Hackathon Day 3 - API Integration and Data Migration

With Sanity CMS

API Integration

- **API Integration**: Connect external APIs for product management, user authentication, and payment processing.
- **Data Migration**: Move content from Sanity CMS to the new system and ensure correct frontend display.

Outcome: Fully functional features and accurate data on the marketplace.

API Integration Steps:

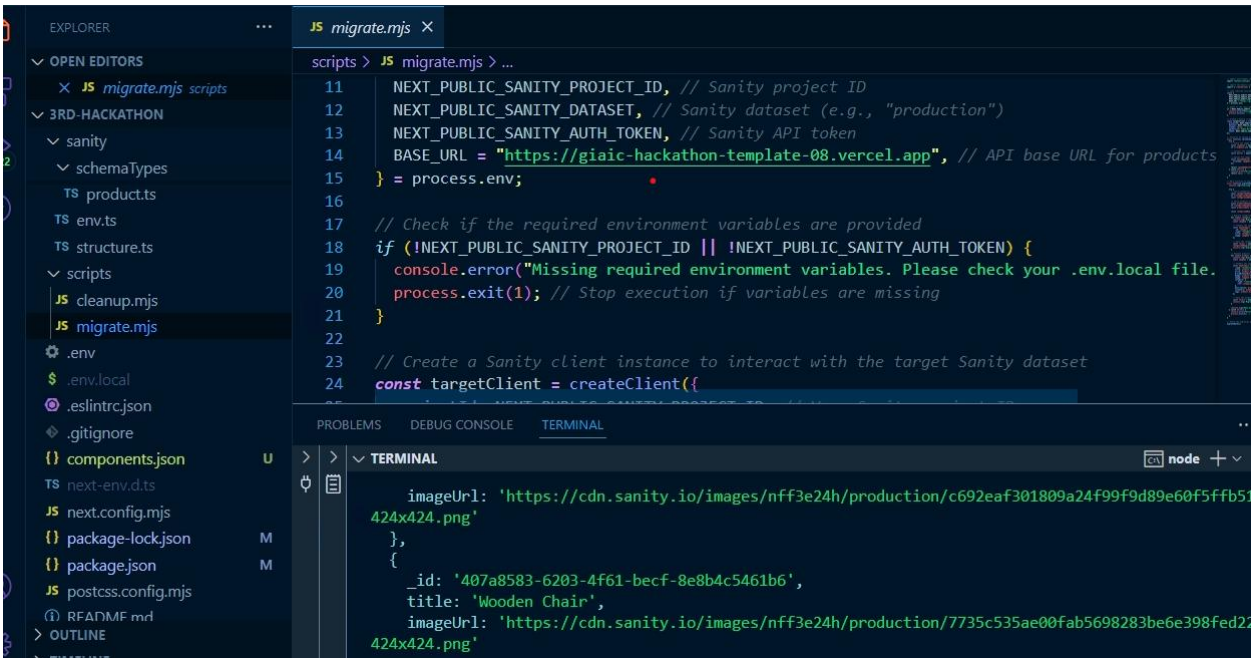
1. **Define Endpoints**:
    - Identify API endpoints for products, orders, customer data, and payment systems (e.g., EasyPaisa, JazzCash).
2. **Sanity Integration**:
    - Use the Sanity client to manage product, customer, and order data.
    - Create schema models and queries to fetch necessary data.
3. **API Communication**:
    - Utilize Axios or the fetch API to interact with external APIs.
    - Ensure efficient data fetching in frontend components.
4. **Error Handling**:
    - Implement robust mechanisms for handling API request errors.
    - Display clear, user-friendly error messages for failed operations.
5. **Data Mapping**:
    - Accurately format API data for proper frontend display.
    - Migrate any required legacy data for compatibility.
6. **Testing**:
    - Validate API responses thoroughly.
    - Perform end-to-end testing to ensure smooth platform functionality.

