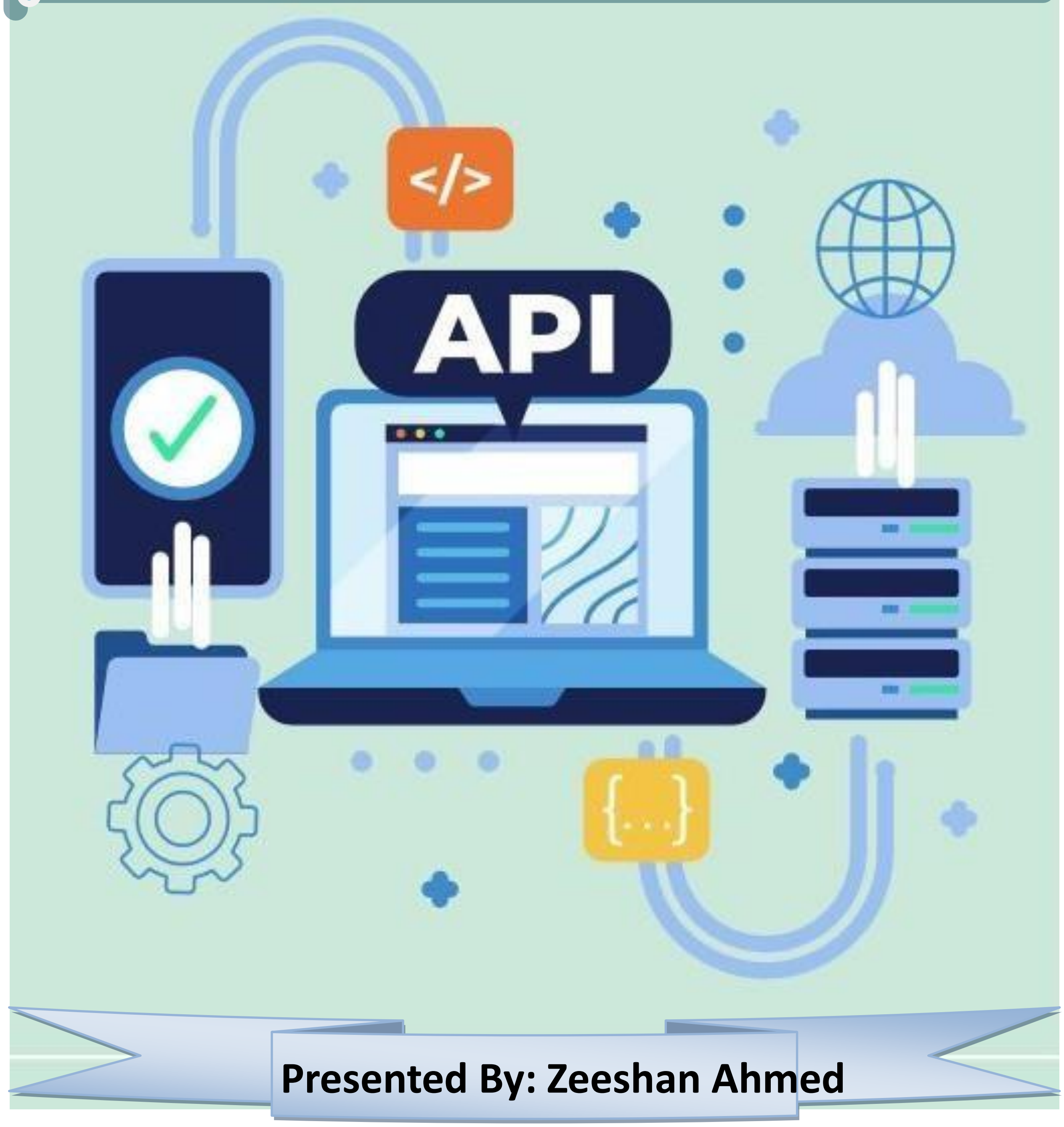
API Integration and Data Migration Report



Presented By: Zeeshan Ahmed

This documentation explains the work done on Day 3 of the E-commerce Marketplace hackathon. It includes custom database setup, integrating data from Sanity, creating schemas, and showing data using GROQ queries in a Next.js app. Each part is explained simply with details about how the code works.

