

# Day 3 - API Integration Progress

## Car Rental Marketplace

### API Integration Workflow

#### 1. Analyzing the Requirements:

The objective was to seamlessly integrate the provided API (given by Sir Asharib Ali) into the Sanity CMS and ensure the data is correctly displayed on the frontend of our Car Rental Marketplace, built using Next.js.

#### 2. Configuring the Development Environment:

Initially, I installed Sanity in the Next.js project folder using the command:  
`npm install -g @sanity/cli` `sanity init`.

Next, I added the Sanity project ID and API token to the .env file for secure configuration, like this:

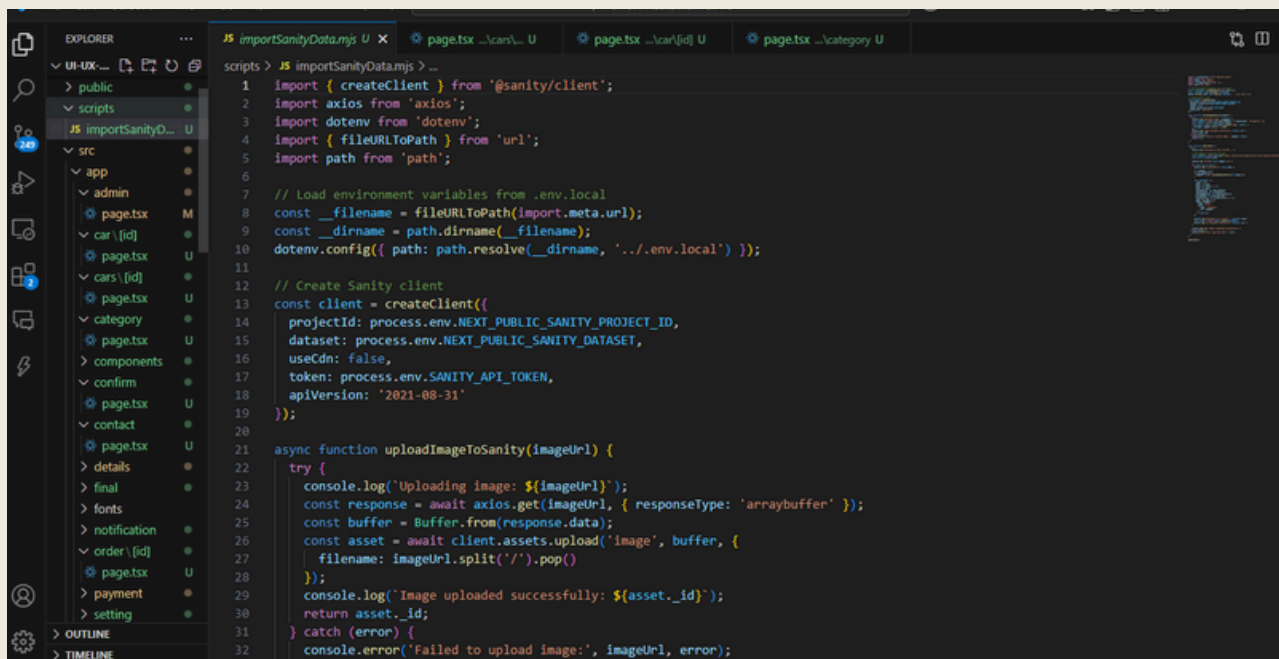
`SANITY_PROJECT_ID=your_project_id`

`SANITY_DATASET=production`

`SANITY_API_TOKEN=your_api_token`

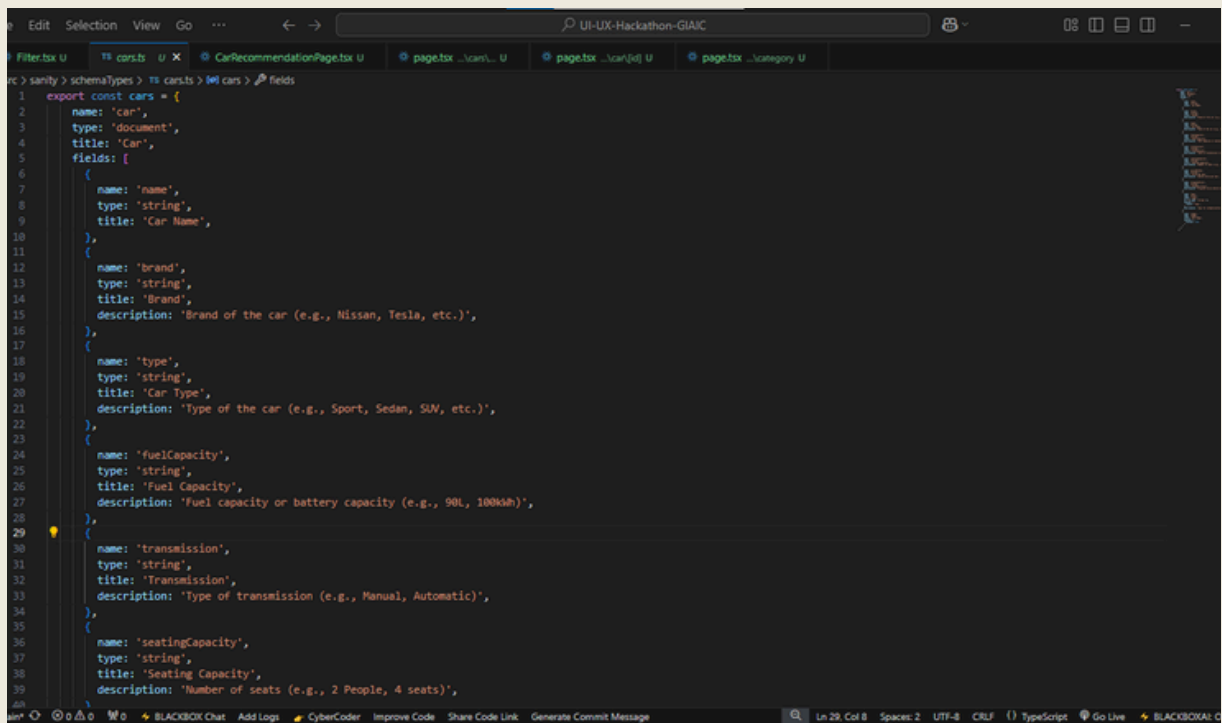
#### 3. Data Migration Process:

I created a new file called `ImportSanityData.mjs` in the project and used the `importData` script provided by Sir Asharib Ali. Then, I integrated the API endpoint into the `ImportSanityData.mjs` file by adjusting the script to fetch the API data and populate it within the Sanity CMS.



```
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import dotenv from 'dotenv';
4 import { fileURLToPath } from 'url';
5 import path from 'path';
6
7 // Load environment variables from .env.local
8 const __filename = fileURLToPath(import.meta.url);
9 const __dirname = path.dirname(__filename);
10 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12 // Create Sanity client
13 const client = createClient({
14   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
15   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16   useCdn: false,
17   token: process.env.SANITY_API_TOKEN,
18   apiVersion: '2021-08-31'
19 });
20
21 async function uploadImageToSanity(imageUrl) {
22   try {
23     console.log('Uploading image: ${imageUrl}');
24     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
25     const buffer = Buffer.from(response.data);
26     const asset = await client.assets.upload('image', buffer, {
27       filename: imageUrl.split('/').pop()
28     });
29     console.log('Image uploaded successfully: ${asset._id}');
30     return asset._id;
31   } catch (error) {
32     console.error('Failed to upload image:', imageUrl, error);
33   }
34 }
```

- Schema Modifications: I created a new schema file, `car.ts`, within the `schemaTypes` folder of the Sanity project. In this file, I defined the schema for car details, aligning it with the structure of the API data.



