Day 3 - API Integration Report: LuxeWalk

Overview

On **Day 3**, the focus was on integrating APIs into the LuxeWalk project and populating Sanity CMS with data sourced from a local API. This report documents the API integration process, schema adjustments, migration steps, and the tools used. Screenshots and code snippets are included to provide a comprehensive understanding of the implementation.

API Integration Process

1. Data Source

Data was fetched from the local API endpoint: `http://localhost:3000/api/products`. The product data included fields such as name, description, price, images, stock level, and more.

2. Integration Steps

- **Schema Design:** Created a custom schema for products in Sanity CMS to align with the structure of the imported data.
- **Import Script:** Developed a migration script to fetch product data from the API, process it, and upload it to Sanity CMS.
- **Image Handling:** Enhanced the script to fetch images as an array and upload them to Sanity CMS with unique references.
- **Data Validation:** Incorporated validation rules for fields like slug, price, and reviews to ensure data consistency.
- **API Endpoints:** Created endpoints to retrieve data from Sanity CMS for use in the frontend.

Schema Adjustments

Product Schema

The schema was customized to accommodate data fields such as `tags`, `colors`, and `images`. Key validation rules and slug uniqueness checks were implemented.

[Schema Source 🔗]

```
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    { name: 'name', title: 'Product Name', type: 'string', validation:
(Rule) => Rule.required() },
    { name: 'slug', title: 'Slug', type: 'slug', options: { source:
'name', maxLength: 96 }, validation: (Rule) => Rule.required() },
    { name: 'description', title: 'Description', type: 'text', validation:
(Rule) => Rule.max(500) },
    { name: 'price', title: 'Price', type: 'number', validation: (Rule) =>
Rule.required().positive().precision(2) },
    { name: 'images', title: 'Product Images', type: 'array', of: [{ type:
'image', options: { hotspot: true } }] },
    // Additional fields omitted for brevity
  ],
};
```

Migration Steps

1. Tools Used

- **Sanity Client:** For uploading data to Sanity CMS.
- **Axios:** For API calls to fetch product data.
- **UUID:** For generating unique keys for images.
- **.env:** For managing environment variables.

2. Migration Script

The script automated the process of fetching data, processing it, and importing it into Sanity CMS while handling images as arrays.

[Script Source 🔗]

```
import { createClient } from '@sanity/client';
import axios from 'axios';
```

```
import { v4 as uuidv4 } from 'uuid';

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: 'production',
  token: process.env.SANITY_API_TOKEN,
  useCdn: false,
});

async function importData() {
  const response = await axios.get('http://localhost:3000/api/products');
  const products = response.data;

  for (const product of products) {
    const imageRefs = await Promise.all(
      product.imageUrl.map(async (url) => {
        const response = await axios.get(`http://localhost:3000${url}`, {
responseType: 'arraybuffer' });
        const asset = await client.assets.upload('image',
Buffer.from(response.data), { filename: url.split('/').pop() });
        return { asset: { _ref: asset._id }, _key: uuidv4() };
      })
    );

    const sanityProduct = {
      _type: 'product',
      name: product.name,
      price: product.price,
      images: imageRefs,
      // Additional fields
    };

    await client.create(sanityProduct);
  }
}

importData();
```

API Calls

1. Fetching All Products

[Query Source 🔗]

```
export const fetchProducts = async () => {
  const query = `
    *[_type == "product"]{
      "id": _id,
      name,
      price,
      "images": images[].asset->url
    }
  `;
  return await client.fetch(query);
};
```

2. Endpoint for Product by Slug

[Endpoint Source 🔗]


```
export async function GET(request) {
  const { slug } = new URL(request.url).searchParams;
  const query = `*[_type == "product" && slug.current == $slug][0]`;
  const product = await client.fetch(query, { slug });
  return new Response(JSON.stringify(product), { status: 200 });
}
```

