

Day 3 (API Integration and Data Migration) of E-commerce Marketplace Builder Hackathon

Project: Modern Haven Furniture Ecommerce Marketplace

Prepared by: Muhammad Ghufraan (00121430)

Date: January 18, 2025

Introduction

This document details the successful completion of Day 3 tasks for the Modern Haven Furniture Marketplace project during the hackathon. The tasks included API integration, data migration to Sanity CMS, schema validation and customization, and frontend data rendering using Next.js. Each section highlights the steps taken, methods used, and best practices followed to ensure seamless functionality.

Completed Tasks Overview

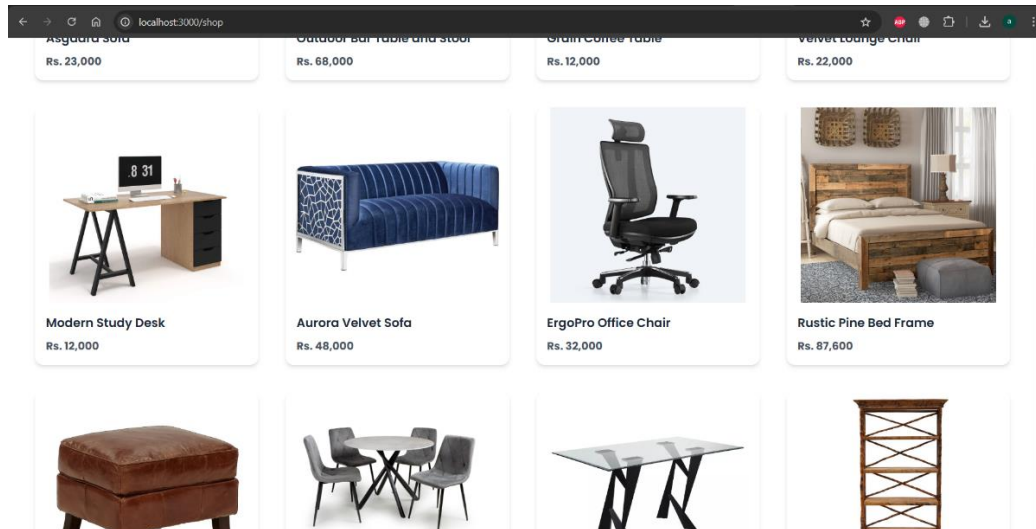
1. API Integration with Next.js

- Implemented API calls to fetch product data from the provided API.
- Created utility functions to handle API requests efficiently.
- Integrated error handling for API failures and displayed user-friendly messages.

Code Snippet: Function for fetching data from API.

```
const data: Product[] = await client.fetch(`
  *[_type == "product"]{
    name,
    imagePath,
    price,
    id,
    "slug": slug.current
  }
`);
```

Screenshot: API response successfully logged in the console.



2. Sanity CMS Data Migration

- I analyzed and understood the API structure provided.
- Adjusted Sanity schemas to align with API data.
- Used migration scripts to import product data into Sanity CMS.

Code Snippet: Custom migration script used for data import.