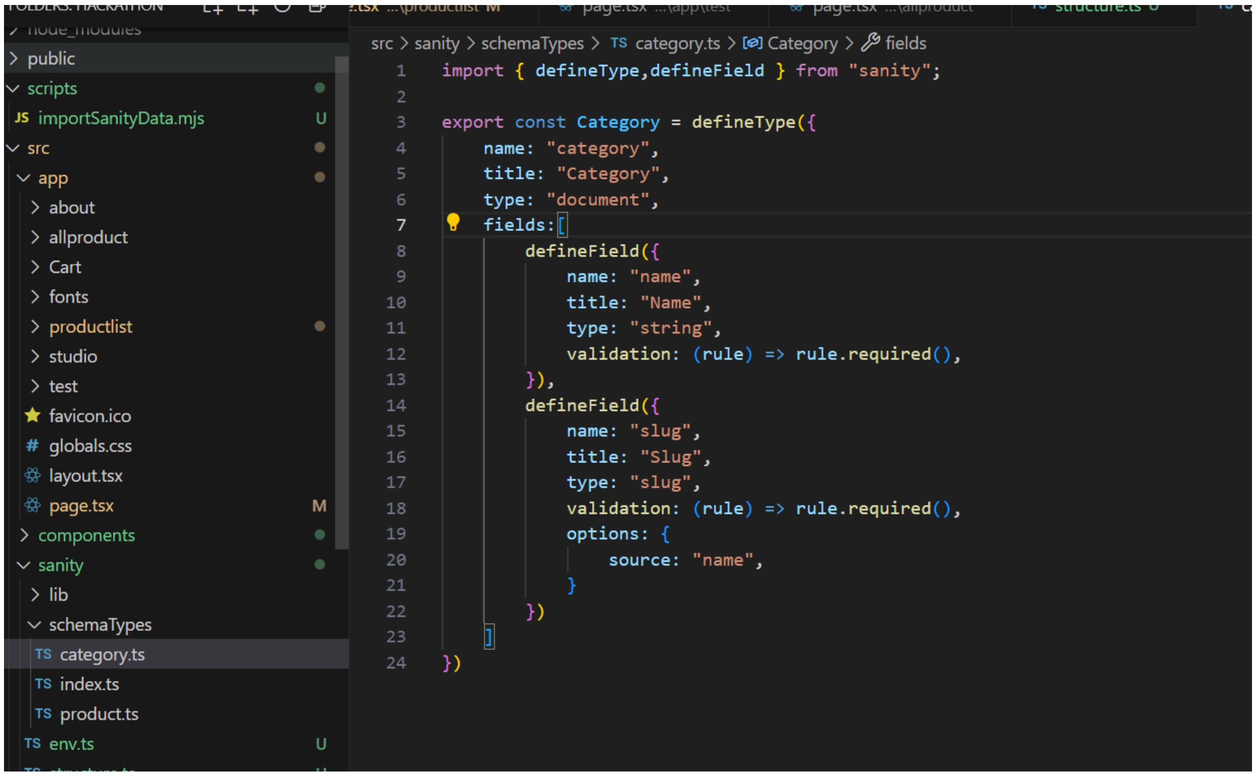
1. Sanity CMS Schema Design:

To ensure the smooth handling of Furniture data, I designed a schema called **Furniture** in Sanity CMS. The schema includes the following fields:

Product Schema Fields:

- **Name:** The name of the product (text).
- **Category:** The category of the product (text).
- **Type:** The type or category of the product (text).
- **Original Price:** The original price of the product (number).
- **Tags:** A list of tags for categorization (list of text). ➢ **Image:** The product's image (image).

Code Snap:



```
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",
            }
        })
    ]
})
```

2. API Integration and Data Migration:

API Data Fetching:

I retrieved product data from an external API, including:

- Name
- Category
- Type
- Original Price
- Tags
- Image

This data was mapped directly to the Sanity CMS schema.

Data Population in Sanity CMS:

After fetching data from the API, I filled the product fields in Sanity CMS automatically. This made sure the product information was consistent and accurate across the platform.

Data Migration:

Using the Sanity CLI, I backed up the dataset from Sanity CMS and later imported it again for testing. This ensured all data was well-organized and displayed correctly on the frontend.

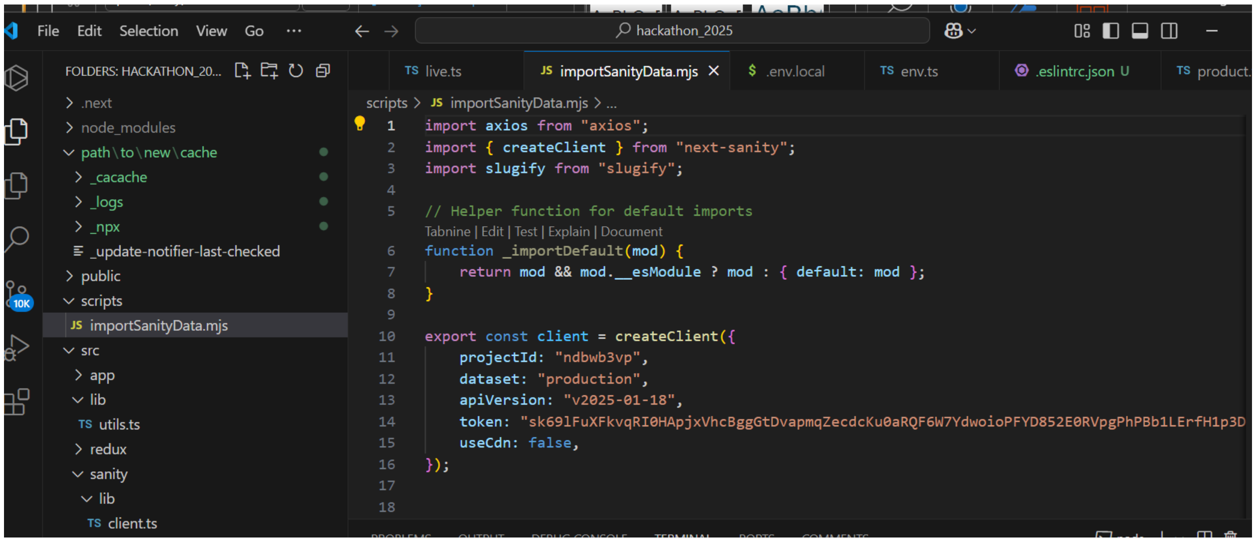
3. Steps Taken for Data Migration:

Exporting Data:

The first step was to export the data from Sanity CMS using the Sanity CLI. This made sure all the car data was safely backed up before doing anything else.

