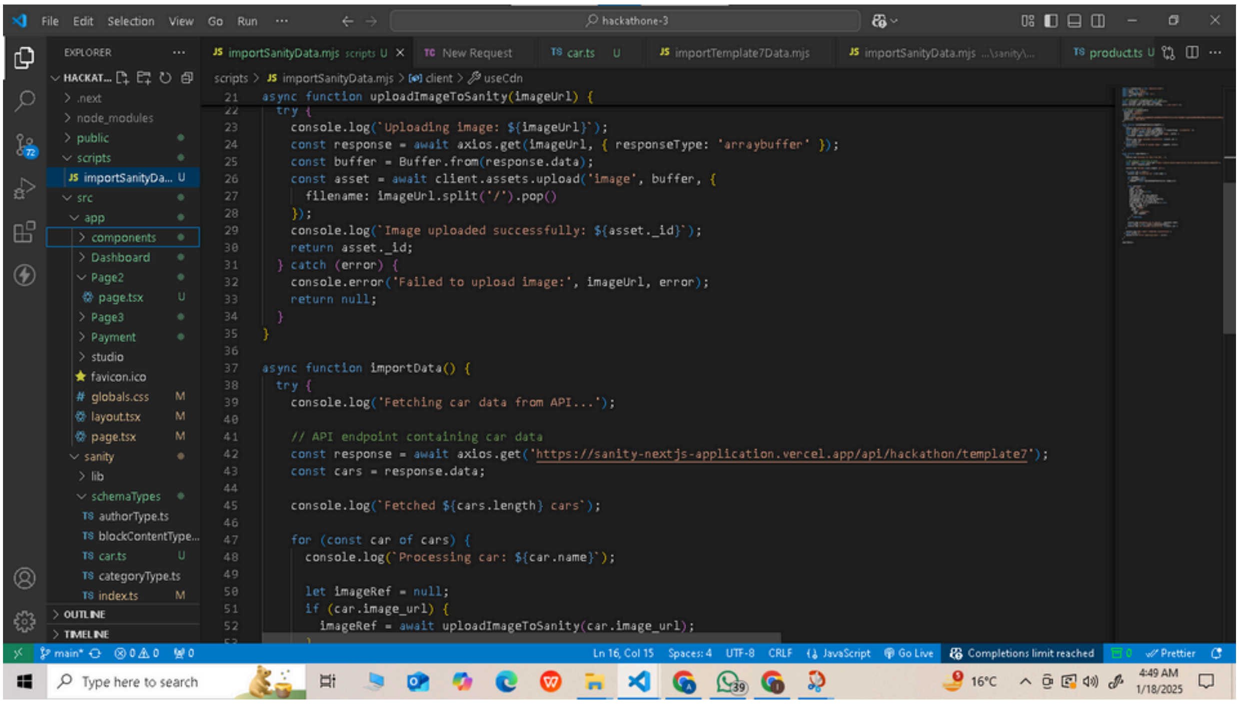
Day 3 - API Integration Report - Marketplace Car Rental Website

