

## Day 3 Hackathon Report: Data Migration and API Integration 🚀

---

### 1. API Integration Process

- **Objective:** Seamlessly integrate multiple APIs to fetch dynamic product and category data into the frontend.
  - **Tools Used:**
    - **Axios:** For efficient API calls.
    - **.env:** For secure environment variable configuration.
  - **Steps Involved:**
    1. Retrieved data from multiple APIs, enabling dynamic updates for both products and categories.
    2. Implemented **data validation** and **error handling** to ensure robust and scalable API integrations.
    3. Mapped the fetched API data to pre-provided schemas in Sanity CMS for efficient data handling.
- 

### 2. Adjustments Made to Schemas

- **Objective:** Configure pre-existing schemas within Sanity CMS to align perfectly with the incoming API data.
  - **Steps Taken:**
    1. Integrated **Product Schema:** Ensured compatibility with dynamic pricing, descriptions, and category fields.
    2. Applied **Category Schema:** Configured pre-defined fields for a hierarchical category structure to match API responses.
    3. Verified that pre-existing schema configurations met the project requirements, ensuring smooth ingestion and storage of data in Sanity CMS.
-

### 3. Data Migration Steps and Tools Used

- **Objective:** Automate and streamline the migration process to Sanity CMS while utilizing the provided schemas.
  - **Tools Used:**
    - **Custom Migration Script:** Developed in Node.js to automate data transfer and integration.
    - **Sanity CMS APIs:** Leveraged for populating fields with data fetched via API.
  - **Steps:**
    1. Extracted raw data from APIs and transformed it into a format compatible with the pre-existing Sanity schemas.
    2. Used the custom script to bulk-upload products and categories into Sanity CMS with minimal manual intervention.
    3. Ensured data consistency and resolved any API-to-CMS mapping issues.
- 

### Screenshots

1. **API Calls in Action:**
  - Screenshot showcasing API request and response flow in the console

```
C:\Users\test\Desktop\Governor Projects\Comfortly_H8\hackathon_8>npm run migrate
```

```
> hackathon_8@0.1.0 migrate
> node scripts/migrate.mjs
```