Verification of Data:

The exported JSON file was checked to make sure all the fields were filled correctly. This step ensured the data would be shown properly on the frontend when fetched.



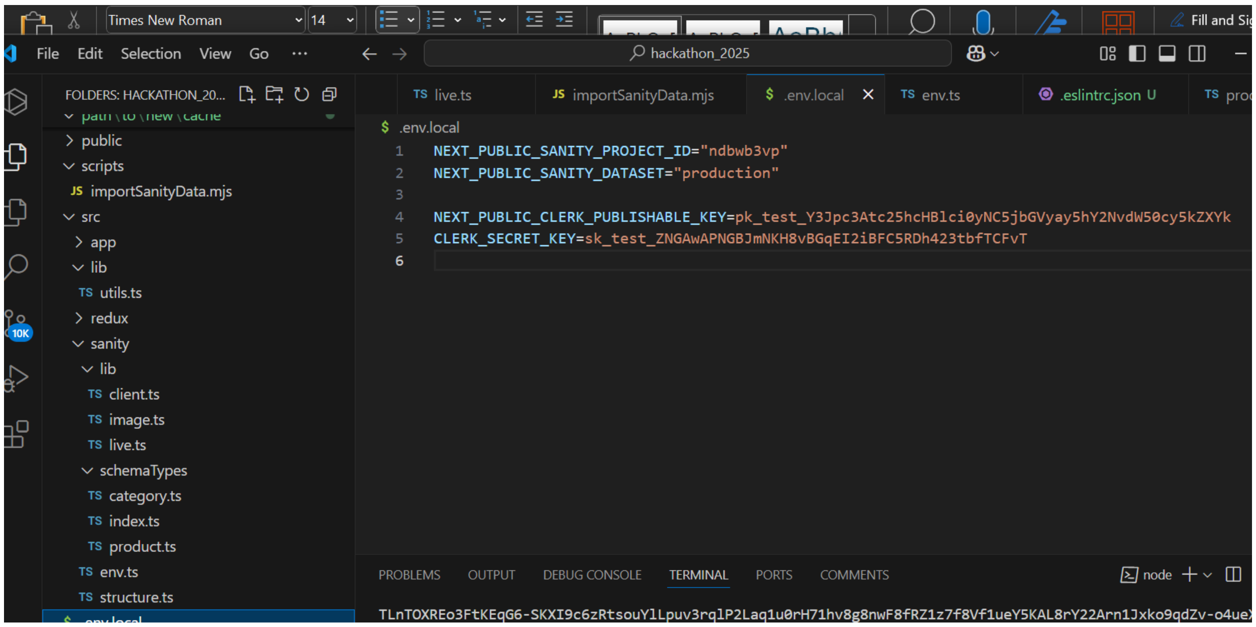
```
import axios from "axios";
import { createClient } from "next-sanity";
import slugify from "slugify";

// Helper function for default imports
function _importDefault(mod) {
    return mod && mod.__esModule ? mod : { default: mod };
}

export const client = createClient({
    projectId: "ndbwb3vp",
    dataset: "production",
    apiVersion: "v2025-01-18",
    token: "sk69lFuXFkvqRI0HApjxVhcBggGtDvapmqZecdcKu0aRQF6W7YdwoioPFYD852E0RVpgPhPBb1LErfH1p3D
    useCdn: false,
});
```

- ## Environment Variables

The .env file contains sensitive settings for the Product application. Key entries:

- ## Item Product Code

This code defines the design and functionality of a single product card. It is used on the client page to display individual items.