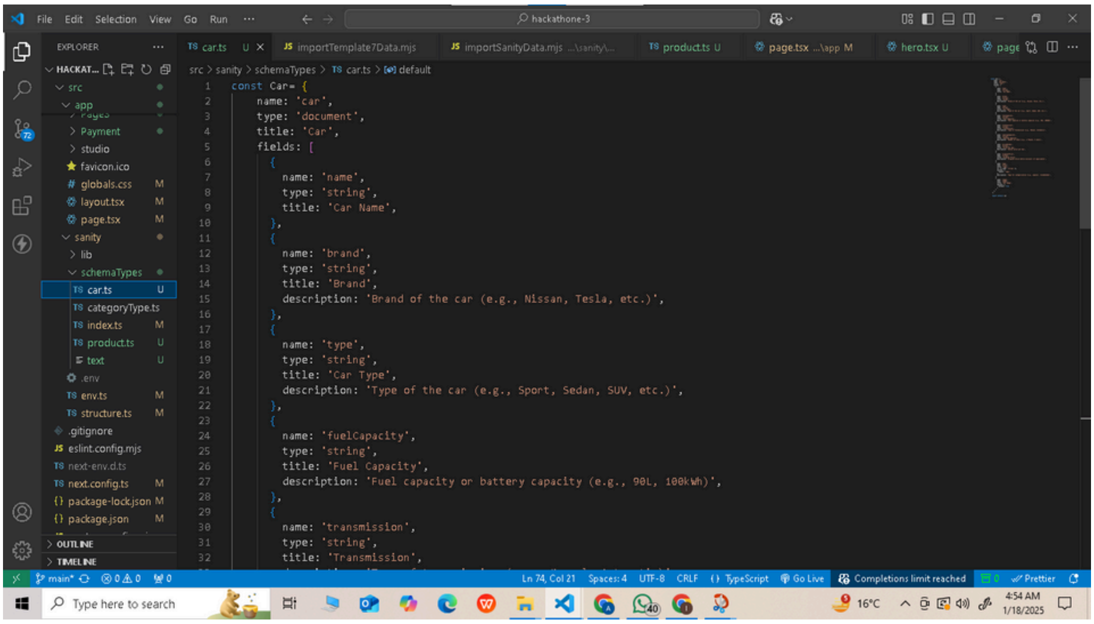
API Integration Process

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.



```
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, {
      filename: imageUrl.split('/').pop()
    });
    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}

async function importData() {
  try {
    console.log('Fetching car data from API...');

    // API endpoint containing car data
    const response = await axios.get('https://sanity-nextjs-application.vercel.app/api/hackathon/template7');
    const cars = response.data;

    console.log(`Fetched ${cars.length} cars`);

    for (const car of cars) {
      console.log(`Processing car: ${car.name}`);

      let imageRef = null;
      if (car.image_url) {
        imageRef = await uploadImageToSanity(car.image_url);
```

   - **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.

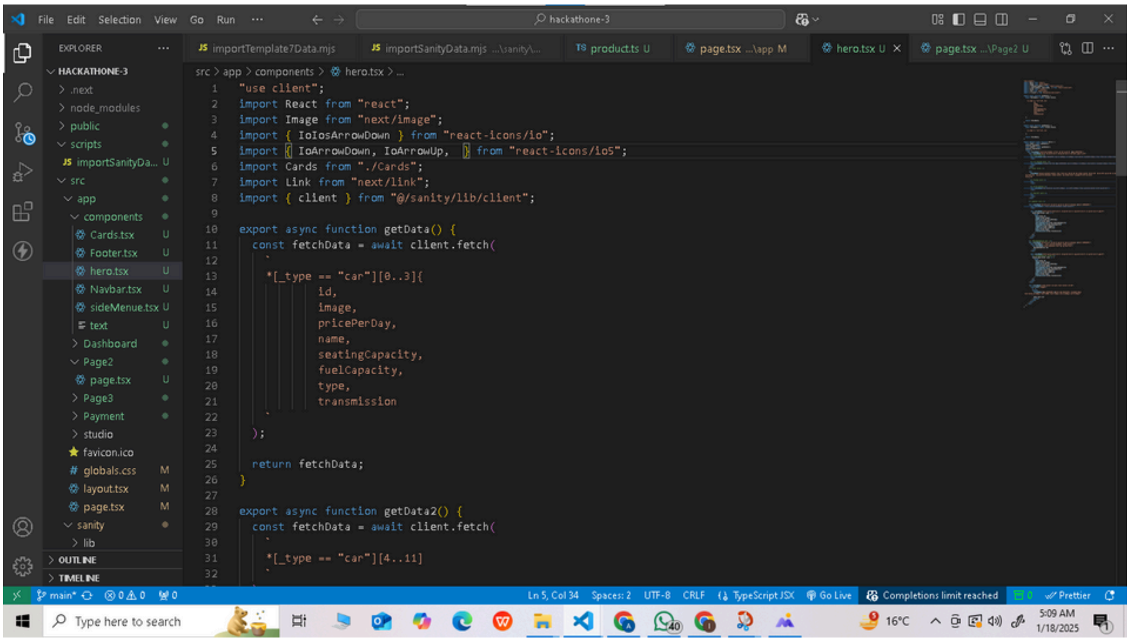


```
const Car= {
    name: 'car',
    type: 'document',
    title: 'Car',
    fields: [
      {
        name: 'name',
        type: 'string',
        title: 'Car Name',
      },
      {
        name: 'brand',
        type: 'string',
        title: 'Brand',
        description: 'Brand of the car (e.g., Nissan, Tesla, etc.)',
      },
      {
        name: 'type',
        type: 'string',
        title: 'Car Type',
        description: 'Type of the car (e.g., Sport, Sedan, SUV, etc.)',
      },
      {
        name: 'fuelCapacity',
        type: 'string',
        title: 'Fuel Capacity',
        description: 'Fuel capacity or battery capacity (e.g., 90L, 100kWh)',
      },
      {
        name: 'transmission',
        type: 'string',
        title: 'Transmission',
```