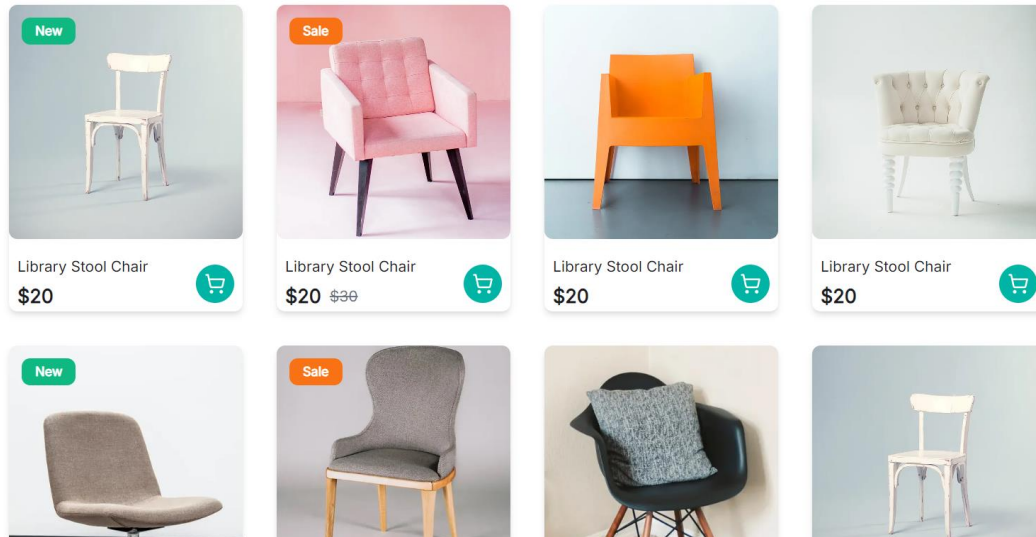
```
Starting data migration...
Migrating category: Wing Chair
Migrated category: Wing Chair (ID: 26fd7176-3c4d-40fc-a73a-3b85a9b5e15f)
Migrating category: Wooden Chair
Migrated category: Wooden Chair (ID: 407a8583-6203-4f61-becf-8e8b4c5461b6)
Migrating category: Desk Chair
Migrated category: Desk Chair (ID: b5710116-09af-4d0e-aa9a-dcd02fe919a9)
Migrating product: SleekSpin
Migrated product: SleekSpin (ID: PadjJq79diW6CWHYoYbrud)
Migrating product: Citrus Edge
Migrated product: Citrus Edge (ID: PadjJq79diW6CWHYoYbrz4)
Migrating product: Rose Luxe Armchair
Migrated product: Rose Luxe Armchair (ID: PadjJq79diW6CWHYoYbs8p)
Migrating product: Library Stool Chair
Migrated product: Library Stool Chair (ID: grIDCMNzKakH6Y34f10JMX)
Migrating product: Modern Cozy
Migrated product: Modern Cozy (ID: 4GgGIux0xATsSIdIhnQnOF)
Migrating product: Scandi Dip Set
Migrated product: Scandi Dip Set (ID: PadjJq79diW6CWHYoYbsM8)
Migrating product: Citrus Edge
Migrated product: Citrus Edge (ID: 4GgGIux0xATsSIdIhnQnzz)
Migrating product: Rose Luxe Armchair
Migrated product: Rose Luxe Armchair (ID: grIDCMNzKakH6Y34f10Jgp)
Migrating product: SleekSpin
```

```
Migrating product: Rose Luxe Armchair
Migrated product: Rose Luxe Armchair (ID: PadjJq79diW6CWHYoYbs8p)
Migrating product: Library Stool Chair
Migrated product: Library Stool Chair (ID: grIDCMNzKakH6Y34f10JMX)
Migrating product: Modern Cozy
Migrated product: Modern Cozy (ID: 4GgGIux0xATsSIdIhnQnOF)
Migrating product: Scandi Dip Set
Migrated product: Scandi Dip Set (ID: PadjJq79diW6CWHYoYbsM8)
Migrating product: Citrus Edge
Migrated product: Citrus Edge (ID: 4GgGIux0xATsSIdIhnQnzz)
Migrating product: Rose Luxe Armchair
Migrated product: Rose Luxe Armchair (ID: grIDCMNzKakH6Y34f10Jgp)
Migrating product: SleekSpin
Migrated product: SleekSpin (ID: grIDCMNzKakH6Y34f10Jq9)
Migrating product: Library Stool Chair
Migrated product: Library Stool Chair (ID: PadjJq79diW6CWHYoYbsbD)
Migrating product: Modern Cozy
Migrated product: Modern Cozy (ID: grIDCMNzKakH6Y34f10K9L)
Migrating product: Scandi Dip Set
Migrated product: Scandi Dip Set (ID: PadjJq79diW6CWHYoYbsrB)
Migrating product: Nordic Spin
Migrated product: Nordic Spin (ID: PadjJq79diW6CWHYoYbt03)
Migrating product: Gray Elegance
Migrated product: Gray Elegance (ID: PadjJq79diW6CWHYoYbt8v)
Migrating product: Ivory Charm
Migrated product: Ivory Charm (ID: grIDCMNzKakH6Y34f10KBX)
Data migration completed successfully!
```

## 2. Dynamic Data on Frontend:

- Screenshot showing products and categories dynamically fetched and displayed in the frontend.