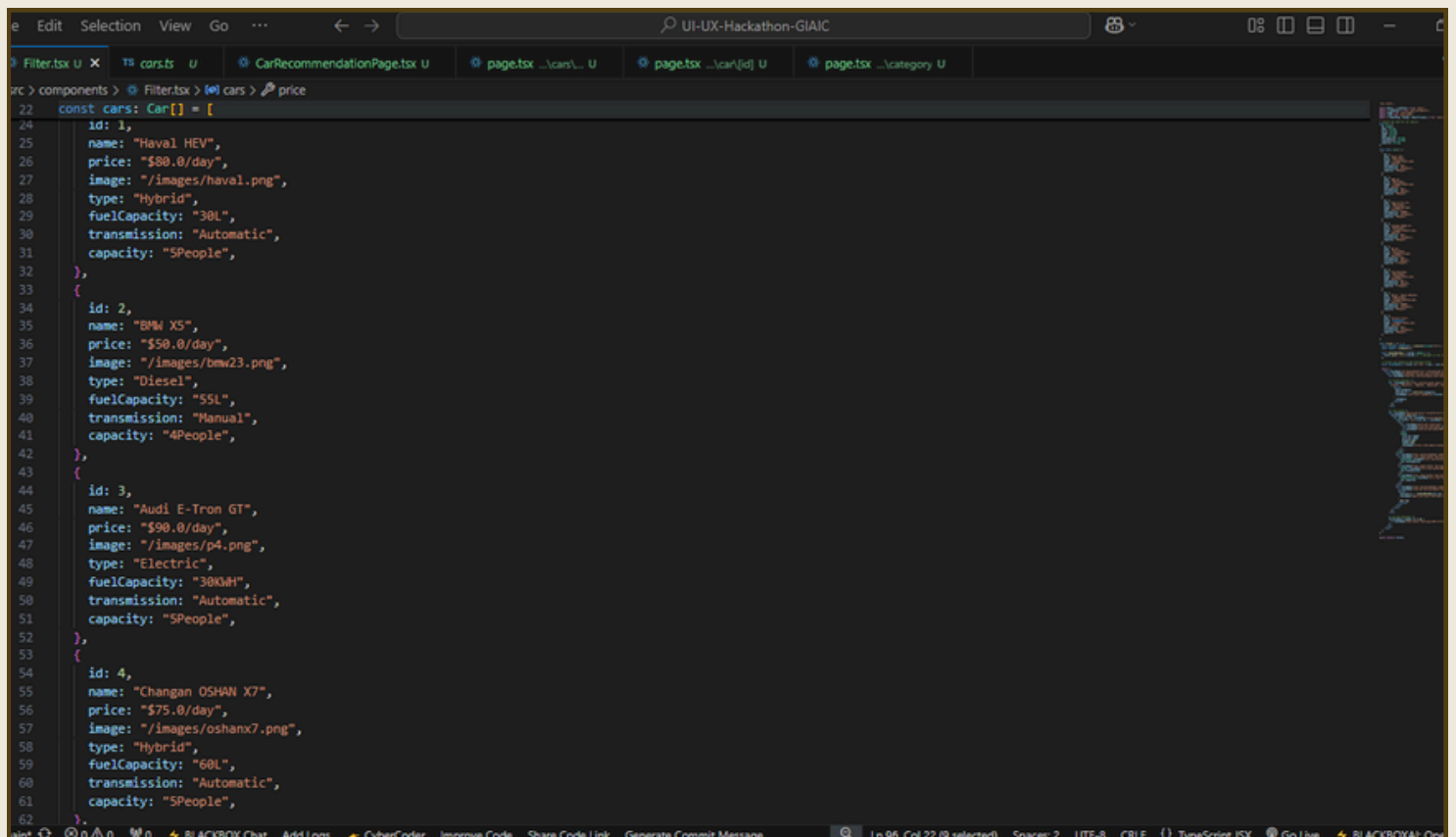
```
src > sanity > schemaTypes > ts cars > fields
1  export const cars = {
2    name: 'car',
3    type: 'document',
4    title: 'Car',
5    fields: [
6      {
7        name: 'name',
8        type: 'string',
9        title: 'Car Name',
10      },
11      {
12        name: 'brand',
13        type: 'string',
14        title: 'Brand',
15        description: 'Brand of the car (e.g., Nissan, Tesla, etc.)',
16      },
17      {
18        name: 'type',
19        type: 'string',
20        title: 'Car Type',
21        description: 'Type of the car (e.g., Sport, Sedan, SUV, etc.)',
22      },
23      {
24        name: 'fuelCapacity',
25        type: 'string',
26        title: 'Fuel Capacity',
27        description: 'Fuel capacity or battery capacity (e.g., 90L, 100kWh)',
28      },
29      {
30        name: 'transmission',
31        type: 'string',
32        title: 'Transmission',
33        description: 'Type of transmission (e.g., Manual, Automatic)',
34      },
35      {
36        name: 'seatingCapacity',
37        type: 'string',
38        title: 'Seating Capacity',
39        description: 'Number of seats (e.g., 2 People, 4 seats)',
40      },
41    ],
42  },
43 }
```