Verification:

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

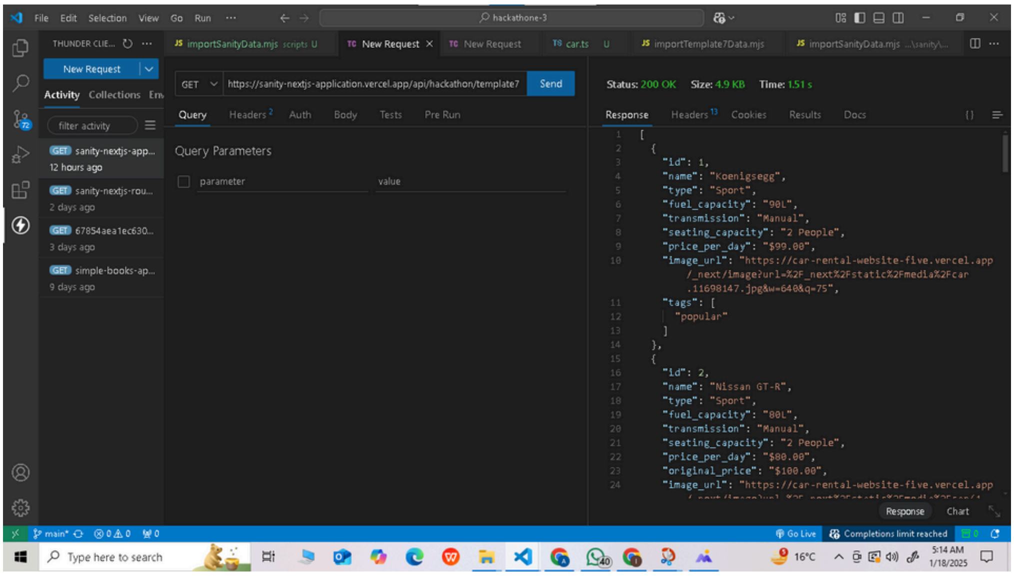
- **Frontend Integration:**
  - Used a GROQ query to fetch the data from Sanity CMS in the Hero component:



```
"use client";
import React from "react";
import Image from "next/image";
import { IoIosArrowDown } from "react-icons/io";
import { IoArrowDown, IoArrowUp,  } from "react-icons/io5";
import Cards from "./Cards";
import Link from "next/link";
import { client } from "@/sanity/lib/client";

export async function getData() {
  const fetchData = await client.fetch(
    `
    *[_type == "car"][0..3]{
            id,
            image,
            pricePerDay,
            name,
            seatingCapacity,
            fuelCapacity,
            type,
            transmission
    `
  );

  return fetchData;
}

export async function getData2() {
  const fetchData = await client.fetch(
    `
    *[_type == "car"][4..11]
    `
```

Screenshots

1. **API Calls:**
   - **(Attach screenshot of API call in Postman or browser network tab.)**