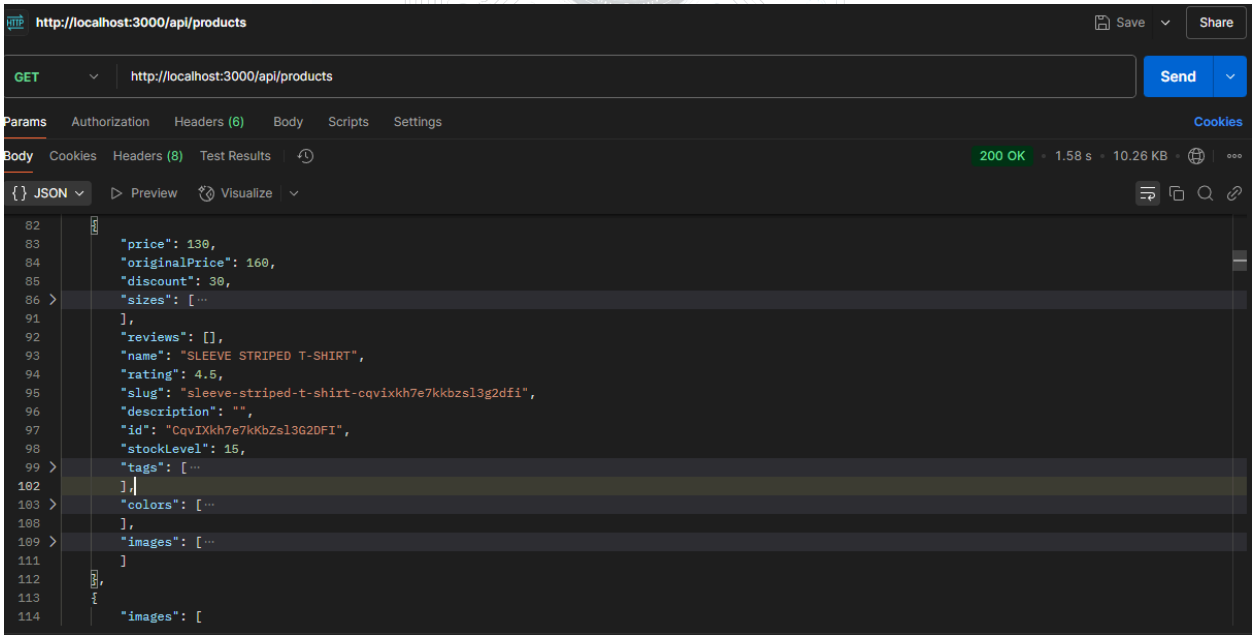
Screenshots

1. **Data Import**

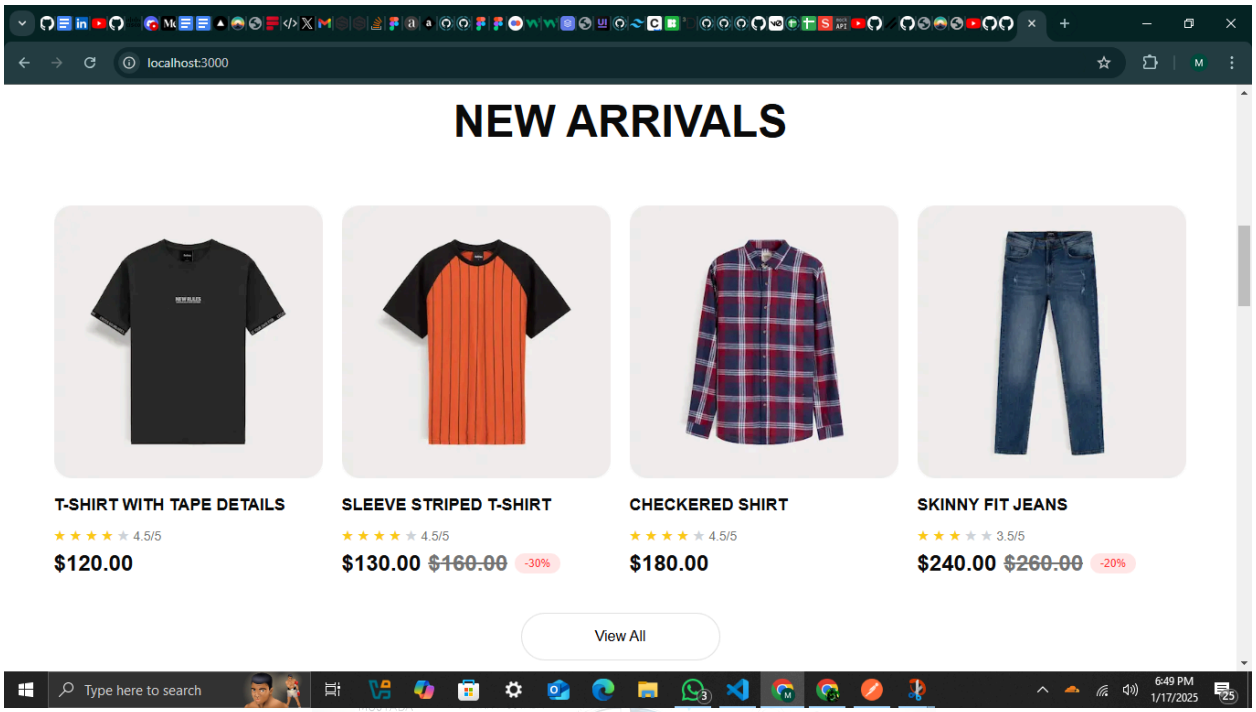
```
Product uploaded successfully: PUo9bqxkDx0BivYMzT5qDA
Processing product: COURAGE GRAPHIC T-SHIRT
Uploading product to Sanity: COURAGE GRAPHIC T-SHIRT
Product uploaded successfully: PUo9bqxkDx0BivYMzT5qct
Processing product: SKINNY FIT JEANS
Uploading product to Sanity: SKINNY FIT JEANS
Product uploaded successfully: PUo9bqxkDx0BivYMzT5r0H
Processing product: VERTICAL STRIPED SHIRT
Uploading product to Sanity: VERTICAL STRIPED SHIRT
Product uploaded successfully: PUo9bqxkDx0BivYMzT5rGe
Processing product: LOOSE FIT BERMUDA SHORTS
Uploading product to Sanity: LOOSE FIT BERMUDA SHORTS
Product uploaded successfully: PUo9bqxkDx0BivYMzT5wOi
Processing product: Polo with Tipping Details
Uploading product to Sanity: Polo with Tipping Details
Product uploaded successfully: PUo9bqxkDx0BivYMzT5wck
Processing product: CHECKERED SHIRT
Uploading product to Sanity: CHECKERED SHIRT
Product uploaded successfully: hK4GXecj6BqL3u8A1EDnRu
Processing product: SLEEVE STRIPED T-SHIRT
Uploading product to Sanity: SLEEVE STRIPED T-SHIRT
Product uploaded successfully: 1pDAoKxEuiH06lq4S7063j
Processing product: VERTICAL STRIPED SHIRT
Uploading product to Sanity: VERTICAL STRIPED SHIRT
