API Integration and Data Migration Report

Day: 3 - API Integration and Data Migration

API Integration Process

1. Setup and Configuration:

- Installed required dependencies like:
    - npm install
    - npm install dotenv etc

- Configured the Sanity client with project-specific credentials in the environment file.

2. Data Fetching:

- Utilized GROQ queries to fetch data from Sanity CMS.
- Implemented dynamic API routes in Next.js to interact with Sanity endpoints.

3. Frontend Integration:

- Retrieved data using the configured API endpoints.
- Displayed fetched data in styled components on the frontend.

4. Error Handling:

- Implemented error handling mechanisms for API requests to ensure stability.
- Logged errors to the console for debugging during development.

Product Schema

Product Schema:

- Defined fields such as title, description, price, and image.
- Ensured all required fields are included to match frontend requirements.

Migration Steps and Tools Used

Export Data:

- Exported existing data from the source system in JSON format.

Sanity Import Tool:

- Utilized Sanity's CLI import tool to migrate data.
- Command: sanity dataset import <file.json> production.

Verification:

- Verified the integrity of imported data using GROQ queries.
- Ensured data fields matched the schema definitions.
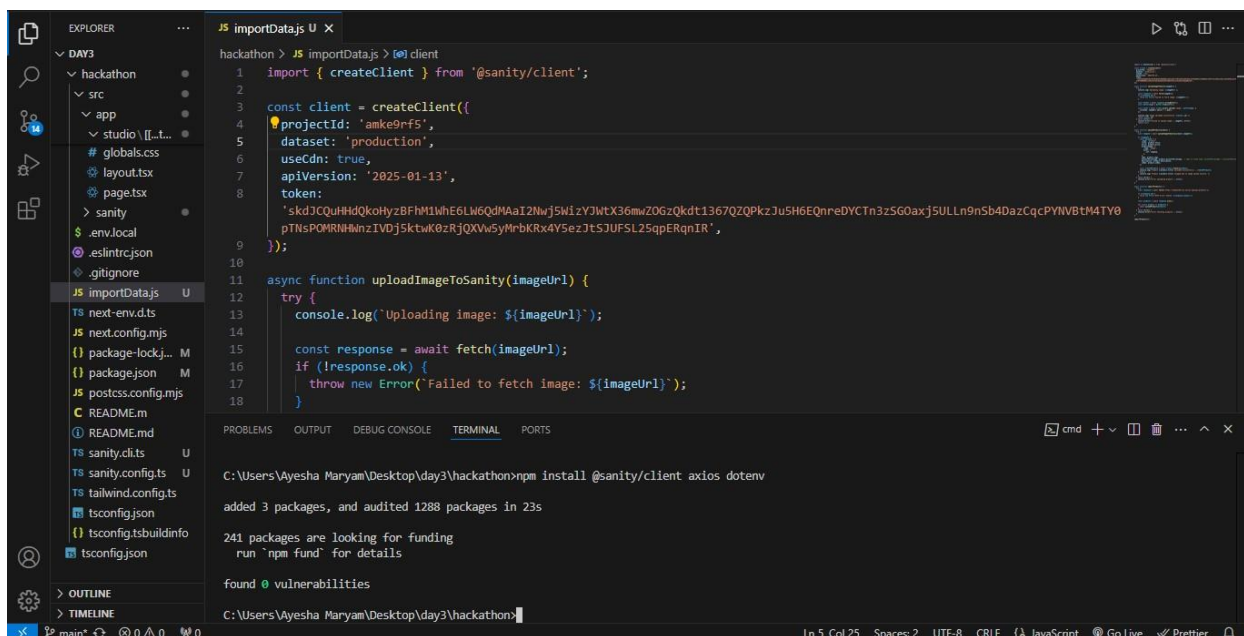
Frontend Updates:

- Updated components to ensure compatibility with migrated data.

Screenshots

- **API Calls:** Screenshots showcasing successful API requests and responses.
- **Frontend Display:** Visual representation of data rendered on the frontend.
- **Sanity CMS Fields**: Screenshots of populated fields within the Sanity Studio.

