

Day 3 - API Integration Report - Marketplace

Car Rental Website

API Integration Process

1. Understanding the Requirements:

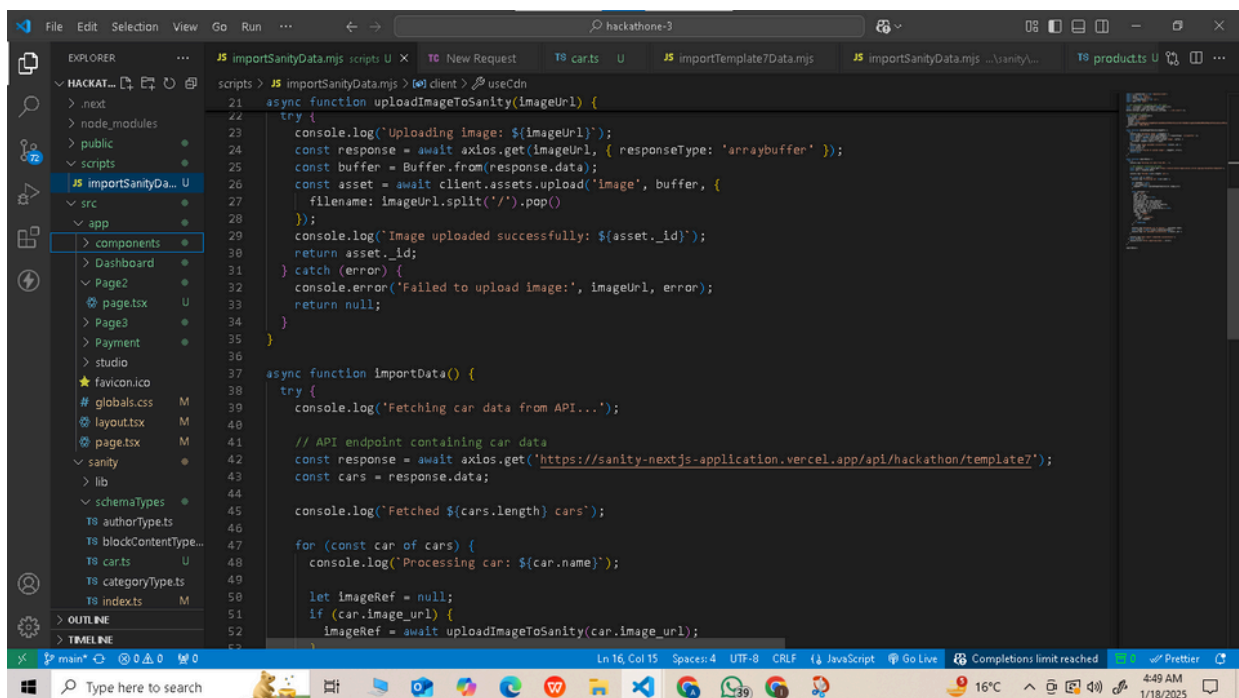
- The task was to integrate the given API for our marketplace (provided by Sir Asharib Ali) into the Sanity CMS and display the data in the frontend of the Marketplace (Car Rental Website using Next.js).

2. Setting Up the Environment:

- First Installed the Sanity in the Next.js project folder using the command:
- `npm install -g @sanity/cli`
- `sanity init`
- Added the Sanity project ID and API token to the .env file for secure configuration.
- `SANITY_PROJECT_ID=your_project_id`
- `SANITY_DATASET=production`
- `SANITY_API_TOKEN=your_api_token`