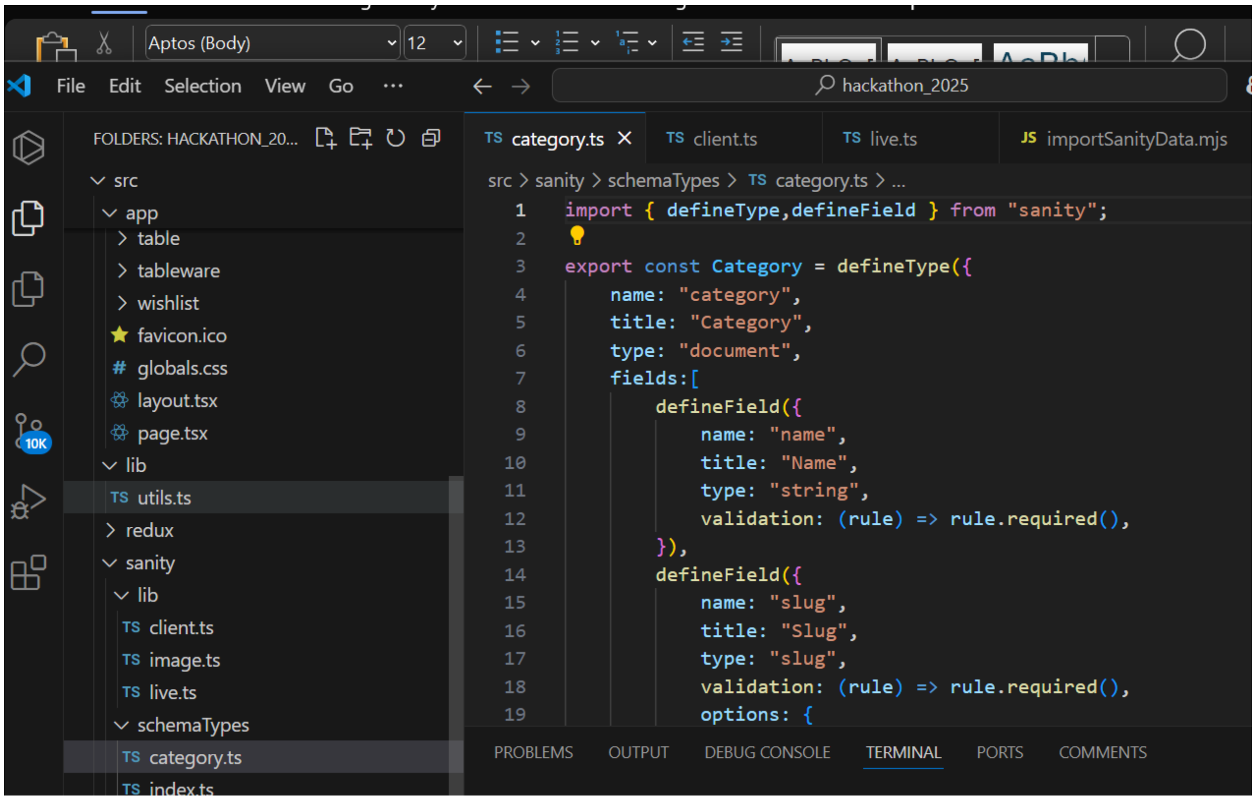


```
NEXT_PUBLIC_SANITY_PROJECT_ID="ndbwb3vp"
NEXT_PUBLIC_SANITY_DATASET="production"

NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY=pk_test_Y3Jpc3Atc25hcHBlci0yNC5jbGVyay5hY2NvdW50cy5kZXYk
CLERK_SECRET_KEY=sk_test_ZNGAwAPNGBJmNKH8vBGqEI2iBFC5RDh423tbfTCFvT

```

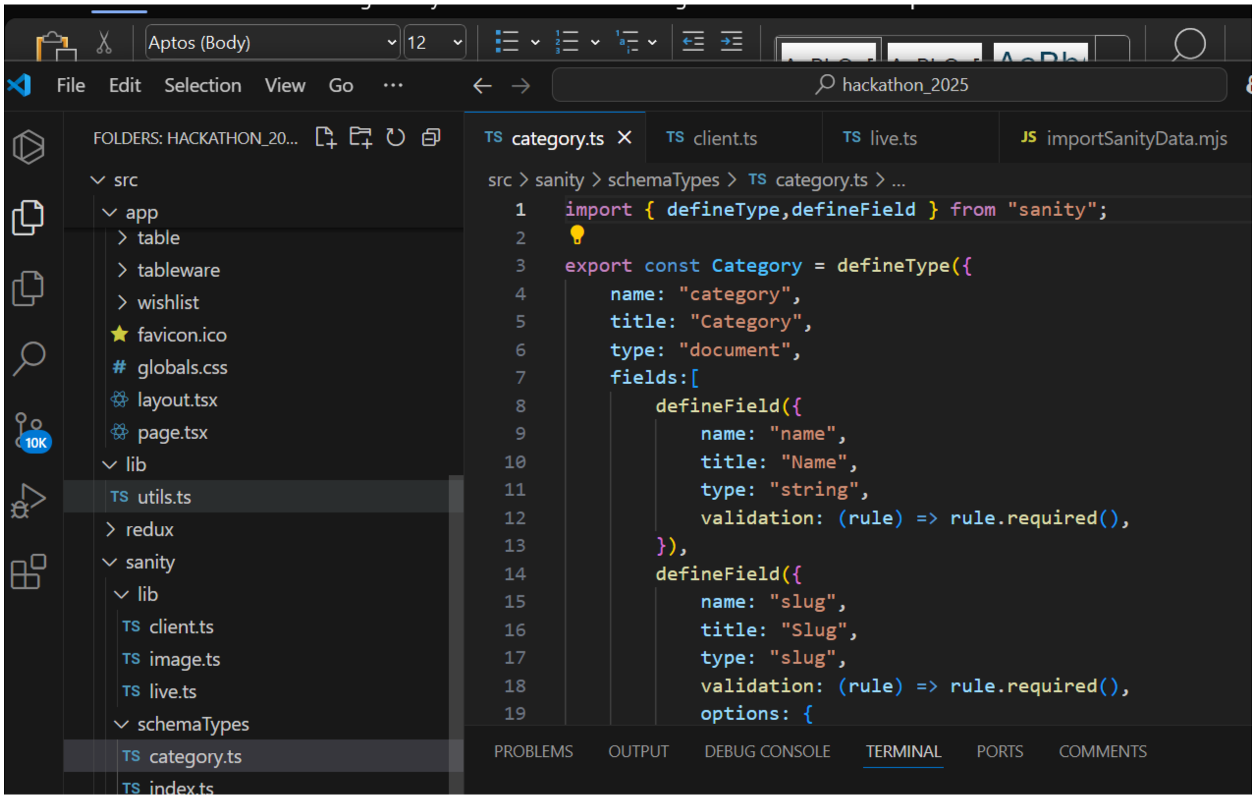
```
TLnTOXREo3FtKEqG6-SKXI9c6zRtsouYlLpuv3rqlP2Laq1u0rH71hv8g8nwF8fRZ1z7f8Vf1ueY5KAL8rY22Arn1Jxko9qdZv-o4ueX
```