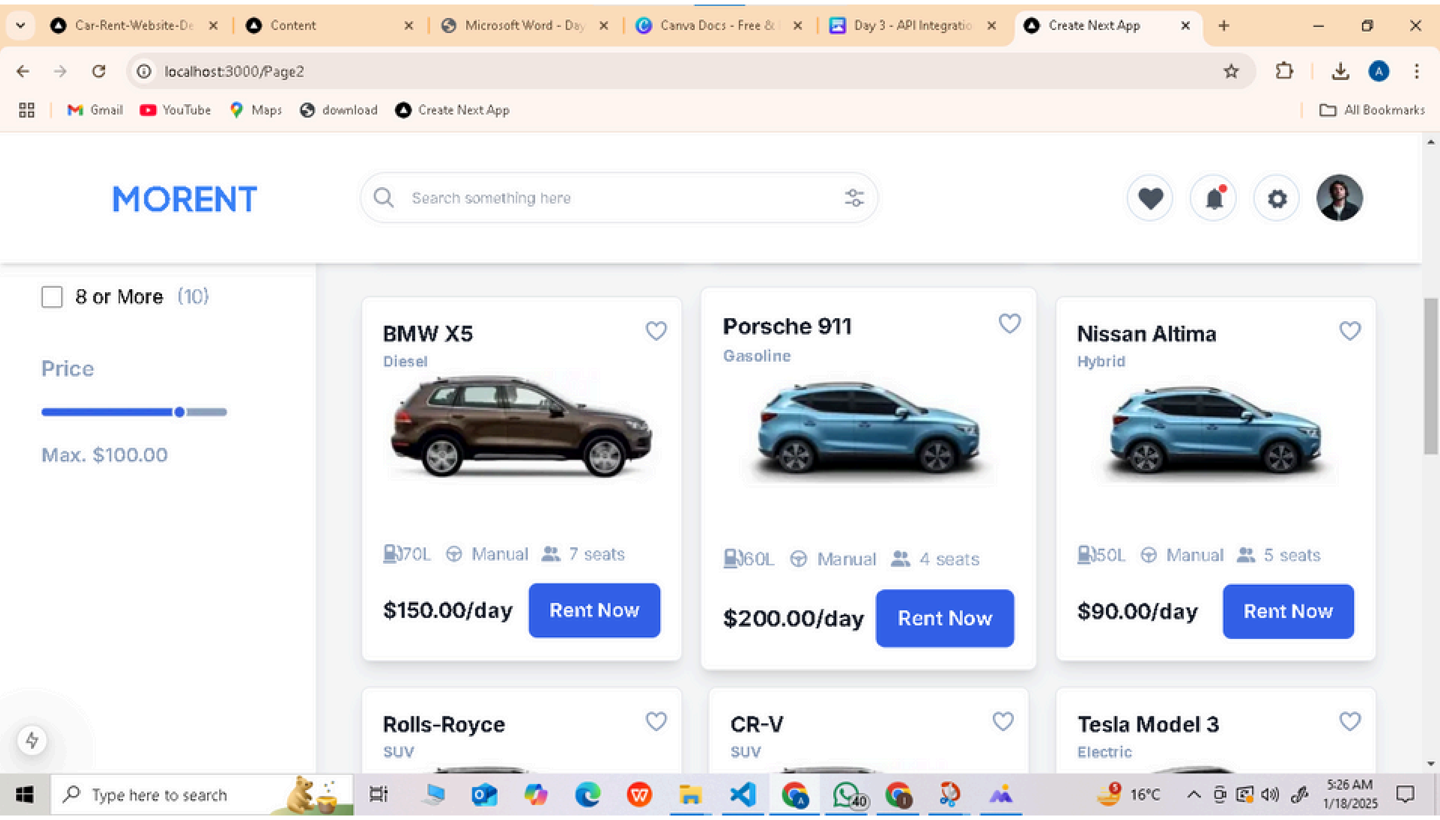

File Edit Selection View Go Run ···
hackathone-3
THUNDER CLIE...
importSanityData.mjs scripts U
New Request
New Request
car.ts U
importTemplate7Data.mjs
importSanityData.mjs ...\sanity\...
New Request
Activity Collections Env
filter activity
GET sanity-nextjs-app... 12 hours ago
GET sanity-nextjs-rou... 2 days ago
GET 67854aea1ec630... 3 days ago
GET simple-books-ap... 9 days ago
Query Headers 2 Auth Body Tests Pre Run
Query Parameters
parameter
value
Response Headers 13 Cookies Results Docs
Response Chart
main* Go Live Completions limit reached
Type here to search
16°C
5:14 AM
1/18/2025

