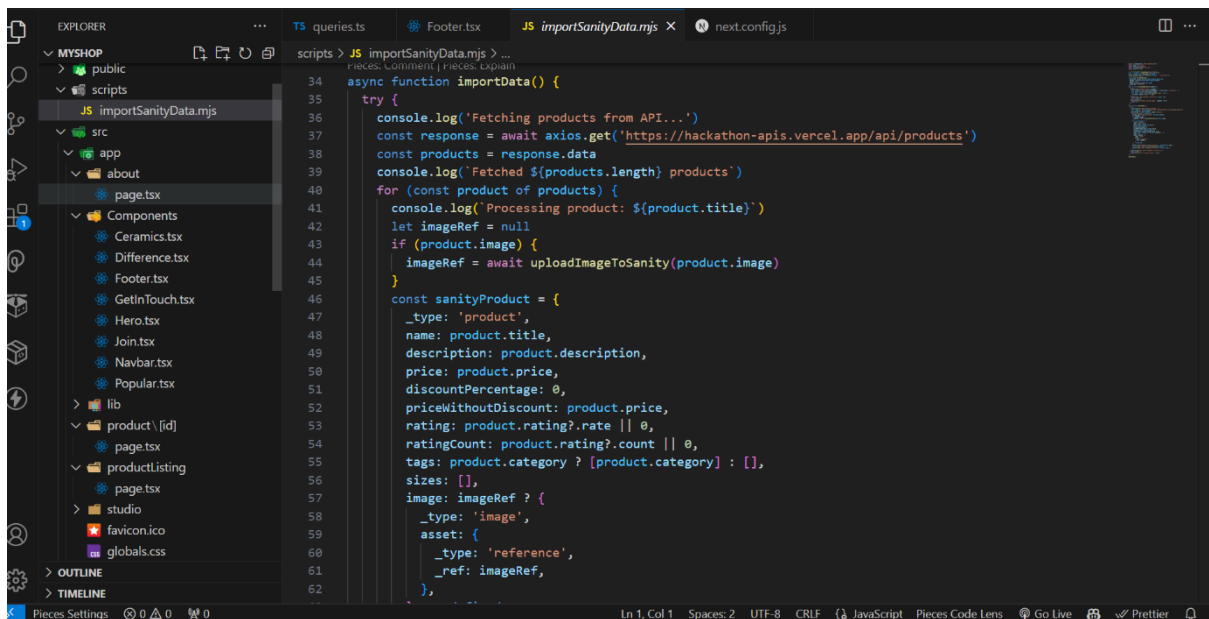
Fetches product data from an external API, which included details like images, titles, descriptions, prices, and category. This data was then mapped to the corresponding fields in the Sanity CMS schema.

Data Migration:

Using the Sanity CLI, exported the dataset from Sanity CMS for backup purposes and later re-imported it for testing. This migration ensured that all data was properly structured and displayed as intended on the frontend.

