Day 3 Hackathon Report: Data Migration and API Integration 🚀

1. API Integration Process

Objective: Ensure seamless integration of multiple APIs to dynamically fetch product and category data for the frontend.

Tools Used:

- **Axios**: Facilitates efficient API calls.
- **.env**: Secures environment variables for configuration.

Implementation Steps:

1. Retrieved and integrated data from multiple APIs to enable real-time updates for products and categories.
2. Applied data validation techniques and error-handling mechanisms to enhance integration reliability and scalability.
3. Mapped the retrieved API data to predefined schemas within Sanity CMS for efficient data storage and management.

---

2. Schema Adjustments

Objective: Optimize Sanity CMS schemas to align seamlessly with the incoming API data structure.

Key Modifications:

1. **Product Schema**: Adapted schema fields to support dynamic pricing, descriptions, and category attributes.
2. **Category Schema**: Enhanced hierarchical structure to match API responses for better organization.
3. **Validation & Testing**: Ensured that the modified schemas aligned with project requirements, facilitating smooth data ingestion and storage.

---

3. Data Migration Process & Tools Used

Objective: Automate and streamline the data migration process into Sanity CMS while ensuring consistency.

Tools Utilized:

- **Custom Migration Script**: Developed using Node.js to automate data transfer and reduce manual input.
- **Sanity CMS APIs**: Leveraged to efficiently populate CMS fields with fetched API data.

Steps Executed:

1. Extracted raw data from APIs and transformed it into a structured format compatible with Sanity schemas.
2. Utilized a custom-built script to bulk-upload products and categories into Sanity CMS, minimizing manual work.
3. Verified data consistency, addressing any mapping discrepancies between the APIs and CMS.