- **Next.js Image Configuration**

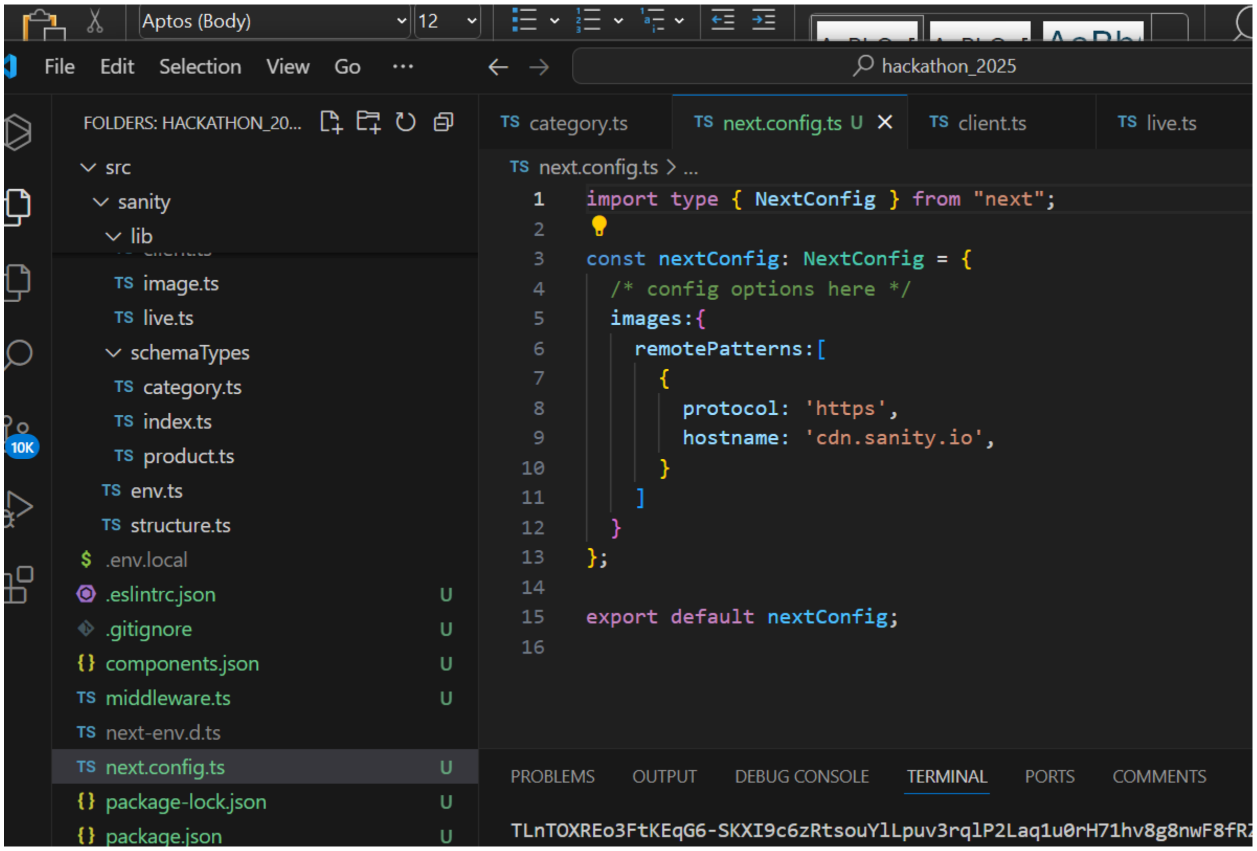
This code is used to allow images from an external source (like Sanity) to be used in a Next.js application. It specifies that images with the hostname "cdn.sanity.io" and the "https" protocol can be displayed in the app.



```
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: {
```



```
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: {
```



```ts
import type { NextConfig } from "next";

const nextConfig: NextConfig = {
  /* config options here */
  images:{
    remotePatterns:[
      {
        protocol: 'https',
        hostname: 'cdn.sanity.io',
      }
    ]
  }
};

export default nextConfig;
```

```
TLnTOXREo3FtKEqG6-SKXI9c6zRtsouYlLpuv3rqlP2Laq1u0rH71hv8g8nwF8fRz
```

Re-importing Data:

After checking the data, it was re-imported into Sanity CMS. This confirmed that the data migration was successful and everything was working as expected.