Product uploaded successfully: 1pDAoKxEuiH06lq4S706aY
Processing product: LOOSE FIT BERMUDA SHORTS
Uploading product to Sanity: LOOSE FIT BERMUDA SHORTS
Product uploaded successfully: 1pDAoKxEuiH06lq4S706sS
Data import completed successfully!

E:\mujtaba data\coding classes\proramming\my code\GitHub_Repo_Codes\dec-hackathon>
```

2. API Call Response



3. **Frontend Display of Data**



Sign up and get 20% off to your first order. Sign Up Now

SHOP.CO Shop On Sale New Arrivals Brands Search for products...

Home > Shop > New Arrival

T-SHIRT WITH TAPE DETAILS

★★★★☆ 4.5/5

$120

Select Colors

Choose Size

S M L XL

– 1 + Add to Cart

Product Details | Rating & Reviews | FAQs

Product Details

Detailed product information will be displayed here.

YOU MIGHT ALSO LIKE

T-SHIRT WITH TAPE DETAILS

★★★★☆ 4.5/5

$120.00

SLEEVE STRIPED T-SHIRT

★★★★☆ 4.5/5

$130.00 ~~$160.00~~ -30%

COURAGE GRAPHIC T-SHIRT

★★★★☆ 4/5

$145.00

Conclusion

This process established a seamless pipeline to import, validate, and display product data from a local API into Sanity CMS. By customizing schemas and scripts, the data was efficiently handled to meet project requirements. The next steps include further optimizing queries and implementing advanced features like search and filtering.