

Day 3 - API Integration Report - FurniSphere

API Creation Process:

- Created my own API using route handlers in Next js.
- Uses GET function for fetching products.
- Created Endpoint **/products** for product listings.
- Tested API Endpoint using Postman to validate API response.
- Created API: <https://custom-api-nu.vercel.app/api/products>

API Integration Process:

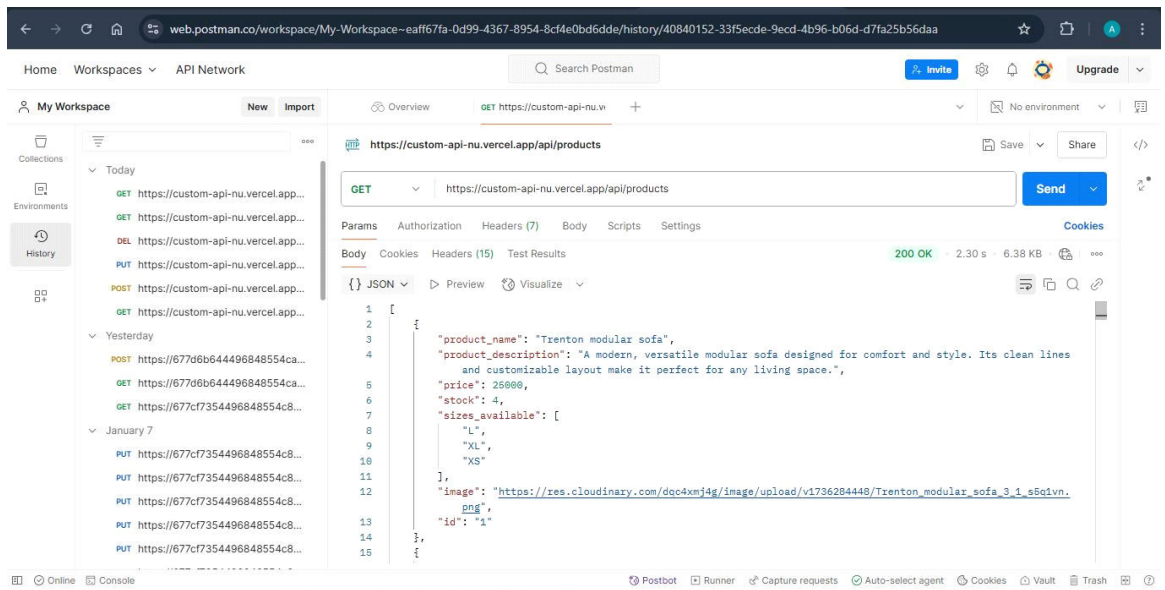
- Integrated my custom API for fetching product data into the Next.js frontend.
- Created Migration script to migrate data into Sanity CMS.
- Migrated fetched data into Sanity CMS for backend management.

Migration Steps and Tools Used:

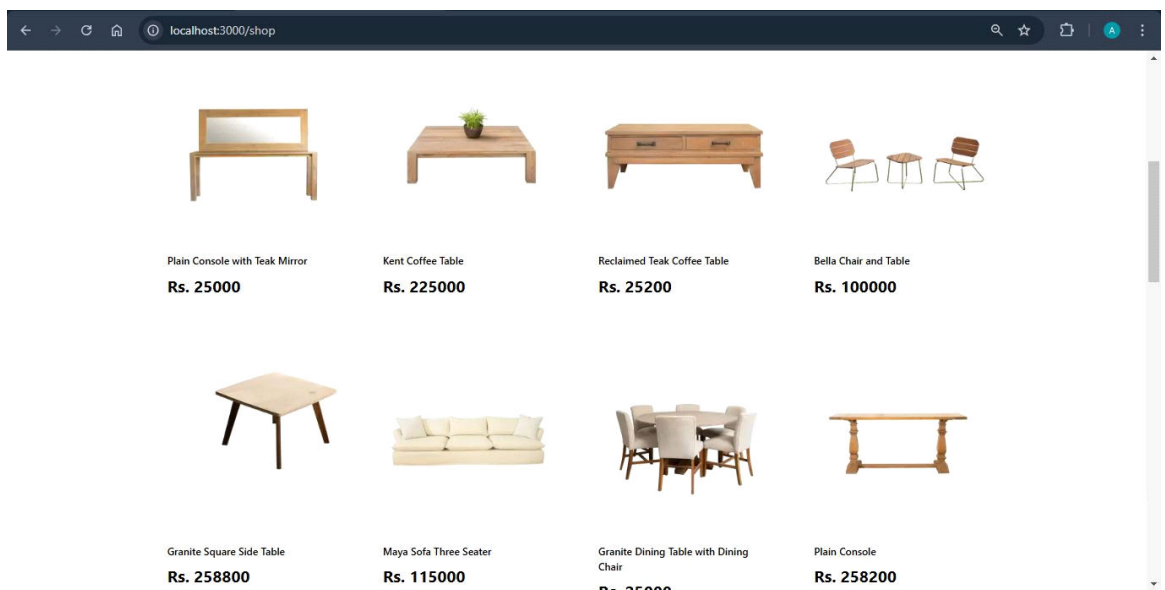
- A script was written to fetch and import data into Sanity CMS.
- **Postman**: For testing API endpoints.
- **Sanity Studio**: For schema management and data validation.
- **VS Code**: For coding the integration and migration scripts.