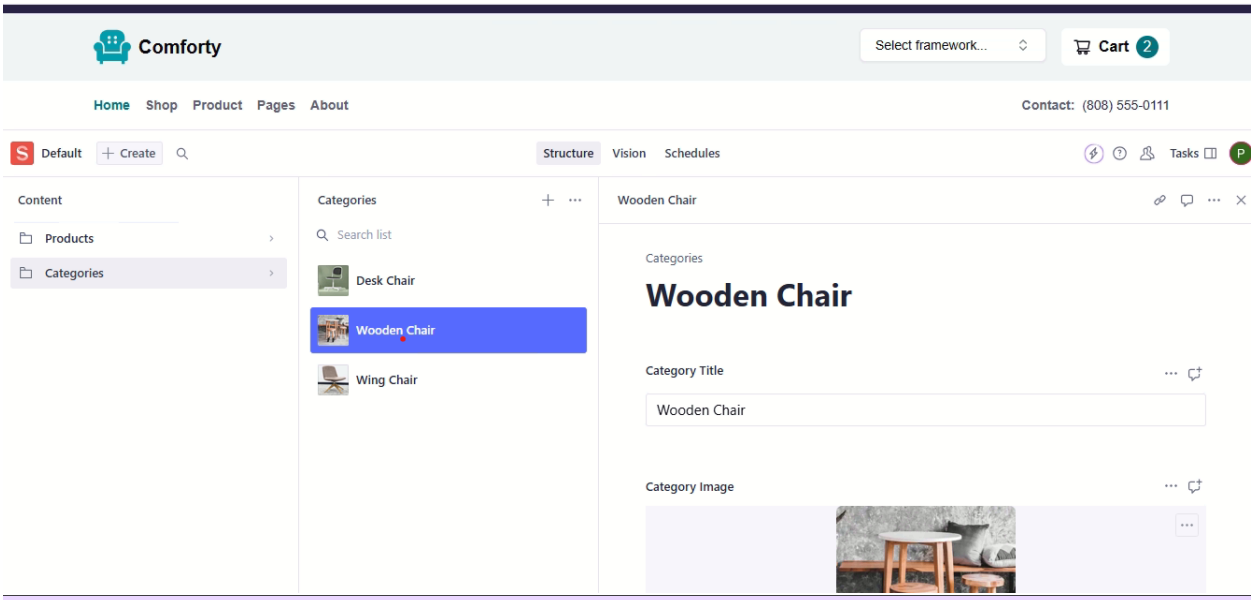
Outcomes:

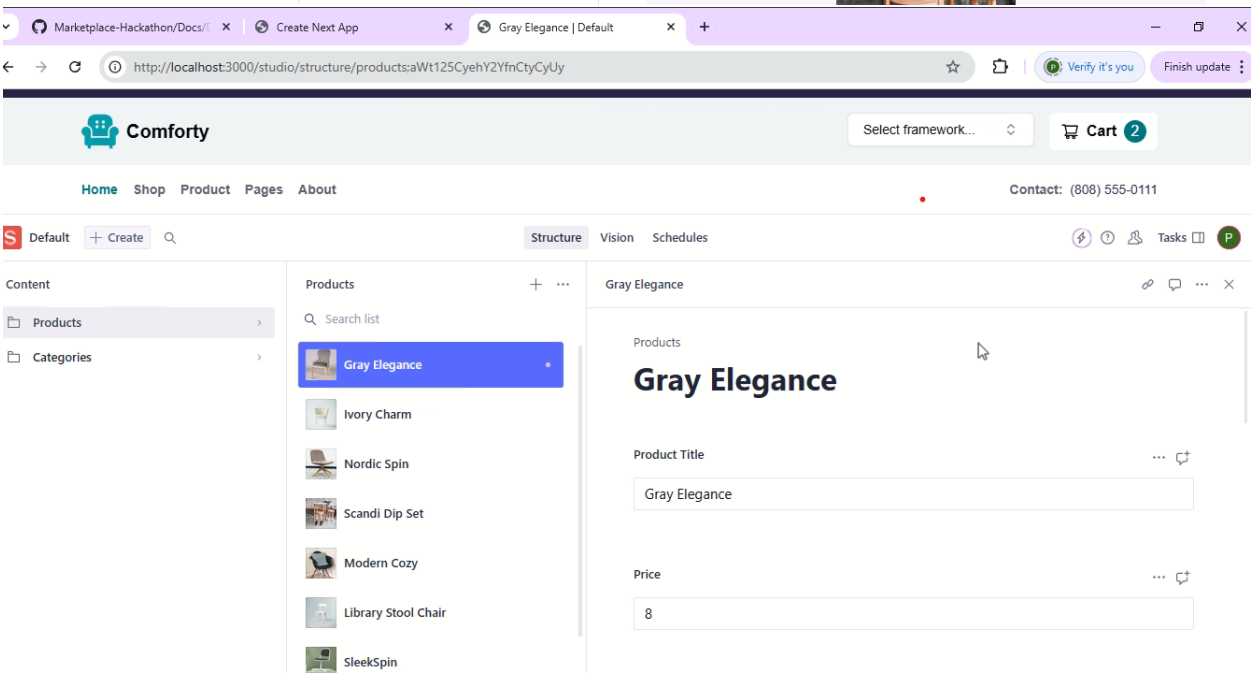
- Successfully integrated APIs with key functionalities operational.
- Sanity content migrated and accurately reflected on the frontend.