Code Snippets

```
EXPLORER                    JS importData.js U    {} package.json M    $ .env.local    TS sanity.cli.ts U    TS route.ts  U ×    TS index.d.mts
∨ DAY3                      hackathon > src > app > api > products > TS route.ts > ⊗ GET
∨ hackathon                 1    import { client } from "@/sanity/lib/client";
  > .next                   2    import { NextResponse } from "next/server";
  > node_modules            3
  > public                  4    export async function GET() {
  > script                  5      try {
  ∨ src                     6        const data = await client.fetch(`*[_type=="product"]{
    ∨ app                   7          _id,
      ∨ api\products        8          title,
        TS route.ts    U    9          "imageUrl" :productImage.asset -> url,
      > components          10         price,
      > fonts              11         tags,
      > pages              12         dicountPercentage,
      ∨ studio\[[...t...    13         description,
        ⊕ page.tsx     U    14         isNew
      # globals.css         15       }`);
      ⊕ layout.tsx          16
      ⊕ page.tsx            17       return  NextResponse.json(data, { status: 200 });
      ∨ sanity              18     } catch (error) {
        > lib               19       console.error('Error fetching data from Sanity:', error);
        ∨ schemaTypes       20       return new NextResponse('Error fetching data', { status: 500 });
          TS index.ts   U   21     }
          TS product.ts U        }
        TS env.ts       U
        TS structure.ts U
      $ .env.local
      ⊕ .eslintrc.json
> OUTLINE
> TIMELINE
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

> sanity@0.1.0 dev
> next dev

▲ Next.js 14.2.23
- Local:        http://localhost:3000
- Environments: .env.local

✓ Starting...
✓ Ready in 9.5s
```

```
∨ hackathon          1    {
  ∨ src              2      "name": "sanity",
    ∨ app            3      "type": "module",
      ∨ studio\[[...tool]]
         page.tsx    4      "version": "0.1.0",
      # globals.css  5      "private": true,
      ⊕ layout.tsx
      ⊕ page.tsx     6      "scripts": {
      ∨ sanity       7        "dev": "next dev",
        ∨ lib        8        "build": "next build",
          TS client.ts  9     "start": "next start",
          TS image.ts  10     "lint": "next lint",
          TS live.ts   11     "importData": "node script/importData.js"
        ∨ schemaTypes 12    },
          TS index.ts  13    "dependencies": {
          TS product.ts 14     "@sanity/client": "^6.25.0",
        TS env.ts     15      "@sanity/image-url": "^1.1.0",
        TS structure.ts 16    "@sanity/vision": "^3.68.3",
      $ .env.local    17      "axios": "^1.7.9",
                      18      "dotenv": "^16.4.7",
                      19      "next": "^14.2.23",
                      20      "next-sanity": "^9.8.30",
```

S   Ayesha Maryam ⌄   hackathon ⌄   30 days left in trial   ⓘ   A

⚡ Getting started   ⊞ Overview   ⚙ Members   ⊡ Studios   ⊟ Datasets   ⚿ Access   ⋀ Activity   ⊵ Usage   ⚡ Plan   ⊗ API   ⚙ Settings

Webhooks   Tokens are used to authenticate apps and scripts to access project
                  data.
CORS origins
                  NAME   PERMISSIONS   CREATED
Tokens
                  sanity-project   Editor   2 hours   🗑

                  Copy the token below – this is your only chance to do so!

                  skdJCQuHHdQkoHyzBFhM1WhE6LW6QdMAaI2Nwj5WizYJWtX36mwZOGzQkdt1367QZQPkzJu5H6EQnreDYCTn3z
                  SGOaxj5ULLn9nSb4DazCqcPYNVBtM4TY0pTNsPOMRNHWnzIVDj5ktwK0zRjQXVw5yMrbKRx4Y5ezJtSJUFSL25
                  qpERqnIR