### Our Products

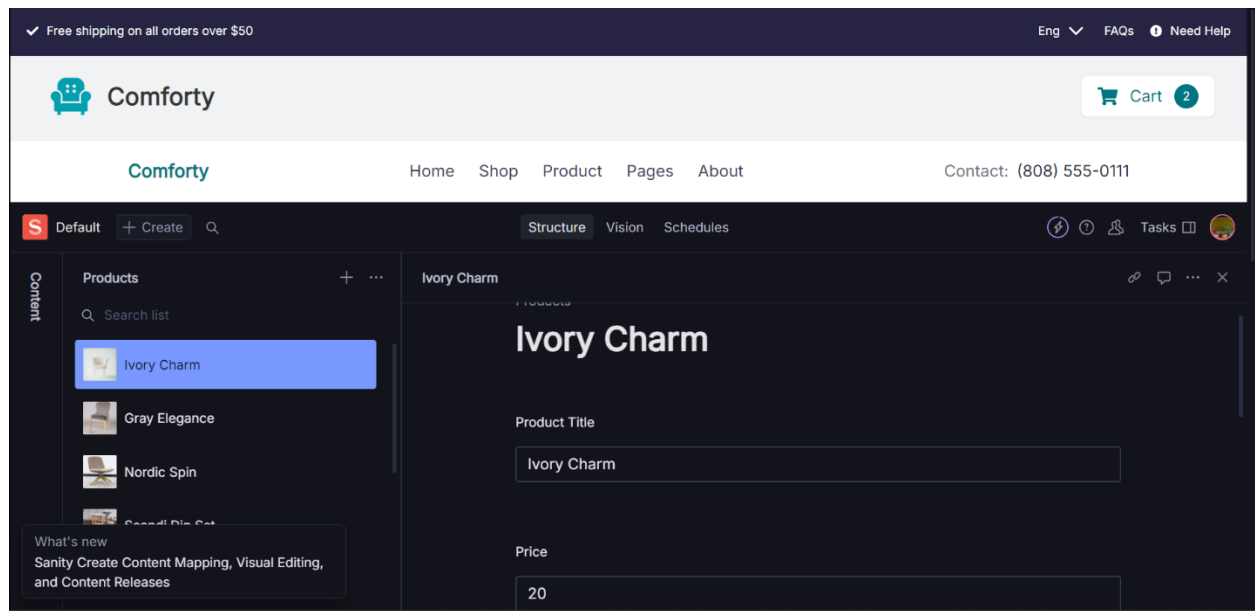


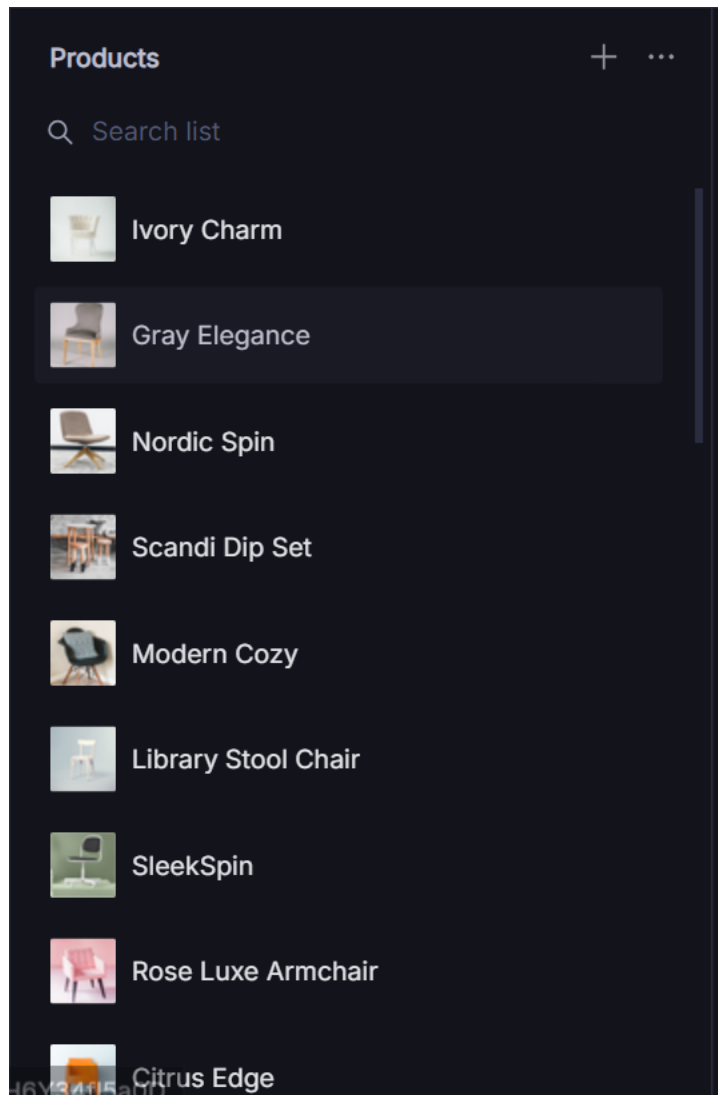
### Top Categories

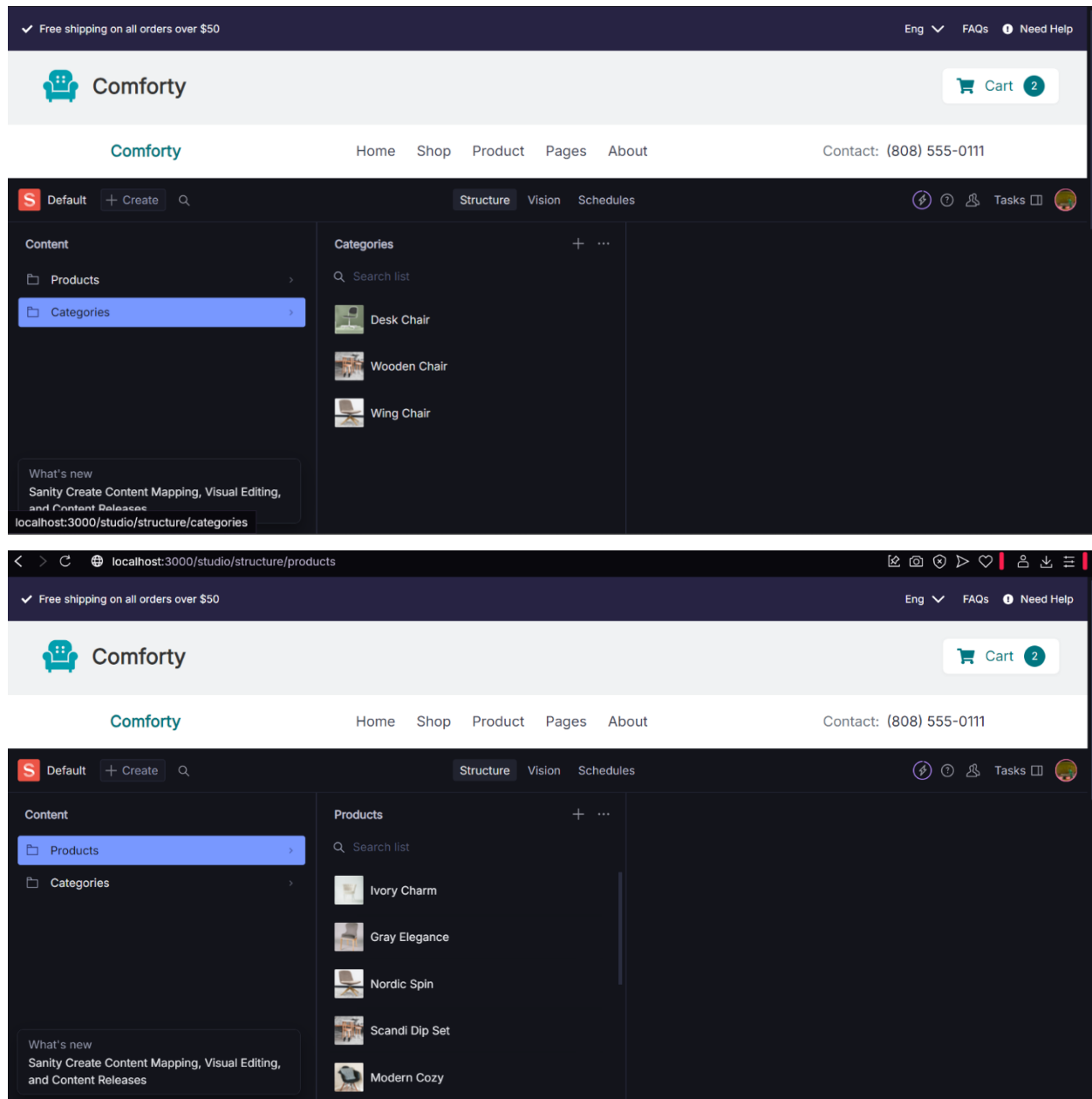


### 3. Sanity CMS Fields Populated:

- Screenshot of populated product and category fields in Sanity CMS, reflecting successful data migration and integration.







#### 4. Code Files Screenshot:

```
src > sanity > schemaTypes > TS products.ts > ...
1  import { defineType } from "sanity";
2
3  export const productSchema = defineType({
4    name: "products",
5    title: "Products",
6    type: "document",
7    fields: [
8      {
9        name: "title",
10       title: "Product Title",
11       type: "string",
12     },
13     {
14       name: "price",
15       title: "Price",
16       type: "number",
17     },
18     {
19       title: "Price without Discount",
20       name: "priceWithoutDiscount",
21       type: "number",