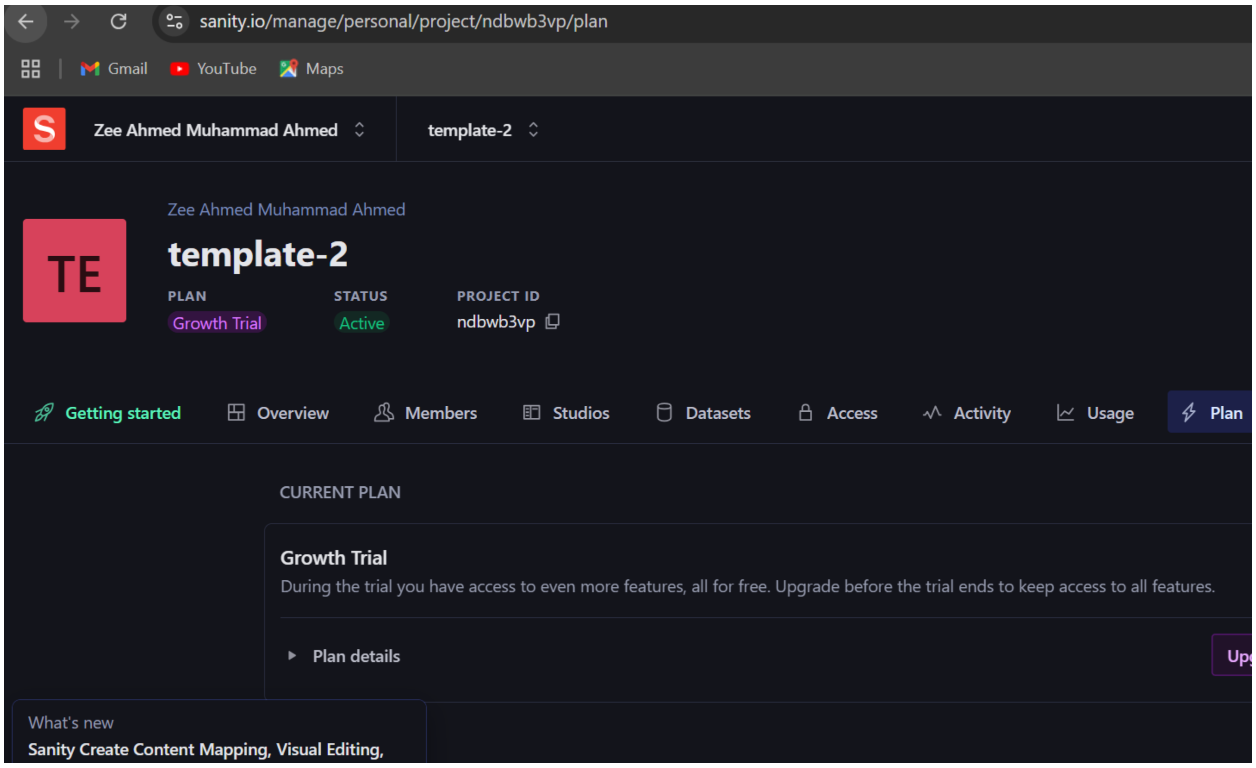
4. Tools Used:

- **Sanity Studio:**

Used to create schemas, manage content, and display car data.

- **Sanity Vision:**

A tool used for testing and debugging GROQ queries to fetch and preview data in Sanity Studio.



Zee Ahmed Muhammad Ahmed

template-2

PLAN: Growth Trial
STATUS: Active
PROJECT ID: ndbwb3vp

CURRENT PLAN

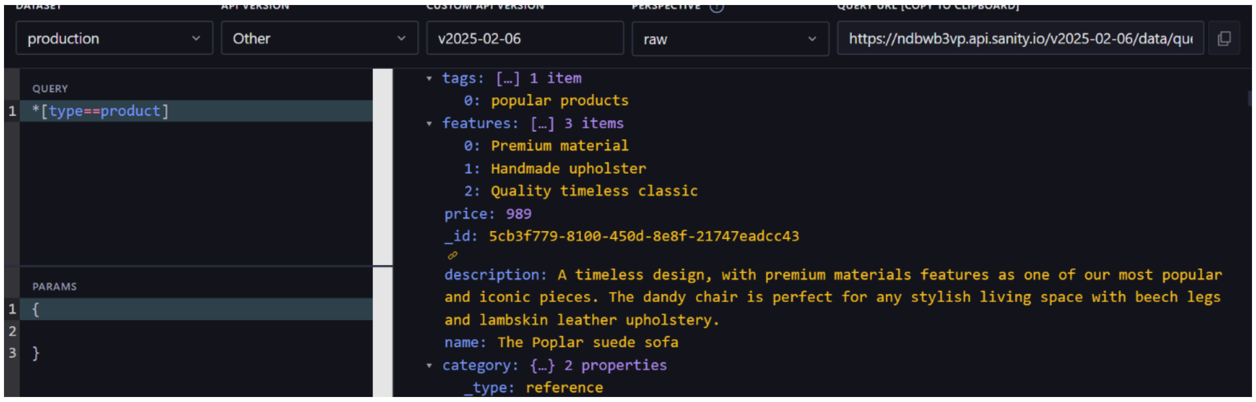
Growth Trial

During the trial you have access to even more features, all for free. Upgrade before the trial ends to keep access to all features.