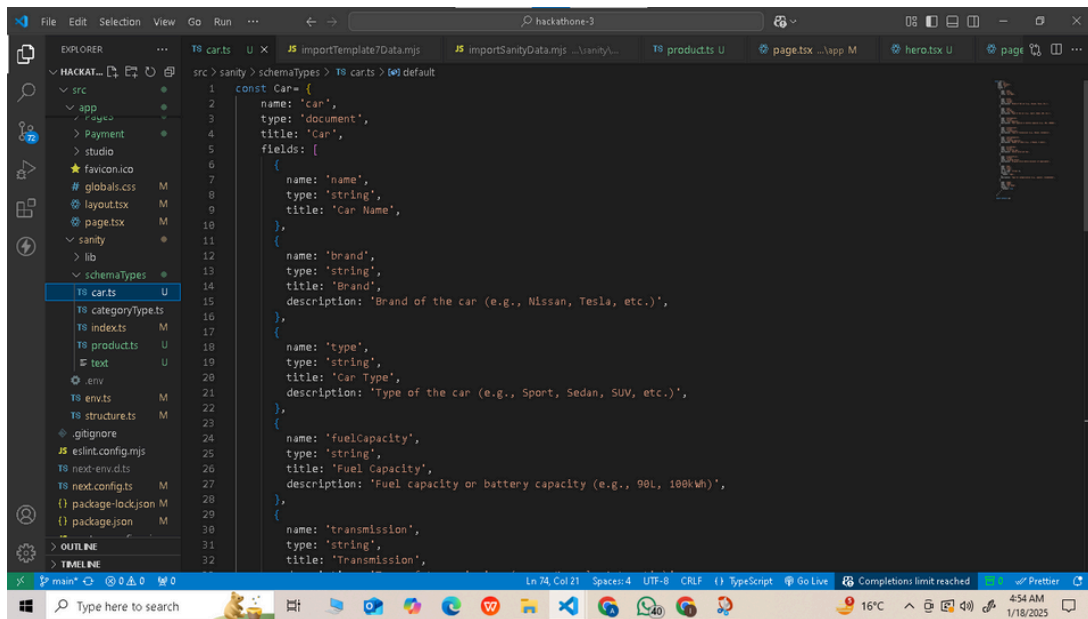
3. Data Migration:

- Created a new file `ImportSanityData.mjs` in the project and used the `importData` script provided by Sir Asharib Ali.
- Integrated the API endpoint into the `ImportSanity.mjs` file by modifying the script to fetch the API data and populate it in Sanity CMS.



```
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     return null;
34   }
35 }
36
37 async function importData() {
38   try {
39     console.log('Fetching car data from API...');
40
41     // API endpoint containing car data
42     const response = await axios.get('https://sanity-nextjs-application.vercel.app/api/hackathon/template?');
43     const cars = response.data;
44
45     console.log('Fetched ${cars.length} cars');
46
47     for (const car of cars) {
48       console.log('Processing car: ${car.name}');
49
50       let imageRef = null;
51       if (car.image_url) {
52         imageRef = await uploadImageToSanity(car.image_url);
53       }
54     }
55   }
56 }
```

- **Schema Adjustments:**
- Created a new schema file `car.ts` under the `schemaTypes` folder in the Sanity project and defined the schema for car details based on the API data.



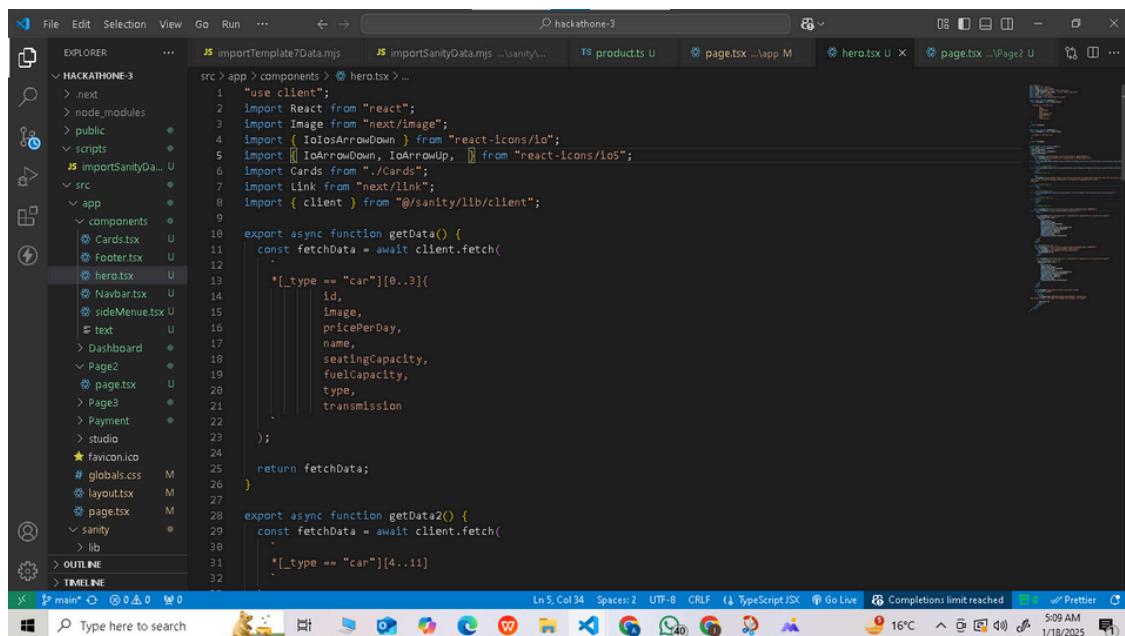
Verification:

- Navigated to the Sanity Studio and verified that the API data was successfully populated in the CMS.

