

Day 3 Progress & Challenges

On Day 3, I concentrated on data visualization before moving on to develop the homepage and its components. My next step was integrating dynamic data.

Challenges Encountered

A major hurdle was the mismatch between the furniture template I initially selected and the requirements of a Furniture marketplace. Due to the limited timeframe, I made the decision to adapt the project to better align with the available data. Currently, the dynamic product detail page is experiencing issues with ID and slug handling, preventing it from functioning correctly. I plan to resolve this issue as soon as possible.

Lessons Learned

For future projects, I've learned the importance of choosing a template that's a good fit for both the marketplace concept and the data from the start. This will lead to a smoother development process and avoid potential problems.

I invested a significant amount of time in this project, including working through the night. While some components were adapted from existing resources to expedite development, this still required considerable effort. I recognize areas where my approach could have been better, and I apologize for any shortcomings. I'm committed to demonstrating substantial progress in future hackathons if given the opportunity. Thank you for your understanding.s

Migration Script

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

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

const client = createClient({
  projectId: "cvghk2d8",
  dataset: "production",

  token: "skchqFXPC0Ddgx0P2P8BL1MkyL5HBNX9ULR9KMeRlBq8tqSon0nUnzmJ9XJSRV2RdhhecWtFkdaQIBuyZ9jTAcvFcKq8Ayliob3H7AwzG5dh8WWzTA1Hhc6TUQvn0Axfc0iW3wsuV3dRII1AeToXv7d6VrFxCwIDuaoet3cuzdkQD9lZ5bEJ",
  apiVersion: '2025-01-15',
  useCdn: false,
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading Image : ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop(),
    });
    console.log(`Image Uploaded Successfully : ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to Upload Image:', imageUrl, error);
  }
}
```

```

    return null;
  }
}

async function importData() {
  try {
    console.log('Fetching Product Data From API ...');

    const response = await axios.get("https://next-e-commerce-
template-4.vercel.app/api/product")
    const products = response.data.products;

    for (const item of products) {
      console.log(`Processing Item: ${item.name}`);

      let imageRef = null;
      if (item.imagePath) {
        imageRef = await uploadImageToSanity(item.imagePath);
      }

      const sanityItem = {
        _type: 'product',
        name: item.name,
        category: item.category || null,
        price: item.price,
        description: item.description || '',
        discountPercentage: item.discountPercentage || 0,
        stockLevel: item.stockLevel || 0,
        isFeaturedProduct: item.isFeaturedProduct,
        image: imageRef
        ? {
            _type: 'image',
            asset: {
              _type: 'reference',
              _ref: imageRef,
            },
          }
        : undefined,
      };

      console.log(`Uploading ${sanityItem.category} - ${sanityItem.name} to
Sanity !`);
      const result = await client.create(sanityItem);
      console.log(`Uploaded Successfully: ${result._id}`);
      console.log("-----")
      console.log("\n\n")
    }

    console.log('Data Import Completed Successfully !');
  } catch (error) {
    console.error('Error Importing Data : ', error);
  }
}

```

importData();

Schemas

```

export default {
  name: "product",
  type: "document",
  title: "Product",
  fields: [
    {

```

```

    name : "id",
    type : "string",
    title : "id"
  },
  {
    name: "name",
    type: "string",
    title: "Name",
    validation: (Rule: any) => Rule.required().error("Name is required"),
  },
  {
    name: "slug",
    type: "slug",
    title: "slug",
    options: {
      source: "name",
    },
  },
  {
    name: "image",
    type: "image",
    title: "Image",
    options: {
      hotspot: true,
    },
    description: "Upload an image of the product.",
  },
  {
    name: "price",
    type: "number",
    title: "Price",
    validation: (Rule: any) => Rule.required().error("Price is required"),
  },
  {
    name: "description",
    type: "text",
    title: "Description",
    validation: (Rule: any) =>
      Rule.max(150).warning("Keep the description under 150 characters."),
  },
  {
    name: "discountPercentage",
    type: "number",
    title: "Discount Percentage",
    validation: (Rule: any) =>
      Rule.min(0).max(100).warning("Discount must be between 0 and 100."),
  },
  {
    name: "isFeaturedProduct",
    type: "boolean",
    title: "Is Featured Product",
  },
  {
    name: "inventory",
    type: "number",
    title: "inventory",
    validation: (Rule: any) =>
      Rule.min(0).error("Stock level must be a positive number."),
  },
  {
    name: "category",
    type: "string",
    title: "Category",
    options: {

```

```
list: [
  { title: "Chair", value: "Chair" },
  { title: "Sofa", value: "Sofa" },
],
},
validation: (Rule: any) => Rule.required().error("Category is required"),
],
};
```