## Verification:

- I accessed the Sanity Studio and confirmed that the API data had been successfully populated within the CMS.

## Screenshots

### 1. API Calls:




```
src > components > Filter.tsx > cars > price
22  const cars: Car[] = [
23    {
24      id: 1,
25      name: "Haval HEV",
26      price: "$80.0/day",
27      image: "/images/haval.png",
28      type: "Hybrid",
29      fuelCapacity: "30L",
30      transmission: "Automatic",
31      capacity: "5People",
32    },
33    {
34      id: 2,
35      name: "BMW X5",
36      price: "$50.0/day",
37      image: "/images/bmw23.png",
38      type: "Diesel",
39      fuelCapacity: "55L",
40      transmission: "Manual",
41      capacity: "4People",
42    },
43    {
44      id: 3,
45      name: "Audi E-Tron GT",
46      price: "$90.0/day",
47      image: "/images/p4.png",
48      type: "Electric",
49      fuelCapacity: "300kWh",
50      transmission: "Automatic",
51      capacity: "5People",
52    },
53    {
54      id: 4,
55      name: "Changan OSHAN X7",
56      price: "$75.0/day",
57      image: "/images/oshanx7.png",
58      type: "Hybrid",
59      fuelCapacity: "60L",
60      transmission: "Automatic",
61      capacity: "5People",
62    },
63  ],
64 }
```

1. Data Displayed in Frontend:

localhost:3000/category

Audi E-Tron GT

Sport



90L

Manual

2 People


\$99.00/ day

\$200.00/day

Rent Now

BMW X5

Hatchback



90L

Manual

2 People


\$99.00/ day

\$200.00/day

Rent Now

Rolls- Royce Phantom

Sedan



90L

Manual

2 People


\$99.00/ day

\$200.00/day

Rent Now

MG HS

Manual




90L

Manual

2 People

Changan Oshan X7

Manual




90L

Manual

2 People

Corolla Cross



90L

Manual

2 People

Populated Sanity CMS Fields:

localhost:3000/studio/structure/car:UnfAWFpxkyivE5nYvinA6C

MORENT

Search something here

Default

+ Create

Structure

Vision

Schedules

Blog

Posts

Categories

Authors

Product

Car

Car

Search list

CR-V

Rolls-Royce

Nissan Altima

Chevrolet Camaro

Porsche 911

Mercedes-Benz C-Class

Audi A6

BMW X5

Ford Mustang

Nissan GT-R

Original price, taxes, discounts for approximately


\$100.00

Tags

Tags for categorization (e.g., popular, recommended)

popular

Car Image



What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases

## Collaboration Notes

### Learning and Knowledge Sharing:

- Knowledge sharing was crucial, with team members exchanging valuable insights on Sanity CMS, GROQ queries, and Next.js. While I completed the tasks independently, I collaborated with friends to resolve some errors and also assisted them in solving their own challenges.

- Feedback Exchange:

### **Conclusion:**

The successful integration of the API into the Sanity CMS and the display of data on the frontend of the Car Rental Website marks a key milestone in the project. By following a structured approach—from environment setup to schema adjustments and verification—the process ensured both accuracy and efficiency. This task not only enhanced understanding of Sanity CMS and API integration but also set the stage for future improvements in the marketplace platform. The outcome is a dynamic, CMS-driven car rental website, primed for scalability and future updates.