○

● Frontend Integration:

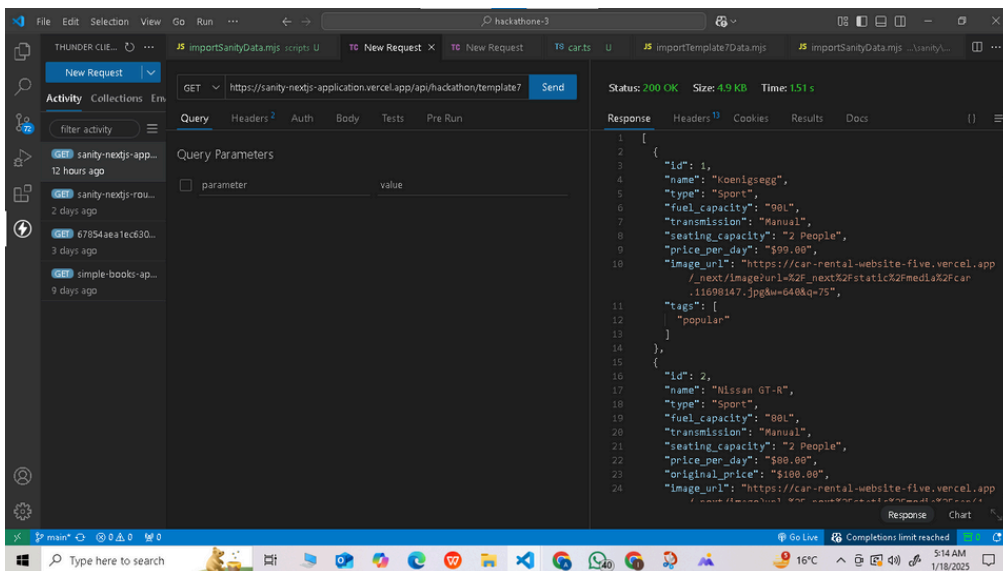
- Used a GROQ query to fetch the data from Sanity CMS in the Hero component:



Screenshots

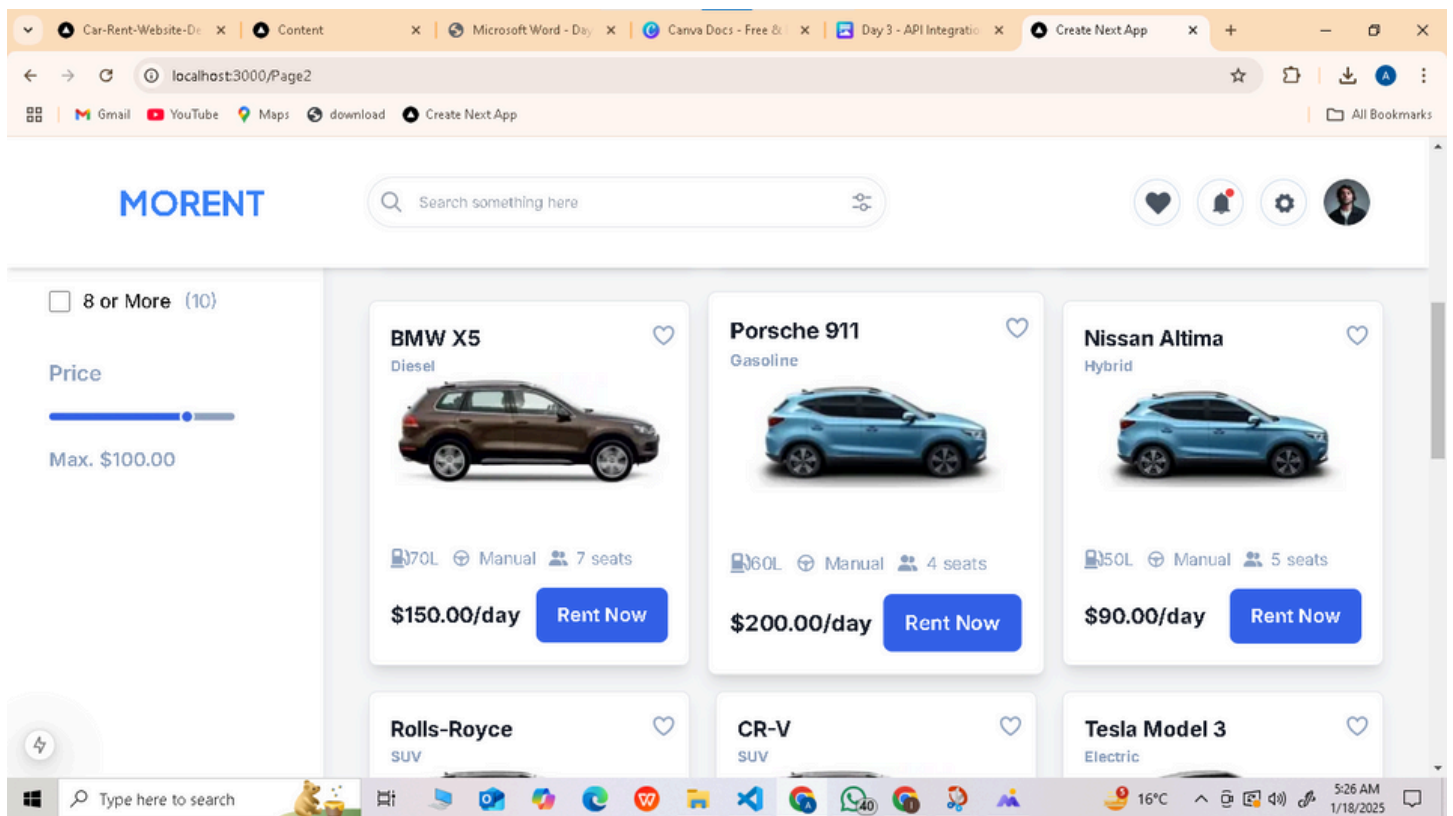
1. API Calls:

○

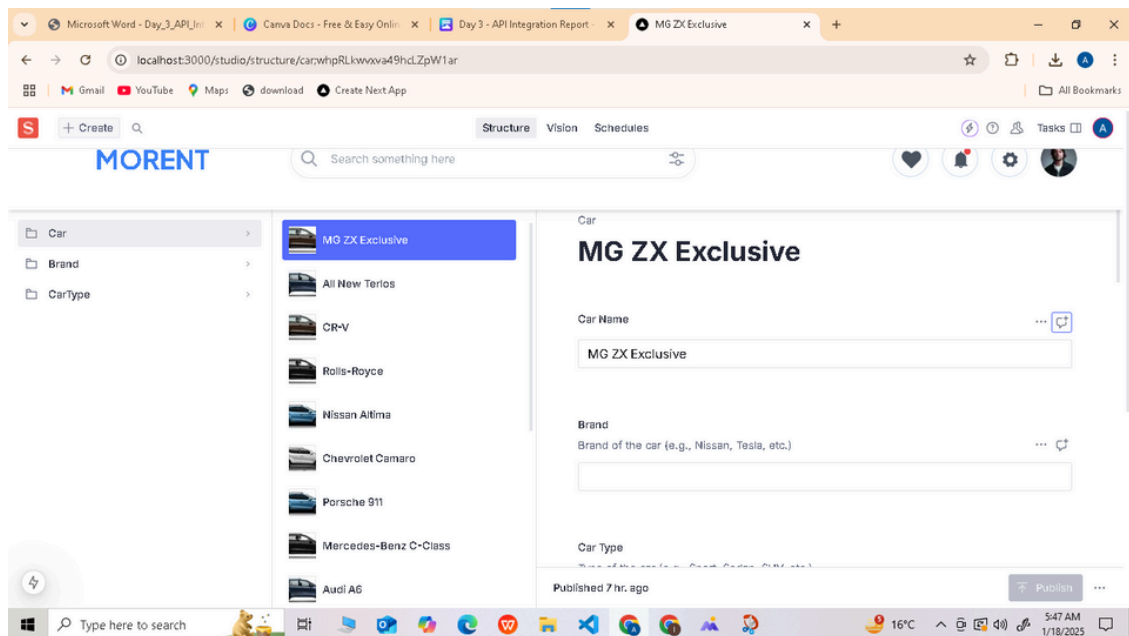


1. Data Displayed in Frontend:

○



Populated Sanity CMS Fields:



Collaboration Notes

Learning and Knowledge Sharing:

- Knowledge sharing played a vital role, with team members exchanging insights about Sanity CMS, GROQ queries, and Next.js.
- Done it individually but take help in some error solving with friend and help them in solving their problems.
- Taken and given then feedback

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 significant milestone in this project. By following a structured approach—starting from setting up the environment to schema adjustments and verification—the process ensured accuracy and efficiency. This task not only deepened understanding of Sanity CMS and API integration but also laid the foundation for further enhancements in the marketplace platform. The result is a dynamic, CMS-driven car rental website, ready for future scalability and updates.