

**Day 3 - API Integration Report**

**E-Commerce Website**

**Product: Pudding & Jelly Powders**

**Name: Umehani**

**Roll No: 00364678**

**Day/Time: Friday – 09:00 – 12:00**

**Teacher: Sir Hamzah Syed**

**Steps for Day 3:**

**1. Understand the Provided API:**

My Template was 8, and I applied Api for practice in another practice file, but for hackathon I select to add manual data for my product which I am focusing to set for public use in future. I manually upload data in sanity and fetch it to my project.

**2. Validate and Adjust Your Schema:**

Yes, I create my own schema’s for my products and set it for sanity. I compare my Sanity CMS schema (created on Day 2) with the current structured schema. My product schema contains: name, price, image, description of the product.

**3. Data Migration Options:**

I didn’t apply Api to fetch data so that I manually put data in sanity CMS and then fetch it successfully to my project. I add dynamic routing in my project so that my fetched data from sanity CMS become dynamic to other pages.

**Expected Output:**

**1. Sanity CMS populated with imported data from:**

* Manually uploaded data.

**2. Functional API integration in Next.js displaying:**

* No I didn’t use Api in my project for data.

**What to Submit:**

**A report documenting: API integration process:**

As I manually upload my product data, after Creating a new Next.js project by running this command:

* npx create-next-app .

Then I copy sanity command from panaverse/learn-nextjs, I Install Sanity CLI

* npm create sanity@latest -- --template clean --create-project "learning-sanity-project" --dataset production

After installing sanity I create Schemas for my products from Chatgpt

Then I open the Sanity Studio folder and create a file product.ts in sanity schema type as:

* sanity / schematype / product.ts

And copy the created schema in it:

import { defineField, defineType } from "sanity"

export default defineType({

name: 'pudding',

    title: 'pudding',

    type: 'document',

    fields: [

      defineField({

        name: 'name',

        title: 'name',

        type: 'string',

     }),

      defineField({

        name: 'description',

        title: 'description',

        type: 'text',

      }),

      defineField({

        name: 'slug',

        title: 'slug',

        type: 'slug',

        options: {

          source: "name",

          maxLength: 200

        }

      }),

      defineField({

        name: 'price',

        title: 'price',

        type: 'number',

      }),

      defineField({

        name: 'image',

        title: 'image',

        type: 'image',

        options: {

          hotspot: true,

        }

        }),

    ]

  })

**Start Sanity Studio**

Run this command to start the Studio locally:

* Npm run dev 🡪 [for localhost:3000]

Then open studio [localhost:3000/studio]

**Add Data through Sanity Studio**

* Navigate to the content type (e.g., "Product") in the Studio.
* Fill out the form fields and upload images.
* Click **Publish** to save your data

**Then write a GROQ Query**

\*[\_type == "pudding" && slug.current == $slug]{

          name,

          description,

          "slug":slug.current,

          price,

          image{

            asset->{

              \_id,

              url

      }

    },

  }

Fetch Data in React/Next.js Component

**Day 3 Checklist:**

* API Understanding:

 ✔ (Manually upload)

* Schema Validation:

 ✔

* Data Migration:

 ✔ (Fetching)

* API Integration in Next.js:

 ✔ (other Api’s)

* Submission Preparation:

 ✔