GET https://sanity-nextjs-application.vercel.app/api/hackathon/template7 Send

Status: 200 OK Size: 4.9 KB Time: 1.51 s

```
[
  {
    "id": 1,
    "name": "Koenigsegg",
    "type": "Sport",
    "fuel_capacity": "90L",
    "transmission": "Manual",
    "seating_capacity": "2 People",
    "price_per_day": "$99.00",
    "image_url": "https://car-rental-website-five.vercel.app
      /_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fcar
      .11698147.jpg&w=640&q=75",
    "tags": [
      "popular"
    ]
  },
  {
    "id": 2,
    "name": "Nissan GT-R",
    "type": "Sport",
    "fuel_capacity": "80L",
    "transmission": "Manual",
    "seating_capacity": "2 People",
    "price_per_day": "$80.00",
    "original_price": "$100.00",
    "image_url": "https://car-rental-website-five.vercel.app
```

1. **Data Displayed in Frontend:**
   ○



□ 8 or More (10)

Price

Max. $100.00

BMW X5
Diesel
70L Manual 7 seats
$150.00/day Rent Now

Porsche 911
Gasoline
60L Manual 4 seats
$200.00/day Rent Now

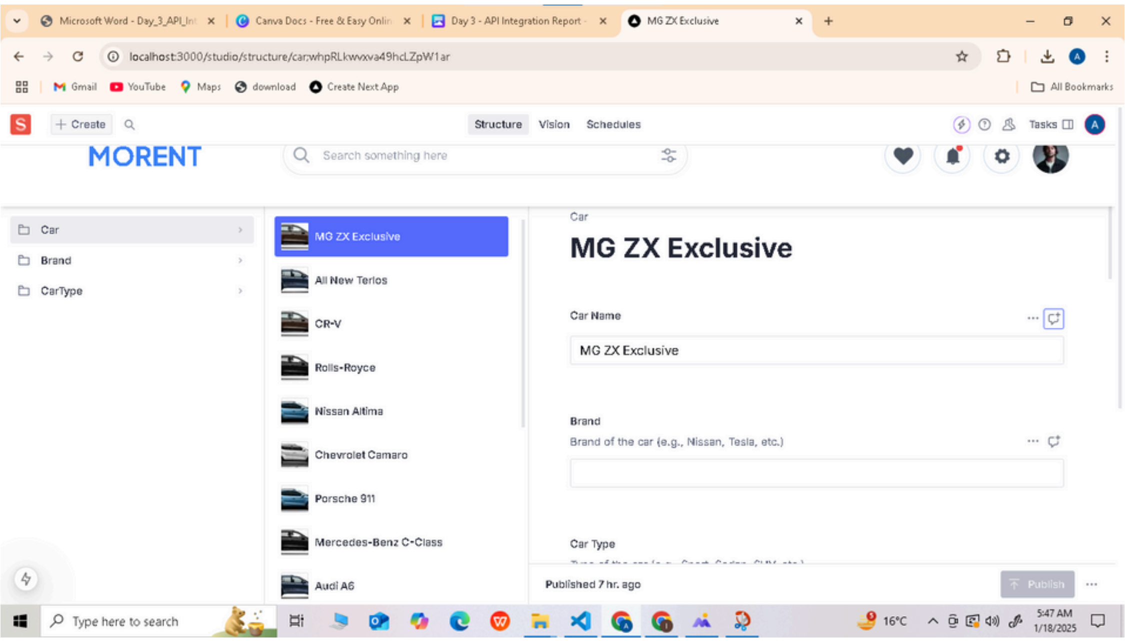
Nissan Altima
Hybrid
50L Manual 5 seats
$90.00/day Rent Now

Rolls-Royce
SUV

CR-V
SUV

Tesla Model 3
Electric

Populated Sanity CMS Fields:



Car Name

MG ZX Exclusive

Brand

Brand of the car (e.g., Nissan, Tesla, etc.)

Car Type

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.