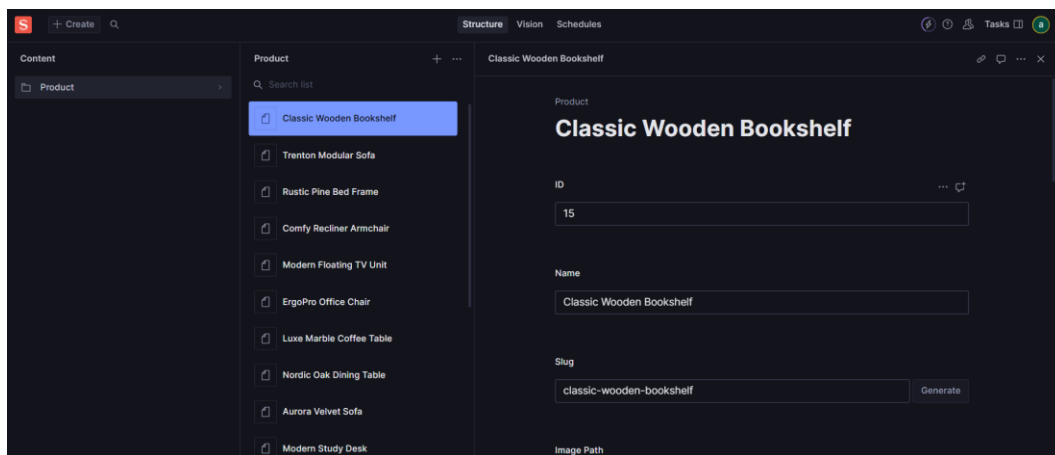
```
import axios from 'axios';
import { client } from '@sanity/lib/client';
type Product = {
  id: string;
  name: string;
  imagePath: string;
  price: number;
  description: string;
  colors: string[];
  reviews: number;
  discountPercentage: number;
  isFeaturedProduct: boolean;
  stock_quantity: number;
  category: string;
};
export async function GET() {
  try {
    const { data }: { data: Product[] } = await
    axios.get('https://677fc37f0476123f76a7fbbb.mockapi.io/product');
    for (const product of data) {
```

```

await client.create({
  _type: 'product',
  id: product.id,
  name: product.name,
  imagePath: product.imagePath,
  price: product.price,
  description: product.description,
  colors: product.colors,
  reviews: product.reviews,
  discountPercentage: product.discountPercentage,
  isFeaturedProduct: product.isFeaturedProduct,
  stock_quantity: product.stock_quantity,
  category: product.category,
});
}
return new Response(JSON.stringify({ message: 'Data inserted successfully!' }), {
  status: 200,
  headers: { 'Content-Type': 'application/json' },
});
} catch (error: unknown) {
  console.error('Error inserting data:', error);
  return new Response(JSON.stringify({ error: 'Failed to fetch or insert data' }), {
    status: 500,
    headers: { 'Content-Type': 'application/json' },
  });
}
}
}

```

Screenshot: Populated Sanity CMS with migrated data.



3. Schema Validation and Customization

- Customized the product schema in Sanity CMS to include:
 - **Colors** (Array)

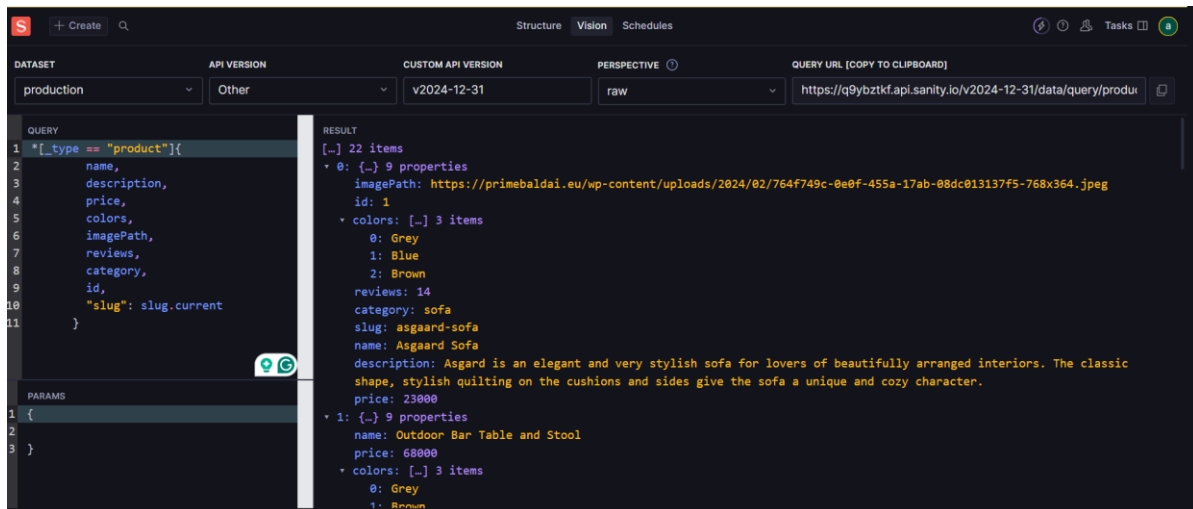
- **Reviews** (number)