22     },
23     {
24       name: "badge",
25       title: "Badge",
26       type: "string",
27     },
28     {
29       name: "image",
30       type: "image",
31     },
32   ],
33 });
```

```
src > sanity > schemaTypes > TS categories.ts > ...
1  import { defineType } from "sanity";
2
3  export const categorySchema = defineType({
4    name: 'categories',
5    title: 'Categories',
6    type: 'document',
7    fields: [
8      {
9        name: 'title',
10       title: 'Category Title',
11       type: 'string',
12     },
13     {
14       name: 'image',
15       title: 'Category Image',
16       type: 'image',
17     },
18     {
19       title: 'Number of Products',
20       name: 'products',
21       type: 'number',
22     },
23   ],
24 });
```



```
src > sanity > schemaTypes > TS index.ts > [🔍] schema > [🔗] types
1  import { type SchemaTypeDefinition } from 'sanity'
2  import { productSchema } from './product'
3  import { categorySchema } from './categories'
4
5  export const schema: { types: SchemaTypeDefinition[] } = {
6    types: [productSchema, categorySchema],
7  }
8
```

```
.env
1  NEXT_PUBLIC_SANITY_PROJECT_ID="mbakvv56"
2  NEXT_PUBLIC_SANITY_DATASET="production"
3  NEXT_PUBLIC_SANITY_AUTH_TOKEN=skeDQKNUVuGYoCnDTZJsHB91YSQ4lA61HbiWOESshyT3b7wi4coMZFI3s2CFEaSbRSASNuakfVHBpc60Bz
  0R29ifLao4MQwUgGnsW4ZJyxYs3J6Wst87t04ksROU9erRFZ96ugopqby1gK5KzPMGDeBlwDfSrrBS3phX3QYrFPQqglJG52nB
```

---

## Insights Gained

- Configuring pre-defined schemas saved time and allowed for seamless integration with APIs.
- Automation with custom scripts significantly reduced manual effort, ensuring efficiency and scalability.
- Proper **data validation** and mapping processes were critical for smooth data handling and a consistent user experience.