Screenshots:

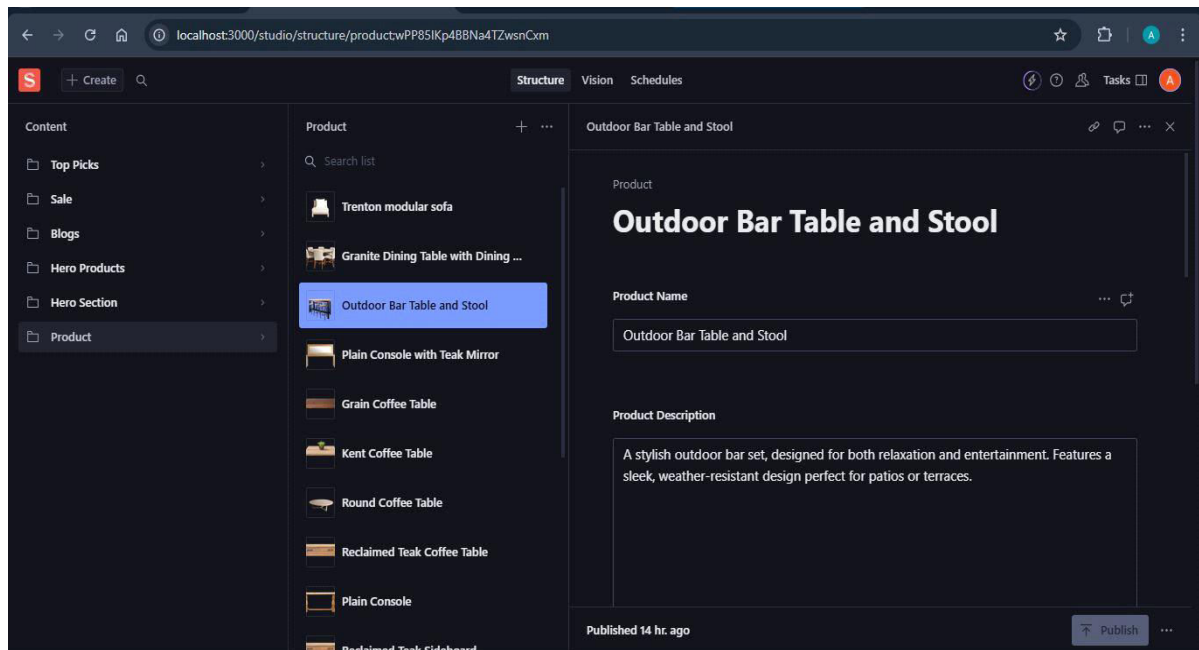
1. API Call:



2. Data Displayed on Frontend:



3. Sanity CMS Fields:



Code Snippets:

- **Migration Script:**

Integrated API through the migration script into the Next.js frontend and Sanity CMS.

```

import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

//! Load environment variables
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

//! Sanity client
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2021-08-31',
});

//! upload image to sanity

const uploadImage = async (imageUrl) => {
  try {
    const response = await axios.get(imageUrl, { responseType: 'stream' });
    const uploadedAsset = await client.assets.upload('image', response.data, {
      filename: imageUrl.split('/').pop(),
    });

    console.log('Image uploaded successfully:', uploadedAsset._id);
    return uploadedAsset._id;
  } catch (err) {
    console.error('Failed to upload image:', imageUrl, err.message);
    throw err;
  }
};

//! migration function

const importData = async () => {
  try {
    console.log('Fetching products from API...');
    const response = await axios.get('https://custom-api-nu.vercel.app/api/products');
    const products = response.data;

    console.log(`Fetched ${products.length} products`);

    for (const product of products) {
      try {
        console.log(`Processing product: ${product.product_name}`);
        const imageRef = await uploadImage(product.image);

        const sanityProduct = {
          _type: 'product',
          product_name: product.product_name,
          product_description: product.product_description,
          price: product.price,
          stock: product.stock,
          sizes_available: product.sizes_available,
          image: {
            _type: 'image',
            asset: {
              _type: 'reference',
              _ref: imageRef,
            },
          },
        };

        console.log('Uploading product to Sanity:', sanityProduct.product_name);
        const result = await client.create(sanityProduct);
        console.log(`Product uploaded successfully: ${result._id}`);
      } catch (error) {
        console.error(`Skipping product ${product.product_name} due to error:`, error.message);
      }
    }

    console.log('Data import completed successfully!');
  } catch (error) {
    console.error('Error importing data:', error.message);
  }
};

importData();

```

Conclusion:

- Created API in Next js.
- Tested API through Postman to validate response.
- Successfully integrated API data through the migration script into the Next.js frontend and Sanity CMS.
- Uses .env file that contains API keys and sensitive information securely.
- Successfully render data on frontend.
- Stored sensitive information and API keys in **.env** file to ensure security.