Day 3 - API Integration Report - Morent

API Integration Process

1. Assigned Template

For this task, I was assigned Template 7. Below are the API and migration resources provided:

- **API URL**: https://sanity-nextjs-application.vercel.app/api/hackathon/template7
- **Sanity Schema**:https://github.com/AsharibAli/sanitynextjs/blob/main/sanity/schemaTypes/cars.ts
- **Migration Script**:https://github.com/AsharibAli/sanity-nextjs/blob/main/scripts/importTemplate7Data.mjs

2. Schema Adjustments

To ensure compatibility with the provided API data, I validated and adjusted the schema in my Sanity CMS as follows:

- **API Field**: `image_url`
  - **Mapped to Schema Field**: `image`
- **API Field**: `price_per_day`
  - **Mapped to Schema Field**: `price`
- **API Field**: `type`
  - **Mapped to Schema Field**: `make`

Changes were made to ensure seamless data migration and display in frontend components.

3. Migration Steps

I followed the steps below to migrate data from the API into my Sanity CMS:

1. **Understanding the Data Structure**:
   - Reviewed the API response using browser developer tools.
   - Identified key fields such as `image_url`, `price_per_day`, and `type`.
2. **Using the Provided Migration Script**:
   - Downloaded the script from the provided link.
   - Modified the script to align with my Sanity CMS schema adjustments.
   - Executed the script to fetch data from the API and populate my Sanity CMS.
3. **Validating Migrated Data**:
   - Checked the Sanity CMS dashboard to verify that data was imported correctly.
   - Ensured that all required fields were populated.

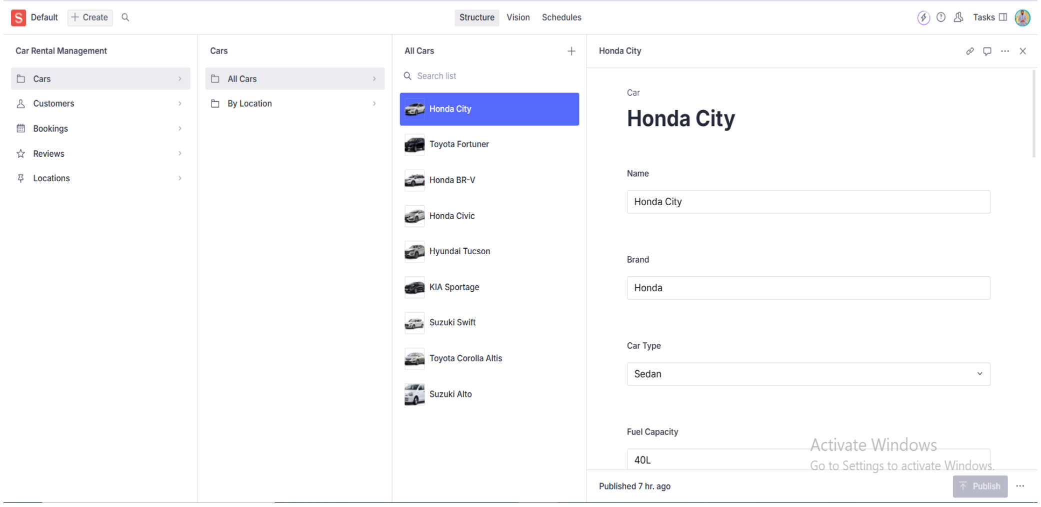
---

4. Frontend Integration

1. **Utility Functions**:
   - Created reusable utility functions to fetch data from the Sanity CMS.
   - Ensured proper error handling and data transformations.
2. **Rendering Data in Components**:
   - Utilized React components to display fetched data on the frontend.
3. **Testing**:
   - Verified API calls.
   - Confirmed accurate rendering of data on the frontend.

