Steps Taken

On Day 3, I focused on data visualization and then proceeded to develop the homepage and various components. Following this, I worked on integrating dynamic data.

Challenges I Faced

A significant challenge arose from the mismatch between the chosen furniture template and the intended clothing marketplace. Due to time constraints, I pivoted to align with the existing data. Currently, the dynamic product detail page is not functioning correctly due to unresolved ID and slug issues, which I plan to address promptly.

Best Practices

Ideally, selecting a template that aligns with both the marketplace concept and the available data from the outset is crucial. This approach streamlines development and minimizes potential confusion.

I dedicated considerable time to this project, including working through the night. I acknowledge that some components were adapted from existing sources, which, while expedient, still required significant effort. I also recognize areas where my approach could have been improved, and I sincerely apologize for any shortcomings. If I have the opportunity to participate in future hackathons, I am committed to demonstrating substantial progress. Thank you for your understanding.

Code Snippets

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:"skchqFXPCODdgxOP2P8BL1MkyL5HBNX9ULR9KMeRlBq8tqSonOnUnzmJ9XJSRV2RdhhecWtFk
daQIBuyZ9jTacvFcKq8Ayliob3H7AWzG5dh8WWzTA1Hhc6TUQvnOAxfcOiW3wsuV3dRII1AeToXv7d6V
rFxCwIDuaoet3cuzdkQD9lZ5bEJ",
  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-ecommerce-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"),
        },
      ],
    };
```