Code Snippet: Product schema code in Sanity CMS.

```
export default {
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    {
      name: "id",
      title: "ID",
      type: "string",
    },
    {
      name: "name",
      title: "Name",
      type: "string",
    },
    {
      name: "slug",
      title: "Slug",
      type: "slug",
      options: {
        source: "name",
        maxLength: 200,
      },
    },
    {
      name: 'imagePath',
      title: 'Image Path',
      type: 'url',
    },
    {
      name: "price",
      title: "Price",
      type: "number",
    },
    {
      name: "description",
      title: "Description",
      type: "text",
    },
    {
      name: "colors",
      title: "Colors",
    },
  ],
}
```

```
    type: "array",
    of: [{ type: "string" }],
  },
  {
    name: "reviews",
    title: "Reviews",
    type: "number",
  },
  {
    name: 'discountPercentage',
    title: 'Discount Percentage',
    type: 'number',
  },
  {
    name: 'isFeaturedProduct',
    title: 'Is Featured Product',
    type: 'boolean',
  },
  {
    name: "stock_quantity",
    title: "Stock Quantity",
    type: "number",
  },
  {
    name: "category",
    title: "Category",
    type: "string",
  },
],
};
```

Screenshot: Sanity CMS schema structure.



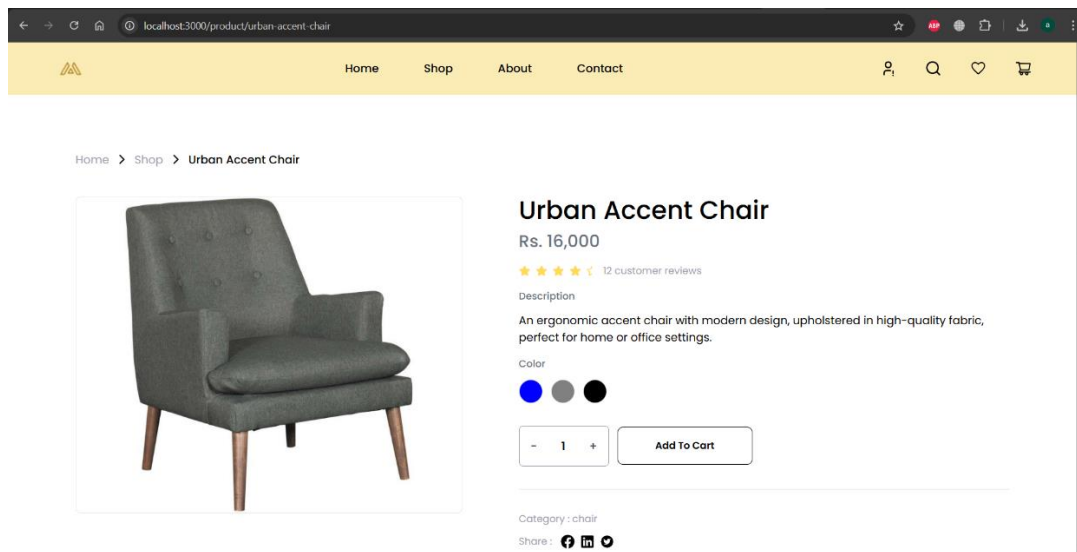
4. Frontend Data Rendering in Next.js

- Fetched data using GROQ.
- Rendered product listings dynamically with responsive design.
- Implemented dynamic routing for product detail pages.

Code Snippet: GROQ query.

```
*[_type == "product" && slug.current == $slug][0]{
  name,
  description,
  price,
  colors,
  imagePath,
  reviews,
  category,
  id,
  "slug": slug.current
}
```

Screenshot: Product listing page displaying data fetched from Sanity.



5. Error Handling and Optimization

- Integrated comprehensive error handling in API calls.
- Used skeleton loaders to improve user experience during data fetching.
- Optimized images using next/image for performance.

Best Practices Followed

- **Environment Variables:** Stored sensitive data like API keys in .env files.
- **Modular Code:** Separated concerns using utility functions and reusable components.
- **Data Validation:** Ensured schema alignment before and after data migration.

Conclusion

All Day 3 tasks for API integration and data migration have been successfully completed. The Modern Haven Furniture Marketplace now features a functional backend powered by Sanity CMS, seamless API integration, and a responsive frontend displaying dynamic product data.