Day 3 - API Integration Progress
Car Rental Marketplace

API Integration Workflow

1. Analyzing the Requirements:

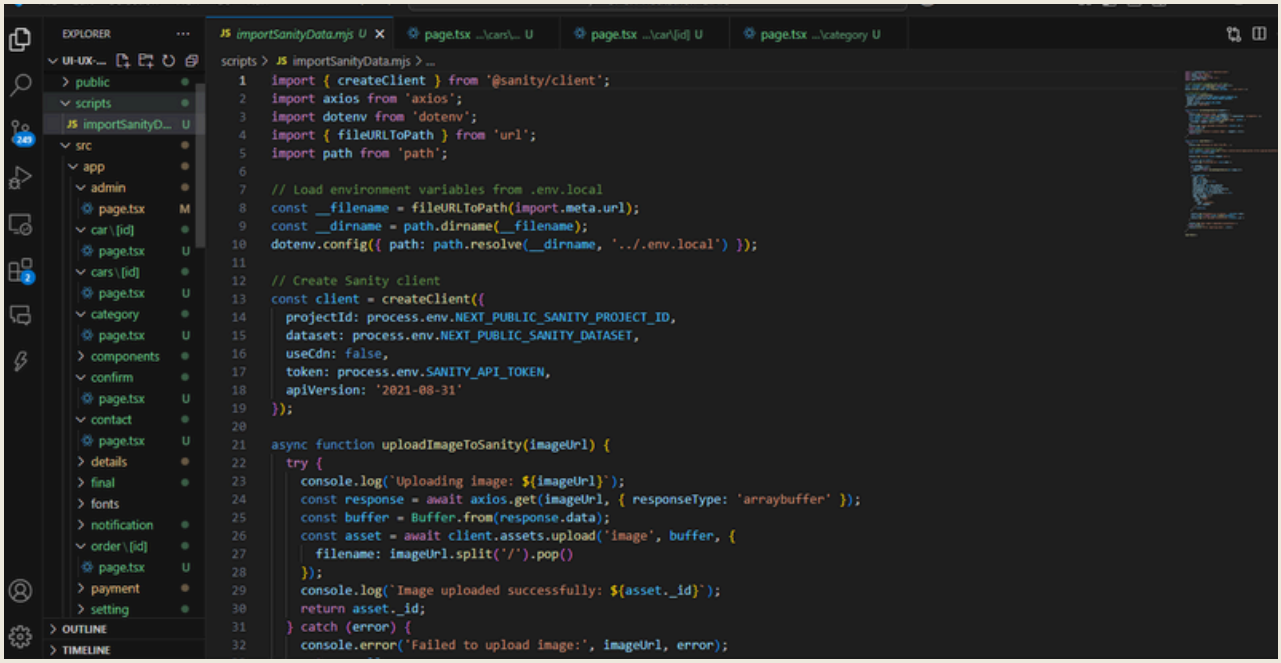
The objective was to seamlessly integrate the provided API (given by Sir Asharib Ali) into the Sanity CMS and ensure the data is correctly displayed on the frontend of our Car Rental Marketplace, built using Next.js.

2. Configuring the Development Environment:

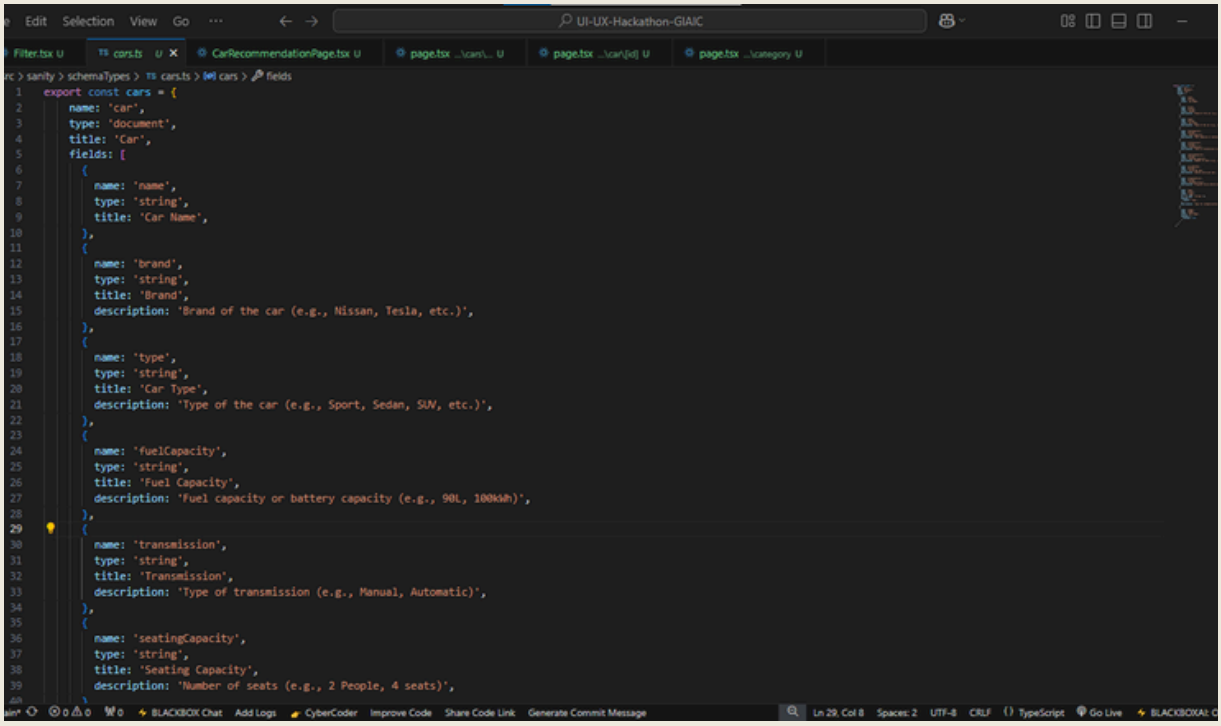
Initially, I installed Sanity in the Next.js project folder using the command:
npm install -g @sanity/cli sanity init.
Next, I added the Sanity project ID and API token to the .env file for secure configuration, like this:
SANITY_PROJECT_ID=your_project_id
SANITY_DATASET=production
SANITY_API_TOKEN=your_api_token