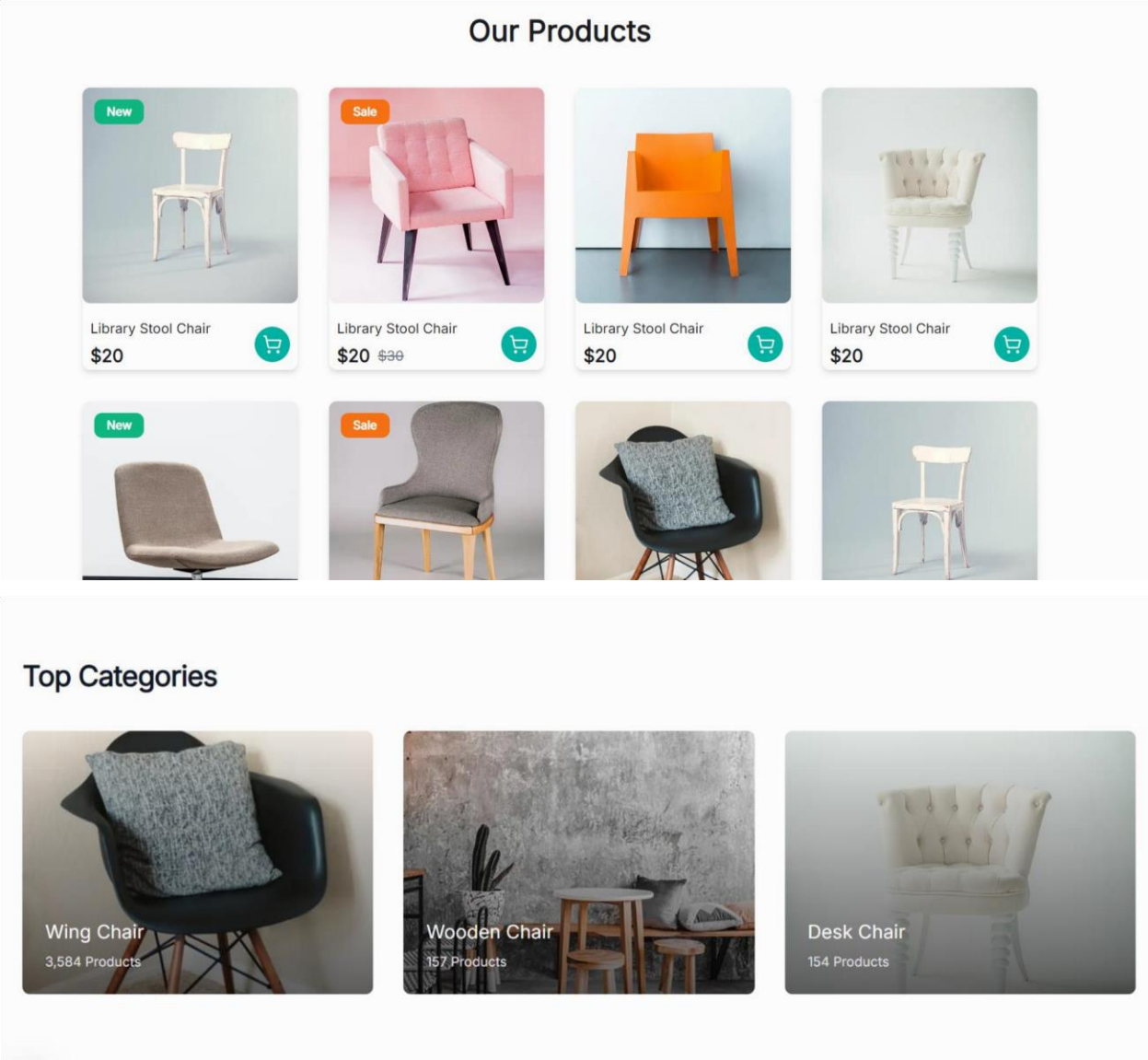
Screenshots

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

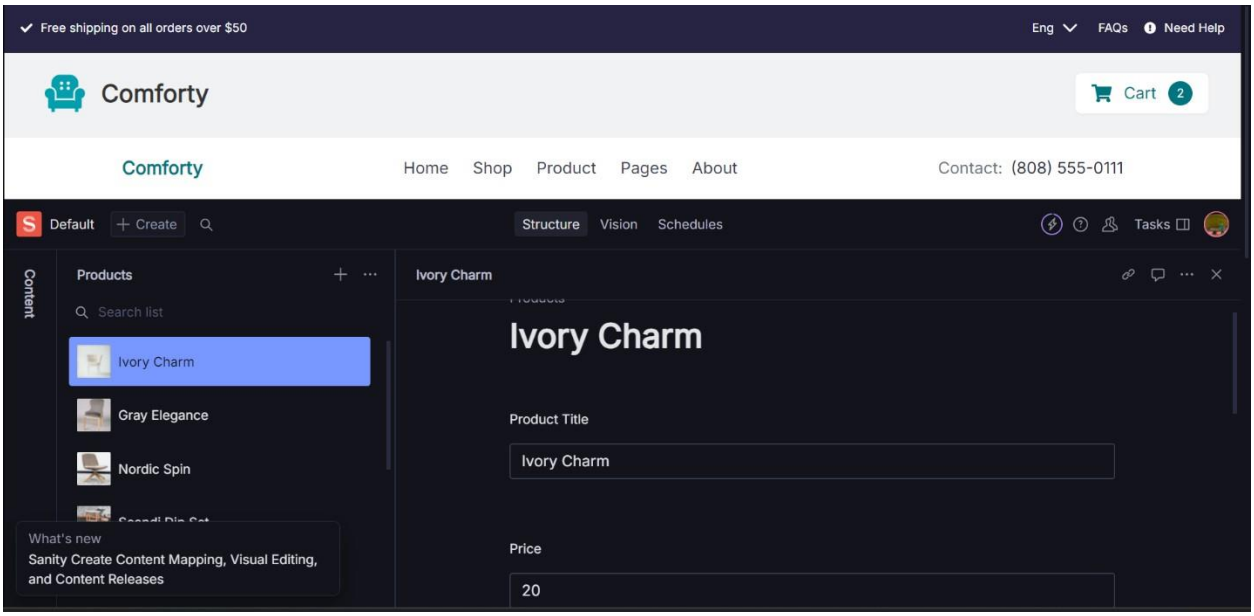
```
C:\Users\test\Desktop\Governor Projects\Comfortly_H8\hackathon_8>npm run migrate

> 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: grIDCMNzKakH6Y34flOJMX)
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: grIDCMNzKakH6Y34flOJgp)
Migrating product: SleekSpin
```