Screenshots

1. Sanity CMS Data:



Car Rental Management

- Cars
- Customers
- Bookings
- Reviews
- Locations

Cars

- All Cars
- By Location

All Cars

Search list

- Honda City
- Toyota Fortuner
- Honda BR-V
- Honda Civic
- Hyundai Tucson
- KIA Sportage
- Suzuki Swift
- Toyota Corolla Altis
- Suzuki Alto

Honda City

Car

Honda City

Name: Honda City

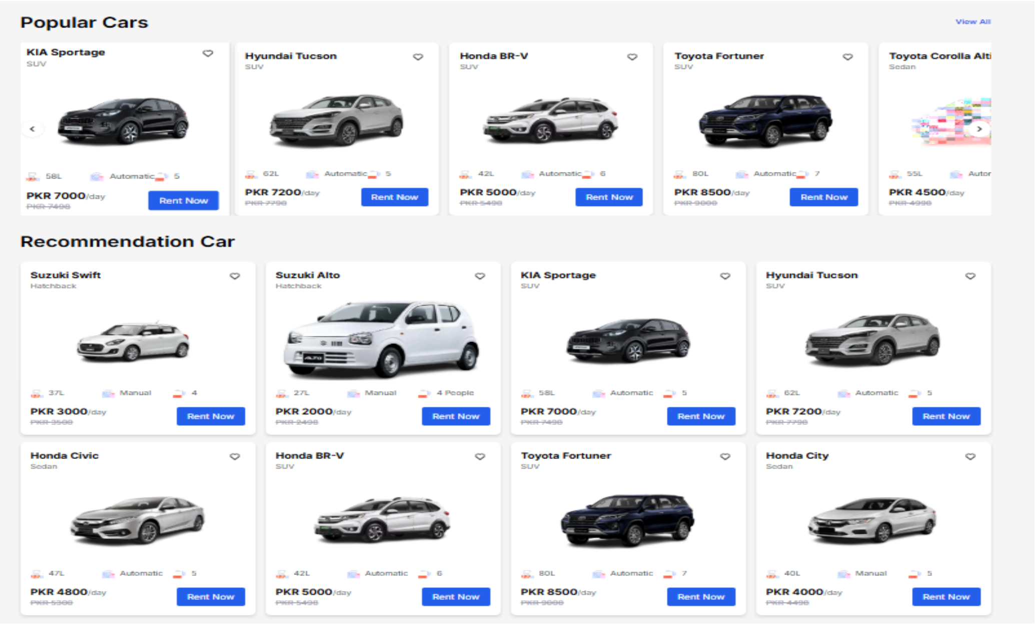
Brand: Honda

Car Type: Sedan

Fuel Capacity: 40L

Published 7 hr. ago

2. Frontend Component:



Popular Cars

KIA Sportage
SUV
58L Automatic 5
PKR 7000/day ~~PKR 7498~~
Rent Now

Hyundai Tucson
SUV
62L Automatic 5
PKR 7200/day ~~PKR 7798~~
Rent Now

Honda BR-V
SUV
42L Automatic 6
PKR 5000/day ~~PKR 5498~~
Rent Now

Toyota Fortuner
SUV
80L Automatic 7
PKR 8500/day ~~PKR 9000~~
Rent Now

Toyota Corolla Alti
Sedan
55L Autor
PKR 4500/day ~~PKR 4998~~

Recommendation Car

Suzuki Swift
Hatchback
37L Manual 4
PKR 3000/day ~~PKR 3500~~
Rent Now

Suzuki Alto
Hatchback
27L Manual 4 People
PKR 2000/day ~~PKR 2498~~
Rent Now

KIA Sportage
SUV
58L Automatic 5
PKR 7000/day ~~PKR 7498~~
Rent Now

Hyundai Tucson
SUV
62L Automatic 5
PKR 7200/day ~~PKR 7798~~
Rent Now

Honda Civic
Sedan
47L Automatic 5
PKR 4800/day ~~PKR 5300~~
Rent Now

Honda BR-V
SUV
42L Automatic 6
PKR 5000/day ~~PKR 5498~~
Rent Now

Toyota Fortuner
SUV
80L Automatic 7
PKR 8500/day ~~PKR 9000~~
Rent Now

Honda City
Sedan
40L Manual 5
PKR 4000/day ~~PKR 4498~~
Rent Now

Validation and Best Practices

1. **Schema Validation**:
    - Verified field mappings and data types.
2. **Error Handling**:
    - Implemented fallback data and user-friendly error messages.
3. **Security**:
    - Stored sensitive data (e.g., sanity token id) in `.env` files.
4. **Testing**:
    - Thoroughly tested API calls and data rendering for consistency.

Conclusion

This task provided hands-on experience with API integration and data migration. By utilizing Template 7, I successfully:

- Migrated API data into Sanity CMS using the provided script.
- Integrated the backend data with Next components for a functional frontend.

These skills are essential for building scalable and dynamic applications, enhancing my proficiency in modern development workflows.