

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:

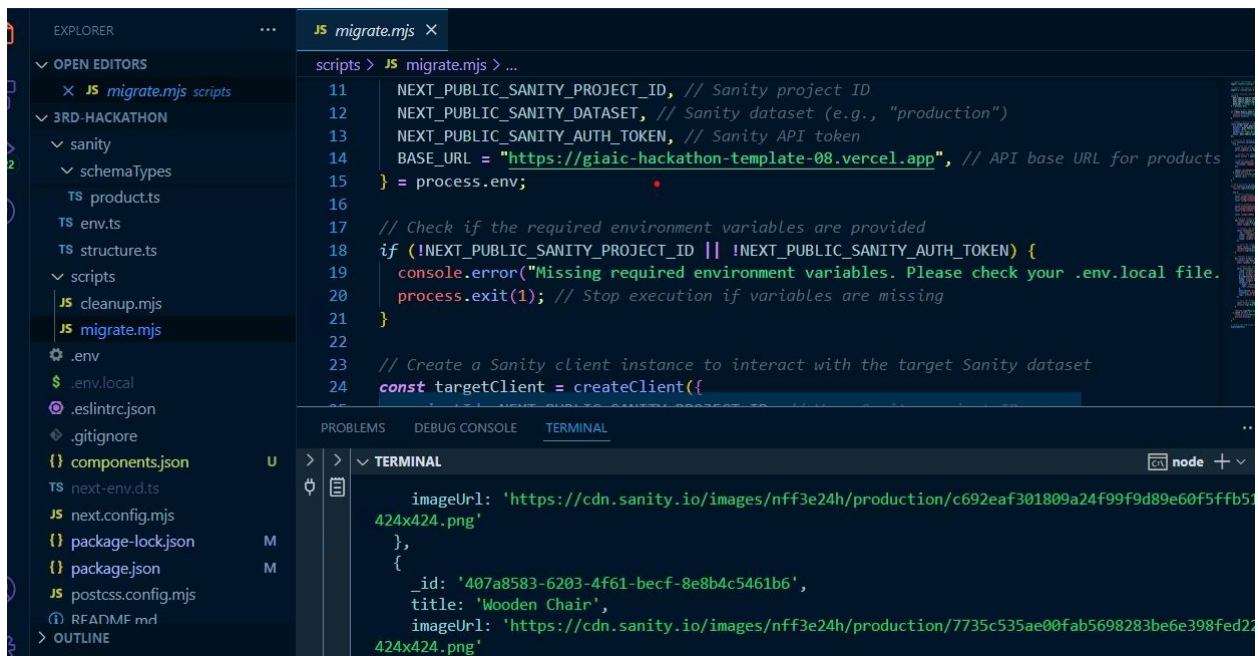
- Deploy a customized migration script to transfer data seamlessly from the legacy system or CSV files into Sanity CMS.

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.




The screenshot displays a Visual Studio Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with folders like '3RD-HACKATHON' and 'sanity', and files like 'migrate.mjs'. The main editor window shows the content of 'migrate.mjs', which is a JavaScript script for migrating data to Sanity. The script includes comments for environment variables and a check for their presence. The terminal at the bottom shows the output of the script, displaying a JSON object for a product named 'Wooden Chair' with its image URL and ID.

```
scripts > JS migrate.mjs > ...
11 NEXT_PUBLIC_SANITY_PROJECT_ID, // Sanity project ID
12 NEXT_PUBLIC_SANITY_DATASET, // Sanity dataset (e.g., "production")
13 NEXT_PUBLIC_SANITY_AUTH_TOKEN, // Sanity API token
14 BASE_URL = "https://giaic-hackathon-template-08.vercel.app", // API base URL for products
15 } = process.env;
16
17 // Check if the required environment variables are provided
18 if (!NEXT_PUBLIC_SANITY_PROJECT_ID || !NEXT_PUBLIC_SANITY_AUTH_TOKEN) {
19   console.error("Missing required environment variables. Please check your .env.local file.");
20   process.exit(1); // Stop execution if variables are missing
21 }
22
23 // Create a Sanity client instance to interact with the target Sanity dataset
24 const targetClient = createClient({
25   projectId: NEXT_PUBLIC_SANITY_PROJECT_ID,
26   dataset: NEXT_PUBLIC_SANITY_DATASET,
27   token: NEXT_PUBLIC_SANITY_AUTH_TOKEN,
28   url: BASE_URL
29 });
30
31 // Example: Import a single product from a CSV row
32 const csvRow = {
33   title: 'Wooden Chair',
34   image: 'https://cdn.sanity.io/images/nff3e24h/production/c692eaf301809a24f99f9d89e60f5ffb51424x424.png',
35   id: '407a8583-6203-4f61-becf-8e8b4c5461b6',
36   imageUrl: 'https://cdn.sanity.io/images/nff3e24h/production/7735c535ae0fab5698283be6e398fed22424x424.png'
37 };
38
39 // Import the product into the Sanity dataset
40 targetClient.create(csvRow);
```