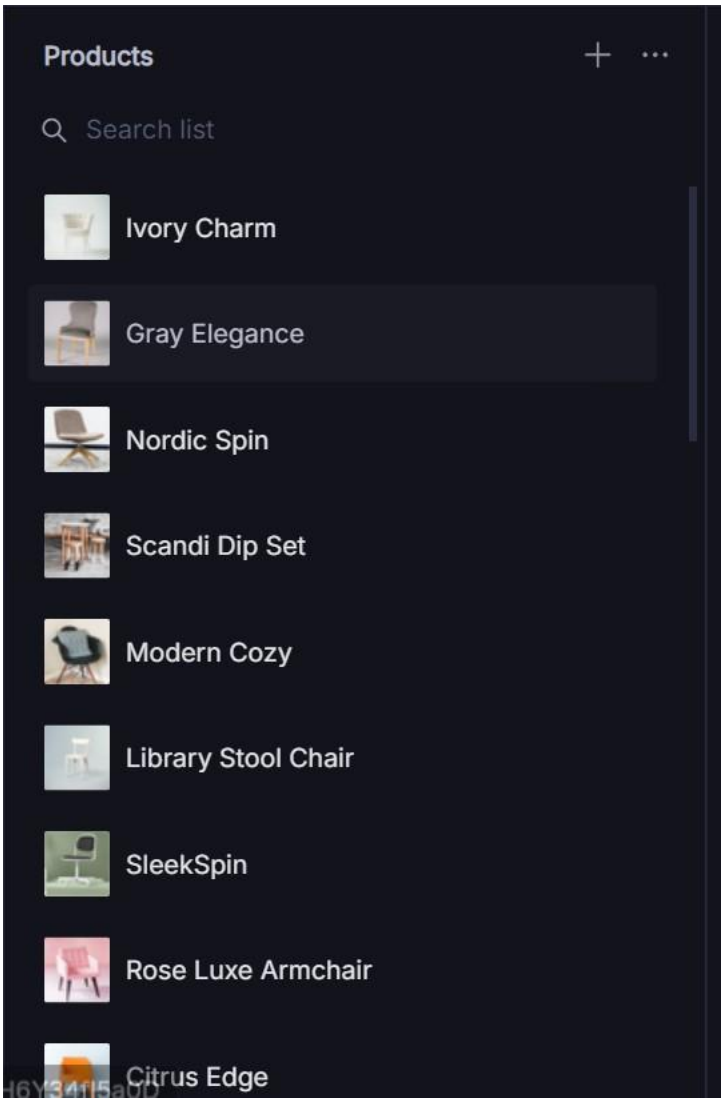
```
Migrating product: Rose Luxe Armchair
Migrated product: Rose Luxe Armchair (ID: PadjJq79diW6CWHYoYbs8p)
Migrating product: Library Stool Chair
Migrated product: Library Stool Chair (ID: grIDCMNzKakH6Y34flOJMX)
Migrating product: Modern Cozy
Migrated product: Modern Cozy (ID: 4GgGIux0xATsSIdIhnQnOF)
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: grIDCMNzKakH6Y34flOJgp)
Migrating product: SleekSpin
Migrated product: SleekSpin (ID: grIDCMNzKakH6Y34flOJq9)
Migrating product: Library Stool Chair
Migrated product: Library Stool Chair (ID: PadjJq79diW6CWHYoYbsbD)
Migrating product: Modern Cozy
Migrated product: Modern Cozy (ID: grIDCMNzKakH6Y34flOK9L)
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: grIDCMNzKakH6Y34flOKBX)
Data migration completed successfully!
```