What's new
Sanity Create Content Mapping, Visual Editing,
and Content Releases

PREPARED BY AYESHA MARYAM

```
DAY3
v hackathon
  > node_modules
  > public
  v script
    JS importData.js    U
  v src
    v app
      v api\products
        TS route.ts      U
      > components
      > fonts
      > pages
      v studio\[[...t...
        page.tsx         U
      # globals.css
      layout.tsx
      page.tsx
  > sanity
  $ .env.local
  .eslintrc.json
  .gitignore
  TS next-env.d.ts
  JS next.config.mjs
  {} package-lock.j...  M
  {} package.json       M
  JS postcss.config.mjs
OUTLINE
```

```
            'Key Features:\n' +
            '\n' +
            'Set includes three uniquely designed rustic vases\n' +
            'Crafted from high-quality materials with a natural, hand-crafted finish\n' +
            'Perfect for displaying flowers, greenery, or as standalone decorative pieces\n' +
            'Versatile design complements both modern and traditional interiors\n' +
            'Ideal for gifting or personal use in any living space\n' +
            'Add warmth and character to your home with the Rustic Vase Set-where classic design meets natural beauty.',
        dicountPercentage: 10,
        isNew: false,
        price: 210,
        productImage: {
          _type: 'image',
          asset: {
            _ref: 'image-2219cafc285ec13a2ed3f88aa36cbea852a11735-305x375-png'
          }
        },
        tags: [ 'rustic ', 'vase ', 'home decor', 'vintage ', 'interior design' ],
        title: 'Rustic Vase Set'
}
Uploading image: https://cdn.sanity.io/images/7xt4qcah/production/cbfd4d2e1e3a58e25763a1d84efe10a8c25aeed7-3264x4928.jpg
Image uploaded successfully: image-35a8072e40a8b80142ff8e96e0f3af21811f5b9f-3264x4928-jpg
Product Timber Craft uploaded successfully: {
  _createdAt: '2025-01-18T19:42:54Z',
  _id: 'sDoE4Hn6nkBPnCufhSU8At',
  _rev: 'sDoE4Hn6nkBPnCufhSU85r',
  _type: 'product',
  _updatedAt: '2025-01-18T19:42:54Z',
  description: 'Introducing TimberCraft-a collection that celebrates the timeless beauty of wood craftsmanship and the art of nature's finest materials. Ins
  ed by the rustic charm and durability of natural wood, TimberCraft brings warmth, character, and a touch of handcrafted elegance to any space. Perfect for t
  e who value authenticity and sustainability, this collection combines the strength of timber with refined design, making it the ideal choice for modern and
  ditional homes alike.\n' +
  '\n' +
  "Each piece in the TimberCraft collection is meticulously crafted to highlight the unique grains and textures of the wood, bringing an organic, earthy f
  to your space. Whether you're looking for furniture, decor, or accents, TimberCraft offers a variety of beautifully designed pieces that blend rustic appea
  ith contemporary sophistication. The collection is designed to stand the test of time, with materials chosen for their durability and lasting beauty.\n" +
```



```
S  Default    + Create    Q                              Structure  Vision  Schedules                          Tasks  A

DATASET              API VERSION          CUSTOM API VERSION        PERSPECTIVE               QUERY URL [COPY TO CLIPBOARD]
production     v     Other          v     v2025-01-18              raw              v         https://amke9rf5.api.sanity.io/v2025-01-18/data/query/p

QUERY                                      RESULT
1  *[type==product]                        [...] 58 items
                                           v 0: {...} 12 properties
                                               _createdAt: 2025-01-18T19:42:41Z
                                               _type: product
                                               _id: YTjpX7gIRs7VphrWQQzxvA
                                               isNew: false
                                               _updatedAt: 2025-01-18T19:42:41Z
                                               dicountPercentage: 10
                                             v tags: [...] 5 items
                                                 0: rustic
                                                 1: vase
PARAMS                                           2: home decor
What's new                                       3: vintage
Sanity Create Content Mapping, Visual Editing,   4: interior design
and Content Releases                           v productImage: {...} 2 properties
                                                 _type: image
                                             v asset: { } 1 property
  ►  Fetch        ►  Listen         Execution: 22ms   End-to-end: 242ms           Save result as   JSON   CSV
```


PREPARED BY AYESHA MARYAM