Plan details

What's new

Sanity Create Content Mapping, Visual Editing,



production | Other | v2025-02-06 | raw | https://ndbwb3vp.api.sanity.io/v2025-02-06/data/qu

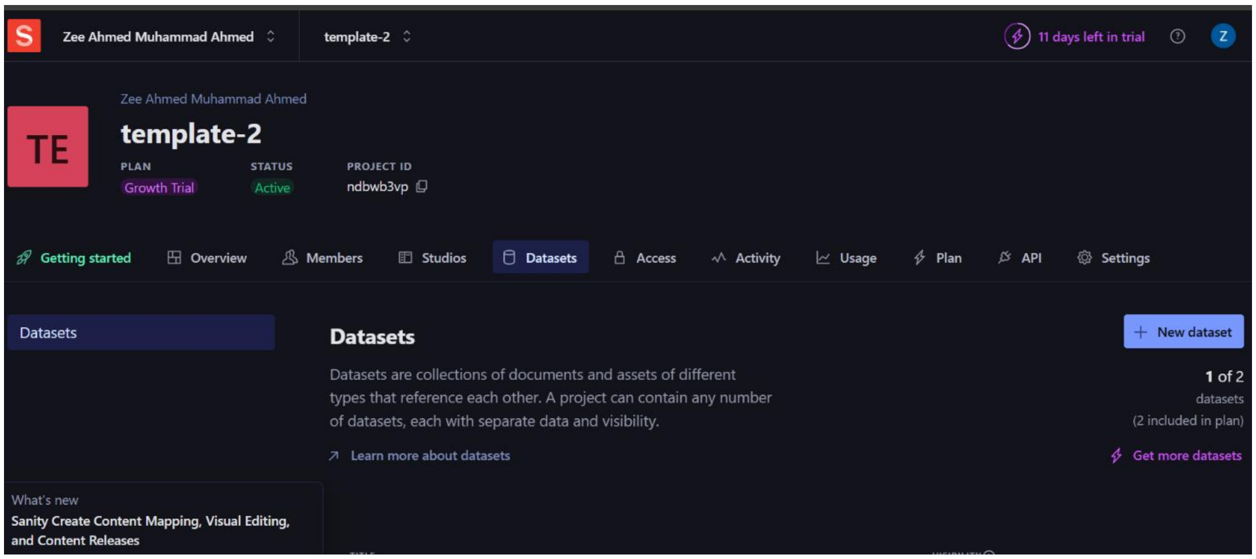
QUERY
```
*[type==product]
```

PARAMS
```
{

}
```

```
▾ tags: […] 1 item
    0: popular products
▾ features: […] 3 items
    0: Premium material
    1: Handmade upholster
    2: Quality timeless classic
  price: 989
  _id: 5cb3f779-8100-450d-8e8f-21747eadcc43
  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 upholstery.
  name: The Poplar suede sofa
▾ category: {…} 2 properties
    _type: reference
```

Sanity Database:

A cloud-based database used to store and manage structured content, like product data, for the application.



Datasets

Datasets are collections of documents and assets of different types that reference each other. A project can contain any number of datasets, each with separate data and visibility.

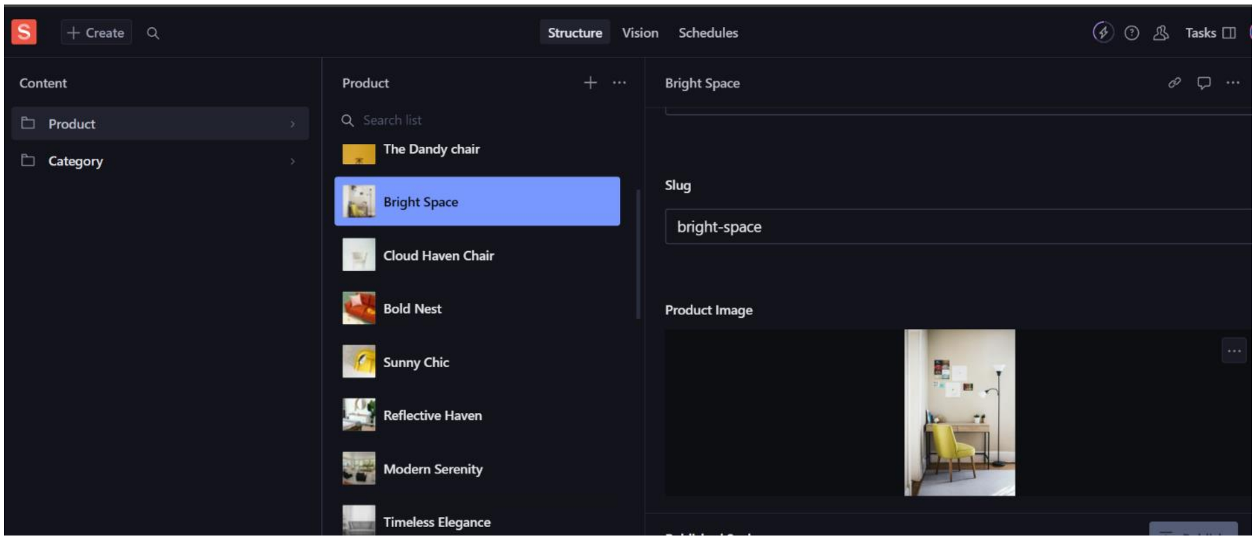
- ## Sanity CLI:

Used for exporting and importing datasets to ensure data is consistent and backed up.

5. Screenshots and Frontend Display:

- **Sanity CMS Fields**:

A screenshot showing the filled fields in Sanity Studio, displaying the product details like name, brand, type, fuel capacity, price per day, and images.



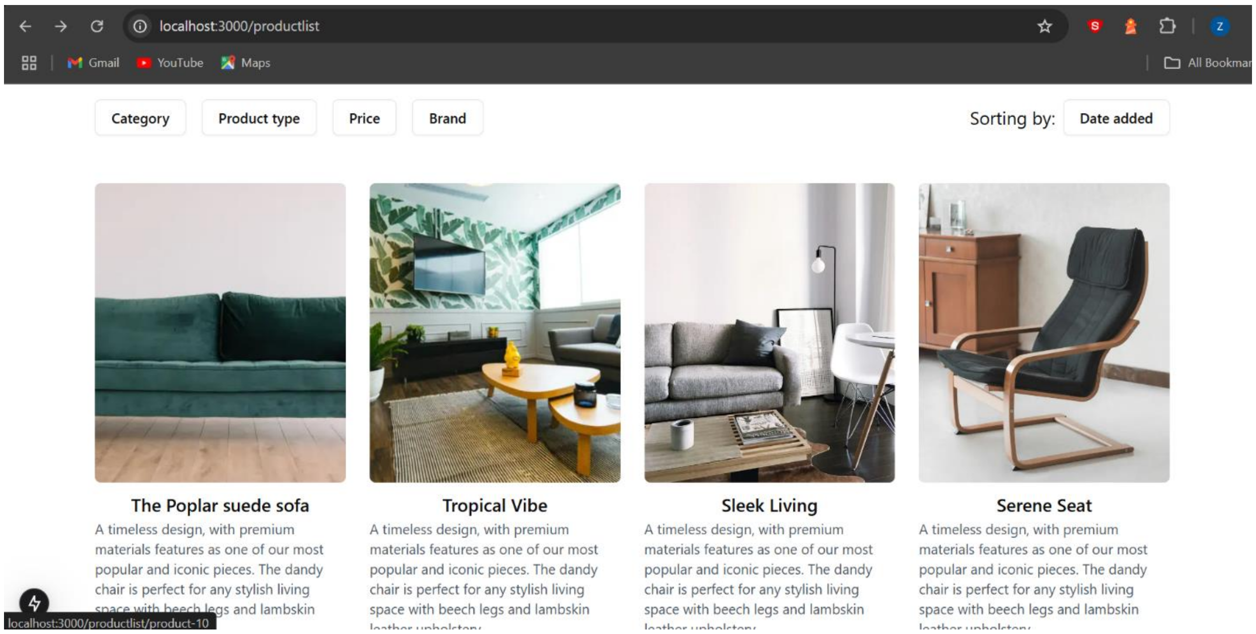
Slug

bright-space

Product Image

 Frontend Display:

A screenshot showing how the car data is dynamically displayed on the frontend of the rental marketplace.



The Poplar suede sofa
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

Tropical Vibe
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

Sleek Living
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

Serene Seat
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

Key Steps for Sanity API Integration and Data Migration

1. Sanity API Setup:

- Connects to Sanity's API using a dataset and project ID with an API token for security. Environment variables (like project ID and dataset) keep sensitive information safe.

2. Fetching Data from Sanity:

- Uses GROQ queries to get structured data from Sanity CMS.
- Example: Queries fetch car categories, descriptions, prices, and other details.

3. Mapping and Formatting:

- Matches the fetched data with the RentalHub schema.
- Formats records to fit the application's requirements.

4. Saving to Database:

- Saves data into the RentalHub database through REST API calls or direct commands. Handles errors to log issues without stopping the process.

5. Code Highlights:

- Reusability: Functions can be reused for future migrations.
- Efficiency: Bulk data insertion reduces API calls and works faster.

6. Conclusion:

The API integration and data migration were successfully completed, improving the efficiency and scalability of the RentalHub project. This integration made it easier to add and update car data in the marketplace, while the migration steps ensured data accuracy and consistency throughout the system. With this setup, the product project is now more dynamic and easier to manage.