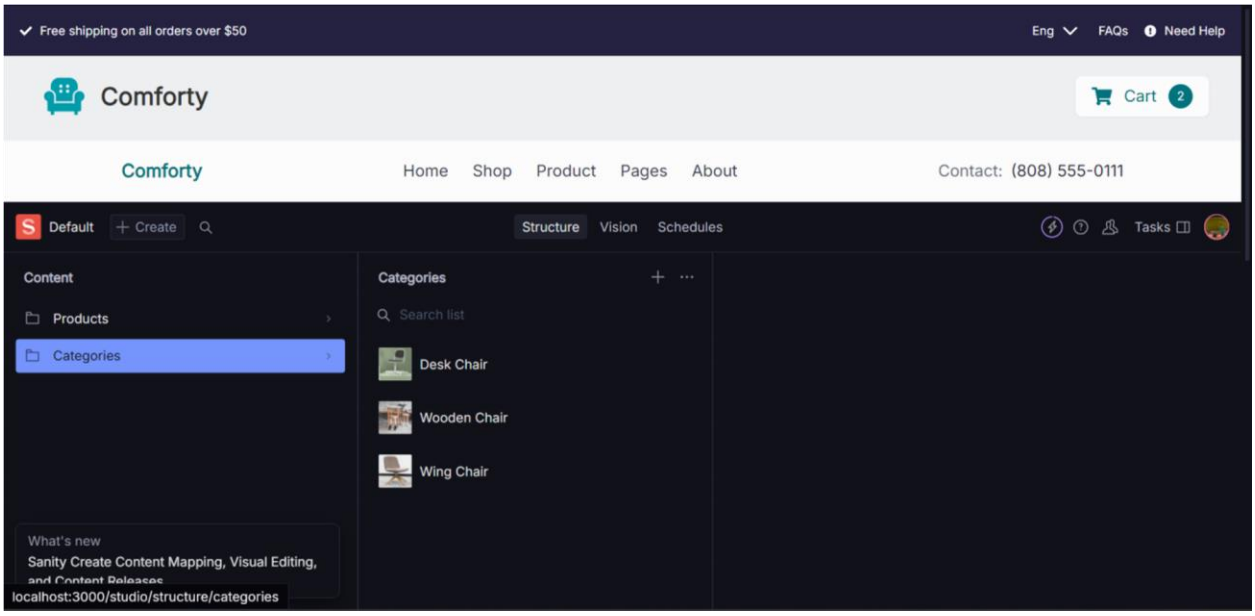
2. **Dynamic Data on Frontend:** ○ Screenshot showing products and categories dynamically fetched and displayed in the frontend.