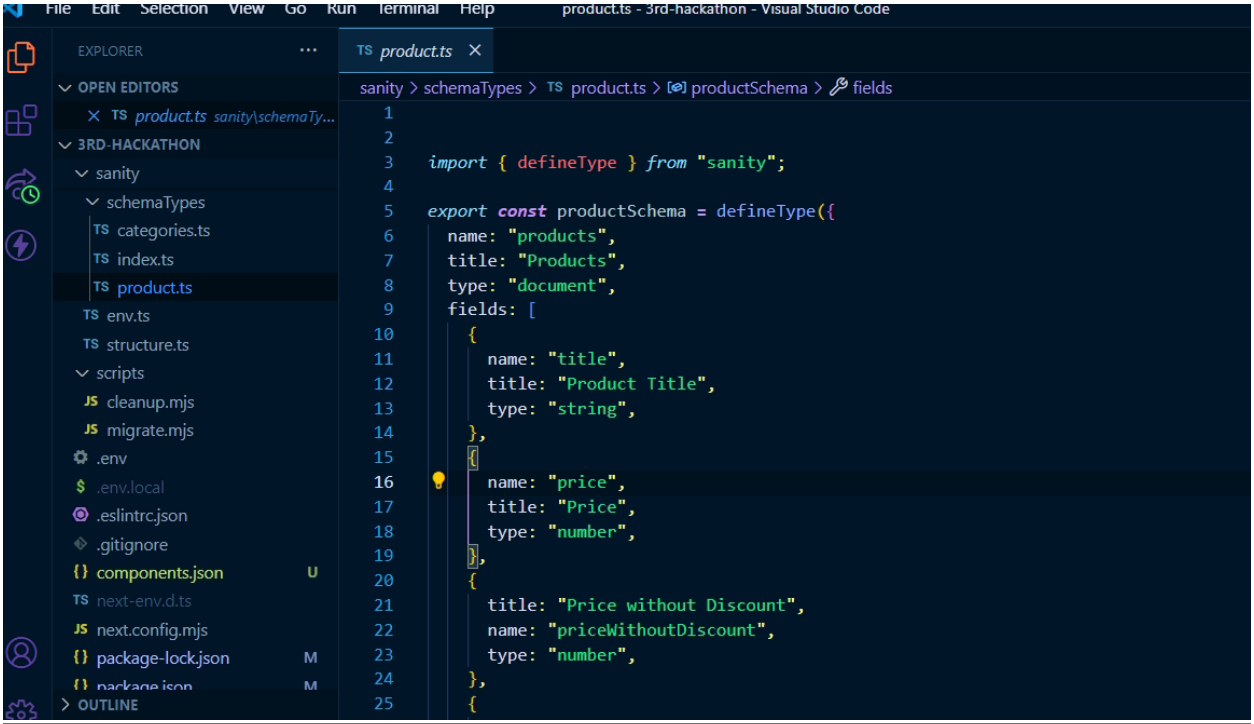
Migration Workflow and Resources Utilized:

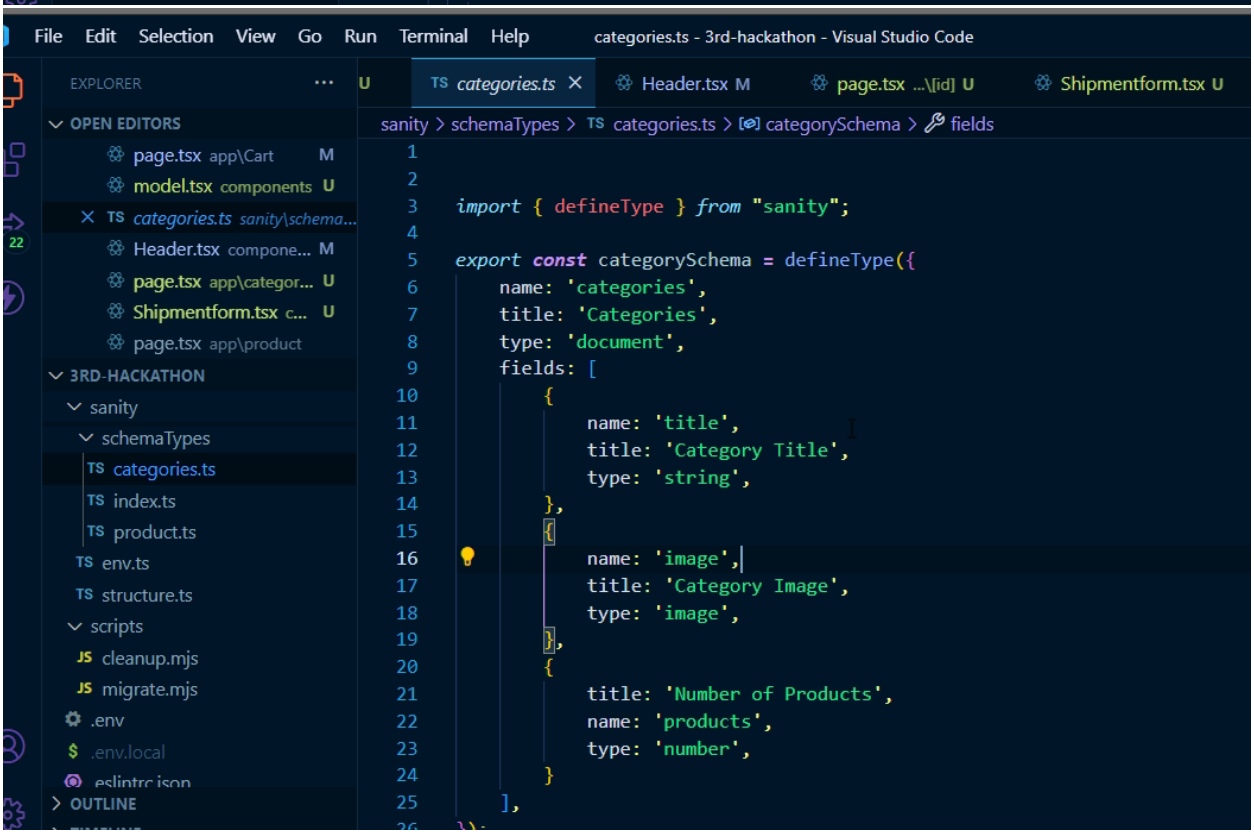
1. **Data Structuring**:
    - Organize product details into the new Sanity schema, including fields like category, price, stock, and description.
2. **Migration Execution**:
    - Deploy a customized migration script to transfer data seamlessly from the legacy system or CSV files into Sanity CMS.
3. **Sanity Toolkit**:
    - Leverage the Sanity CLI for efficient dataset imports, exports, and management tasks.
4. **Validation and Testing**:
    - Conduct thorough checks to ensure data accuracy, integrity, and proper relationships within the migrated datasets.




Comforty
Select framework...
Cart 2
Home Shop Product Pages About
Contact: (808) 555-0111
Default
Create
Structure Vision Schedules
Tasks
Content
Products
Categories
Categories
Search list
Desk Chair
Wooden Chair
Wing Chair
Wooden Chair
Categories
Wooden Chair
Category Title
Wooden Chair
Category Image


Marketplace-Hackathon/Docs/
Create Next App
Gray Elegance | Default
http://localhost:3000/studio/structure/products;aWt125CyehY2YfnCtyCyUy
Verify it's you
Finish update
Comforty
Select framework...
Cart 2
Home Shop Product Pages About
Contact: (808) 555-0111
Default
Create
Structure Vision Schedules
Tasks
Content
Products
Categories
Products
Search list
Gray Elegance
Ivory Charm
Nordic Spin
Scandi Dip Set
Modern Cozy
Library Stool Chair
SleekSpin
Gray Elegance
Products
Gray Elegance
Product Title
Gray Elegance
Price
8


File Edit Selection View Go Run Terminal Help
product.ts - 3rd-hackathon - Visual Studio Code
EXPLORER
OPEN EDITORS
product.ts sanity\schemaTy...
3RD-HACKATHON
sanity
schemaTypes
categories.ts
index.ts
product.ts
env.ts
structure.ts
scripts
cleanup.mjs
migrate.mjs
.env
.env.local
.eslintrc.json
.gitignore
components.json U
next-env.d.ts
next.config.mjs
package-lock.json M
OUTLINE
product.ts
sanity > schemaTypes > product.ts > productSchema > fields
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",
},
{


File Edit Selection View Go Run Terminal Help
categories.ts - 3rd-hackathon - Visual Studio Code
EXPLORER
OPEN EDITORS
page.tsx app\Cart M
model.tsx components U
categories.ts sanity\schema...
Header.tsx compone... M
page.tsx app\categor... U
Shipmentform.tsx c... U
page.tsx app\product
3RD-HACKATHON
sanity
schemaTypes
categories.ts
index.ts
product.ts
env.ts
structure.ts
scripts
cleanup.mjs
migrate.mjs
.env
.env.local
.eslintrc.json
OUTLINE
categories.ts
Header.tsx M
page.tsx ...\[id] U
Shipmentform.tsx U
sanity > schemaTypes > categories.ts > categorySchema > fields
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',
}
],

All products