Day 3 - API Integration Report – HEKTO

1. **Sanity Project Initialization**
   - Created a new Sanity project.
   - Added `NEXT_PUBLIC_SANITY_PROJECT_ID`, `SANITY_API_TOKEN`, and `NEXT_PUBLIC_SANITY_DATASET` in `.env.local`.
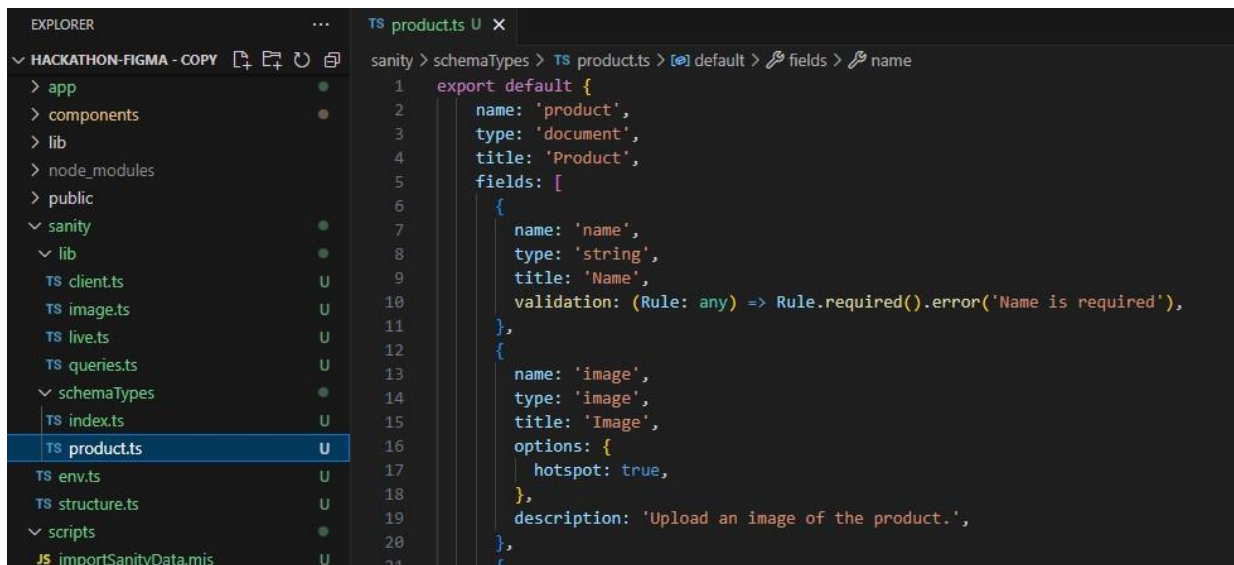
2. **Schema Setup**
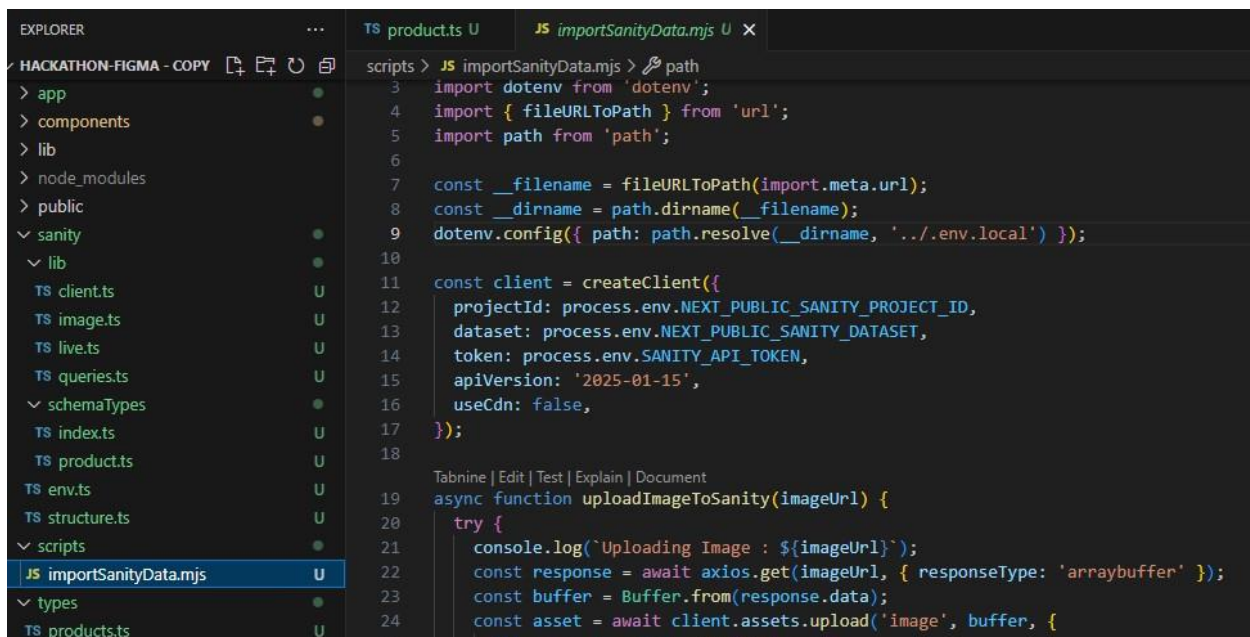   - Created `product.ts` in the schema folder.
   - Defined the product schema with fields like `name`, `price`, `image`, `category`, `stockLevel`.
   - Imported the schema into `index.ts` in the schema folder.

```ts
export default {
    name: 'product',
    type: 'document',
    title: 'Product',
    fields: [
      {
        name: 'name',
        type: 'string',
        title: 'Name',
        validation: (Rule: any) => Rule.required().error('Name is required'),
      },
      {
        name: 'image',
        type: 'image',
        title: 'Image',
        options: {
          hotspot: true,
        },
        description: 'Upload an image of the product.',
      },
```