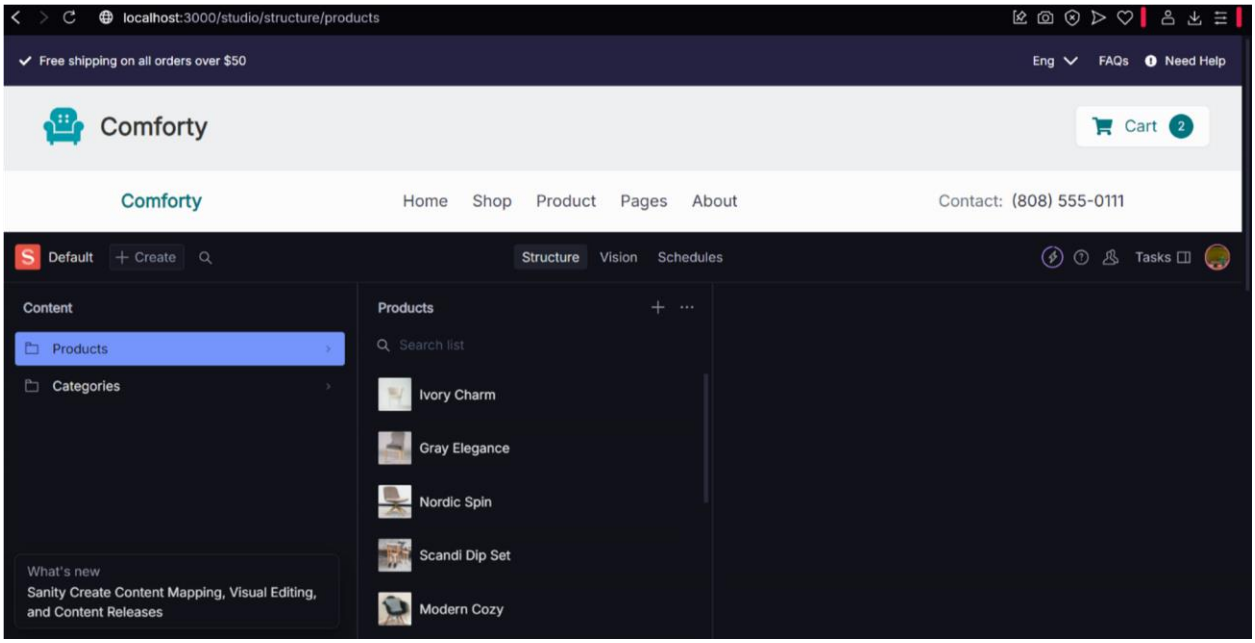


3. **Sanity CMS Fields Populated:**
    - Screenshot of populated product and category fields in Sanity CMS, reflecting successful data migration and integration.


Free shipping on all orders over $50
Eng
FAQs
Need Help
Comforty
Cart
2
Comforty
Home
Shop
Product
Pages
About
Contact: (808) 555-0111
Default
Create
Structure
Vision
Schedules
Tasks
Content
Products
Search list
Ivory Charm
Gray Elegance
Nordic Spin
What's new
Sanity Create Content Mapping, Visual Editing, and Content Releases
Ivory Charm
Ivory Charm
Product Title
Ivory Charm
Price
20


Products
Search list
Ivory Charm
Gray Elegance
Nordic Spin
Scandi Dip Set
Modern Cozy
Library Stool Chair
SleekSpin
Rose Luxe Armchair
Citrus Edge





4. **Code Files Screenshot:**

src > sanity > schemaTypes > TS product.ts > ...

```ts
import { defineType } from "sanity";

export const productSchema = defineType({
  name: "products",
  title: "Products",
  type: "document",
  fields: [
    {
      name: "title",
      title: "Product Title",
      type: "string",
    },
    {
      name: "price",
      title: "Price",
      type: "number",
    },
    {
      title: "Price without Discount",
      name: "priceWithoutDiscount",
      type: "number",
    },
    {
      name: "badge",
      title: "Badge",
      type: "string",
    },
    {
      name: "image",
```

src > sanity > schemaTypes > TS categories.ts > ...

```ts
import { defineType } from "sanity";

export const categorySchema = defineType({
    name: 'categories',
    title: 'Categories',
    type: 'document',
    fields: [
        {
            name: 'title',
            title: 'Category Title',
            type: 'string',
        },
        {
            name: 'image',
            title: 'Category Image',
            type: 'image',
        },
        {
            title: 'Number of Products',
            name: 'products',
            type: 'number',
        }
    ],
});
```

src > sanity > schemaTypes > TS index.ts > schema > types

```ts
import { type SchemaTypeDefinition } from 'sanity'
import { productSchema } from './product'
import { categorySchema } from './categories'

export const schema: { types: SchemaTypeDefinition[] } = {
  types: [productSchema , categorySchema],
}

```

.env

```
NEXT_PUBLIC_SANITY_PROJECT_ID="mbakvv56"
NEXT_PUBLIC_SANITY_DATASET="production"
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.