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.



- 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.



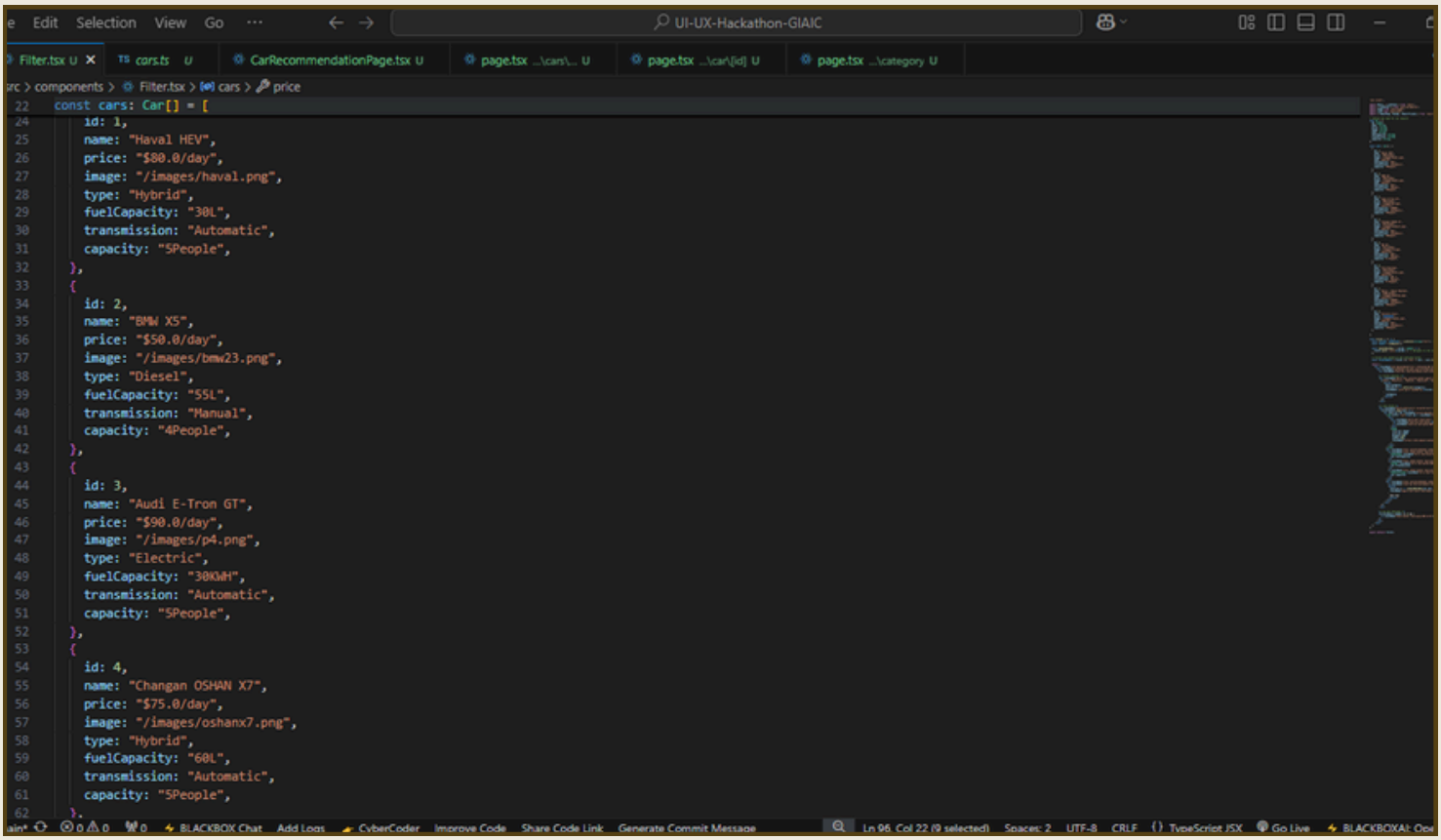
```
export const cars = {
    name: 'car',
    type: 'document',
    title: 'Car',
    fields: [
      {
        name: 'name',
        type: 'string',
        title: 'Car Name',
      },
      {
        name: 'brand',
        type: 'string',
        title: 'Brand',
        description: 'Brand of the 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',
        description: 'Type of transmission (e.g., Manual, Automatic)',
      },
      {
        name: 'seatingCapacity',
        type: 'string',
        title: 'Seating Capacity',
        description: 'Number of seats (e.g., 2 People, 4 seats)',
```