node + v

```
imageUrl: 'https://cdn.sanity.io/images/nff3e24h/production/c692eaf301809a24f99f9d89e60f5ffb51424x424.png'
},
{
  _id: '407a8583-6203-4f61-becf-8e8b4c5461b6',
  title: 'Wooden Chair',
  imageUrl: 'https://cdn.sanity.io/images/nff3e24h/production/7735c535ae0fab5698283be6e398fed22424x424.png'
```

 **Comforty**

Select framework...

Cart 2

HomeShopProductPagesAbout

Contact: (808) 555-0111

SDefault+ Create

StructureVisionSchedules

Content

ProductsCategories

Categories

Search list

Desk Chair

Wooden Chair

Wing Chair

Wooden Chair


Categories

Wooden Chair

Category Title

Wooden Chair


Category Image



Marketplace-Hackathon/Docs/ xCreate Next App xGray Elegance | Default x

http://localhost:3000/studio/structure/products:aWt125CyehY2YfnCtyCyUy

Verify it's youFinish update

 **Comforty**

Select framework...

Cart 2

HomeShopProductPagesAbout

Contact: (808) 555-0111

SDefault+ Create

StructureVisionSchedules

Content

ProductsCategories

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

product.ts - 3rd-hackathon - Visual Studio Code

EXPLORER

OPEN EDITORS

- TS product.ts

3RD-HACKATHON

- sanity
 - schemaTypes
 - TS product.ts

product.ts

```
1
2
3 import { defineType } from "sanity";
4
5 export const productSchema = defineType({
6   name: "products",
7   title: "Products",
8   type: "document",
9   fields: [
10     {
11       name: "title",
12       title: "Product Title",
13       type: "string",
14     },
15     {
16       name: "price",
17       title: "Price",
18       type: "number",
19     },
20     {
21       title: "Price without Discount",
22       name: "priceWithoutDiscount",
23       type: "number",
24     },
25   ],
26 });
```

categories.ts - 3rd-hackathon - Visual Studio Code

EXPLORER

OPEN EDITORS

- page.tsx app\Cart M
- model.tsx components U
- TS categories.ts
- Header.tsx compone... M
- page.tsx app\categor... U
- Shipmentform.tsx c... U
- page.tsx app\product

3RD-HACKATHON

- sanity
 - schemaTypes
 - TS categories.ts

categories.ts

```
1
2
3 import { defineType } from "sanity";
4
5 export const categorySchema = defineType({
6   name: 'categories',
7   title: 'Categories',
8   type: 'document',
9   fields: [
10     {
11       name: 'title',
12       title: 'Category Title',
13       type: 'string',
14     },
15     {
16       name: 'image',
17       title: 'Category Image',
18       type: 'image',
19     },
20     {
21       title: 'Number of Products',
22       name: 'products',
23       type: 'number',
24     },
25   ],
26 });
```

All products