3. **Data Import**

- Created `importSanityData.mjs` in the `scripts` folder.
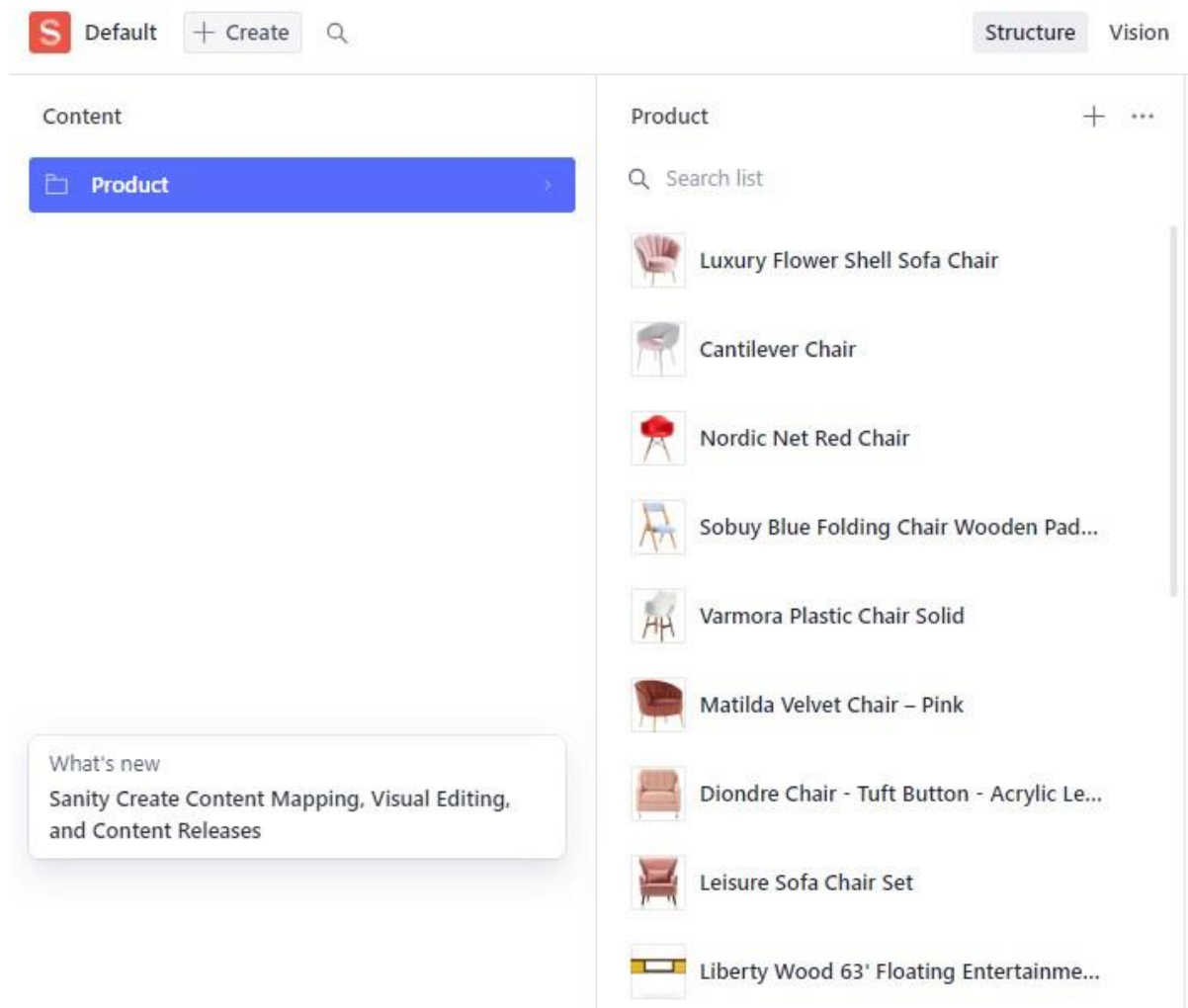- Wrote a script to fetch product data from an external API and upload it to Sanity CMS.


```js
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.local') });

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  token: process.env.SANITY_API_TOKEN,
  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, {
```

Added the following script in `package.json`:

```
{} package.json > {} scripts
  1  {
  2    "name": "hackathon-figma",
  3    "version": "0.1.0",
  4    "private": true,
  5    "type": "module",
       ▷ Debug
  6    "scripts": {
  7      "dev": "next dev --turbopack",
  8      "build": "next build",
  9      "start": "next start",
 10      "lint": "next lint",
 11      "import-data": "node scripts/importSanityData.mjs"
 12    },
```

`npm run import-data` to import data into Sanity Studio.

4. Fetching Data in Next.js

- Created `product.ts` in the `types` folder to define the Product interface.
- Created `queries.ts` in the `lib` folder and wrote a GROQ query to fetch product data:
- Updated `next.config.js` to allow images from Sanity:

5. Displaying Products in Next.js

- Used `useEffect` in a component to fetch products from Sanity:

Our Furniture

Rapson Thirty-Nine Guest Chair

A mid-century modern chair with clean lines and durable materials.

Price: $1300

High Quality Modern Customized Plastic Chair

A simple and lightweight chair ideal for dining or casual use.

Price: $150

Alpha Chair – Solid Ebonised Oak

A sturdy oak chair with a sleek and minimalist design.

Price: $900