Hackathon Day 3 - API Integration & Data Migration

Introduction

This report outlines the tasks completed on Hackathon Day 3, including API integration, data migration, and schema setup in Sanity.

Sanity API Integration

API was used for fetching and storing structured content. The integration included setting up the Sanity client, authentication, and defining API endpoints.

```
[("name":"The Poplar suede sofa", "description":"A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather products of the product of the produc
```

Schema Design

Multiple schemas were created for different data types. Each schema contained defined fields, validation rules, and relationships where necessary.

Schema Example 1

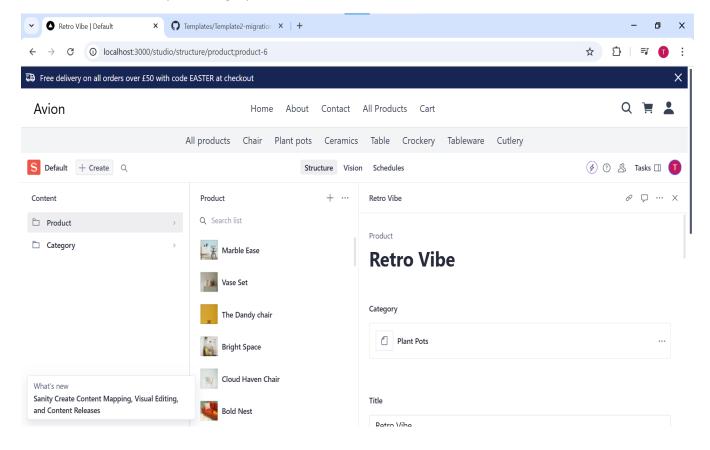
```
hackathone > src > sanity > schemaTypes > rs product.ts > 🔎 product > 🔑 fields
  import { defineType, defineField } from "sanity"
  export const product = defineType({
      name: "product",
      title: "Product",
      type: "document",
      fields: [
          defineField({
              name: "category",
              title: "Category",
               type: "reference",
                   type:"category"
          defineField({
              name: "name",
title: "Title",
              validation: (rule) => rule.required(),
              type: "string"
          defineField({
              name: "slug",
               title: "Slug",
               type: "slug",
               options:{
                   source: "name"
```

Schema Example 1

```
import { defineType,defineField } from "sanity";
export const Category = defineType({
   name: "category",
title: "Category",
    type: "document",
    fields:[
        defineField({
           name: "name",
            title: "Name",
            type: "string",
            validation: (rule) => rule.required(),
        defineField({
           name: "slug",
title: "Slug",
            type: "slug",
            validation: (rule) => rule.required(),
            options: {
                source: "name",
```

Data Migration & Imports

Data was migrated and imported into Sanity using structured JSON files. The import/export process ensured data consistency and integrity.



Challenges & Solutions

Several challenges were faced during integration, including schema validation errors and API rate limits.

These issues were resolved by adjusting the schema structure and optimizing API requests.