Verification:

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

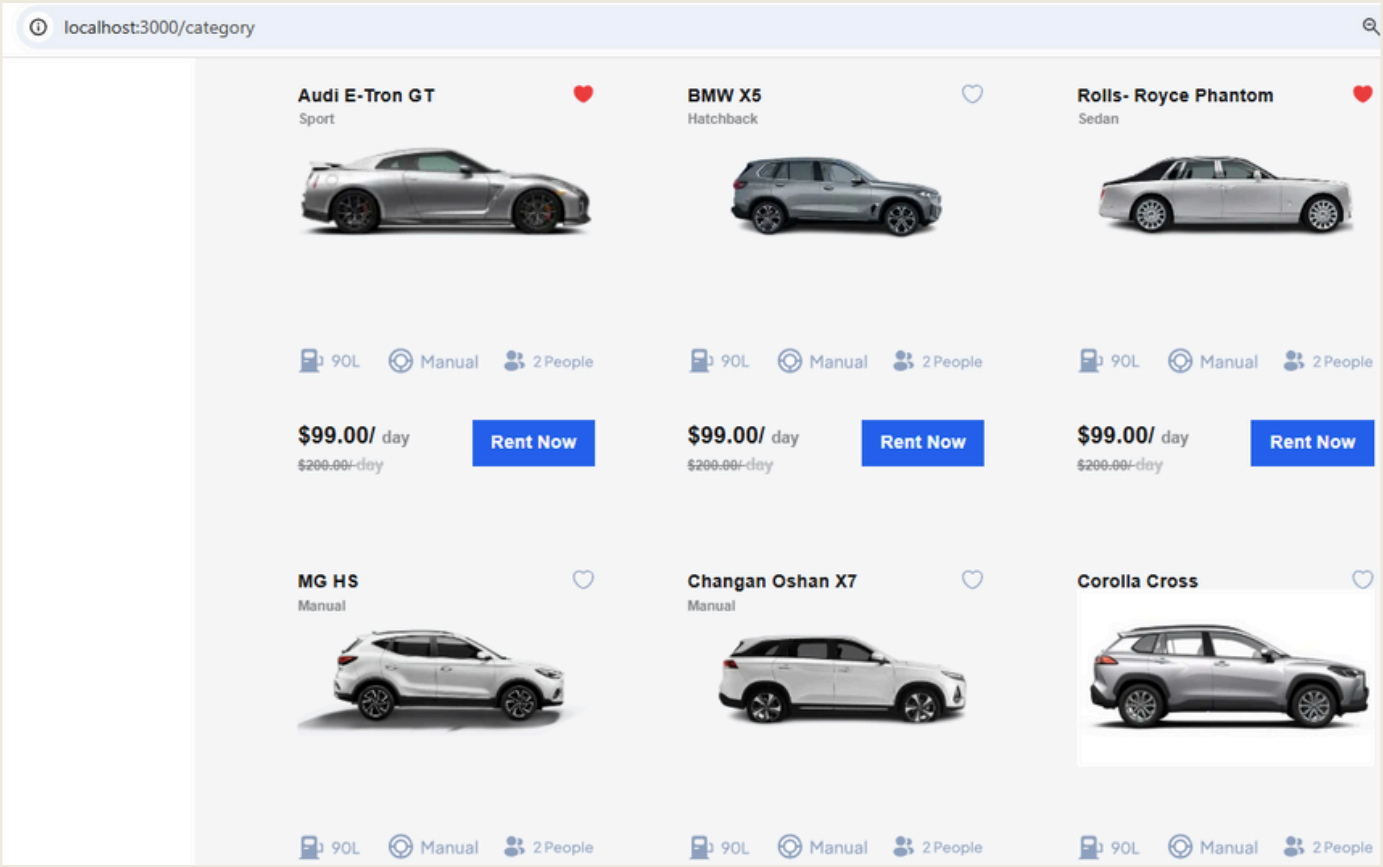
Screenshots

1. API Calls:

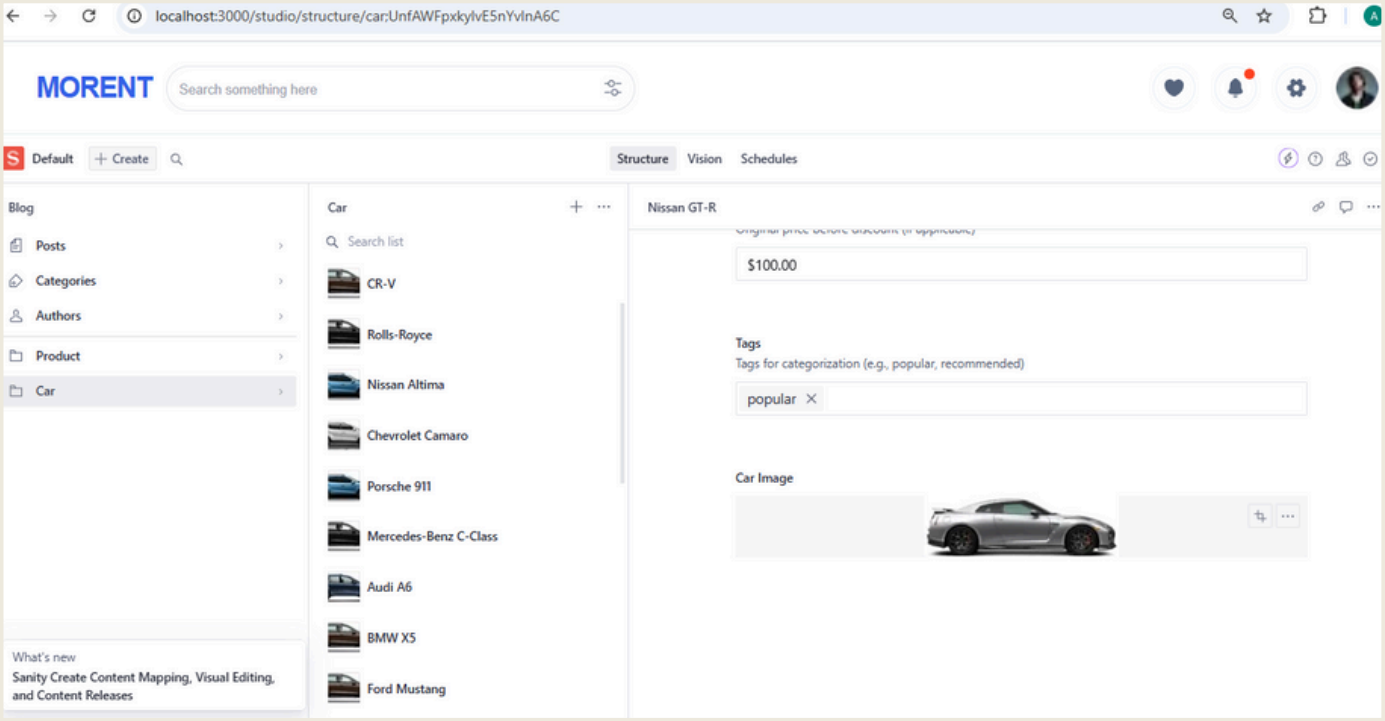


```
const cars: Car[] = [
  {
    id: 1,
    name: "Haval HEV",
    price: "$80.0/day",
    image: "/images/haval.png",
    type: "Hybrid",
    fuelCapacity: "30L",
    transmission: "Automatic",
    capacity: "5People",
  },
  {
    id: 2,
    name: "BMW X5",
    price: "$50.0/day",
    image: "/images/bmw23.png",
    type: "Diesel",
    fuelCapacity: "55L",
    transmission: "Manual",
    capacity: "4People",
  },
  {
    id: 3,
    name: "Audi E-Tron GT",
    price: "$90.0/day",
    image: "/images/p4.png",
    type: "Electric",
    fuelCapacity: "30KWH",
    transmission: "Automatic",
    capacity: "5People",
  },
  {
    id: 4,
    name: "Changan OSHAN X7",
    price: "$75.0/day",
    image: "/images/oshanx7.png",
    type: "Hybrid",
    fuelCapacity: "60L",
    transmission: "Automatic",
    capacity: "5People",
  },
```

1. Data Displayed in Frontend:



Populated Sanity CMS Fields:



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.