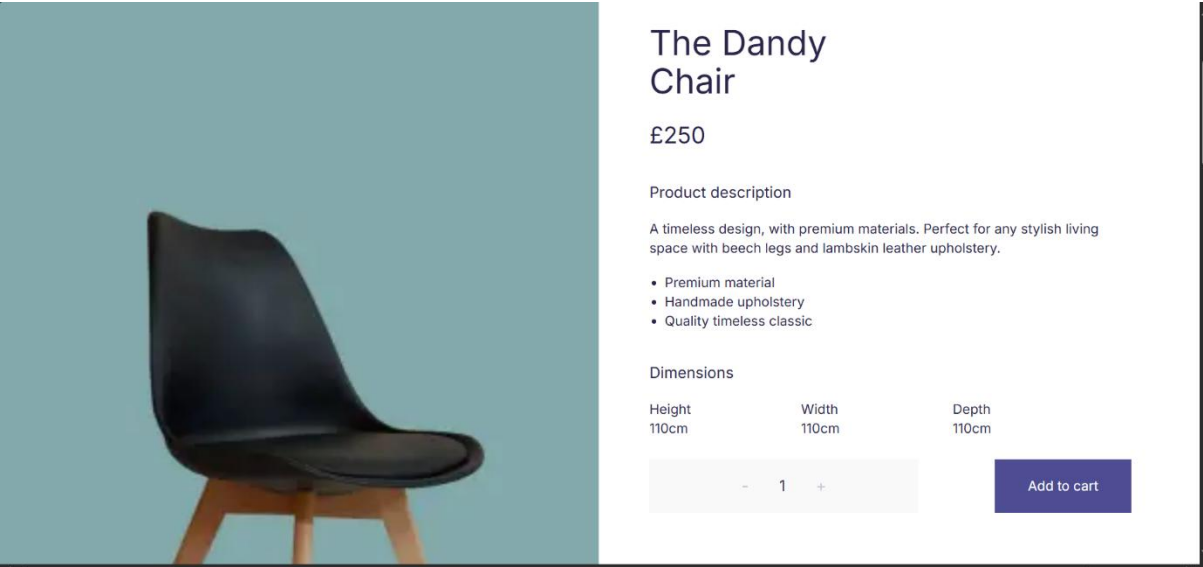
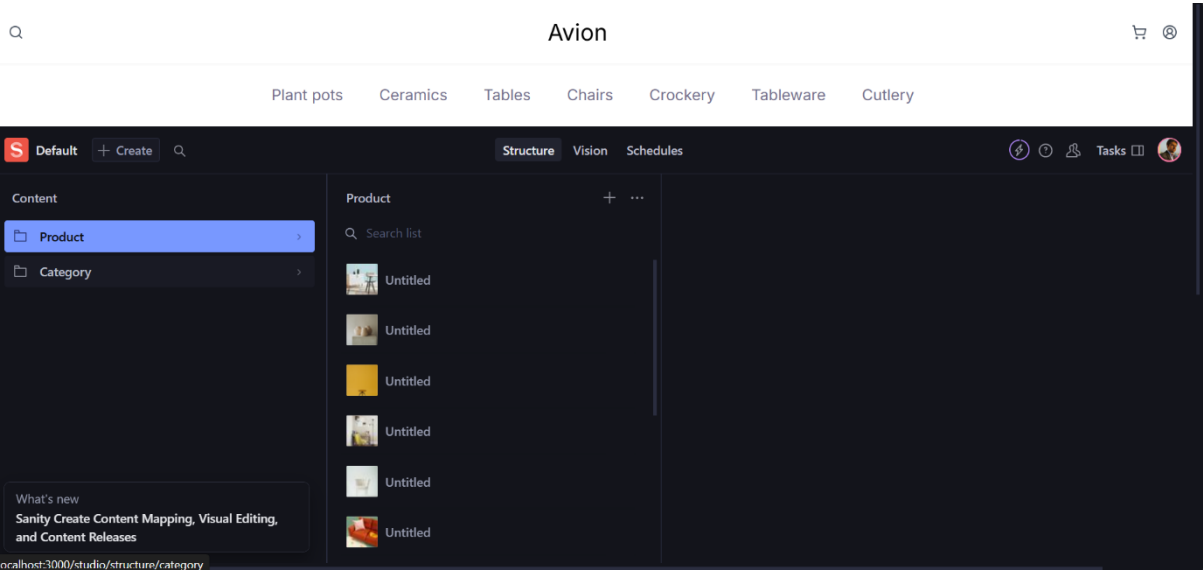


```
1 import { createClient } from '@sanity/client'
2 import axios from 'axios'
3 import dotenv from 'dotenv'
4 import { fileURLToPath } from 'url'
5 import path from 'path'
6
7 // Load environment variables from .env.local
8 const __filename = fileURLToPath(import.meta.url)
9 const __dirname = path.dirname(__filename)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



```
34 async function importData() {
35   try {
36     console.log('Fetching products from API...')
37     const response = await axios.get('https://hackathon-apis.vercel.app/api/products')
38     const products = response.data
39     console.log('Fetched ${products.length} products')
40     for (const product of products) {
41       console.log('Processing product: ${product.title}')
42       let imageRef = null
43       if (product.image) {
44         imageRef = await uploadImageToSanity(product.image)
45       }
46       const sanityProduct = {
47         _type: 'product',
48         name: product.title,
49         description: product.description,
50         price: product.price,
51         discountPercentage: 0,
52         priceWithoutDiscount: product.price,
53         rating: product.rating?.rate || 0,
54         ratingCount: product.rating?.count || 0,
55         tags: product.category ? [product.category] : [],
56         sizes: [],
57         image: imageRef ? {
58           _type: 'image',
59           asset: {
60             _type: 'reference',
61             _ref: imageRef,
62           },
63         } : null
64       }
65     }
66   } catch (error) {
67     console.error('Error importing data: ', error)
68   }
69 }
```

Data Displayed in Sanity and then Frontend:



Check List:

Schema Design	✓
API Data Fetching	✓
Data Migration	✓
Data Displayed in Sanity	✓
Data Displayed on Frontend	✓