

# DAY 4

## DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR MY CAR RENTAL WEB MARKETPLACE

### Objective:

The aim of this project was to build dynamic frontend components for a marketplace application. The focus was multiple components: **Product Listing, Header, Footer, Add to Cart, Review, CategoryTag, Location Date and Time picker, stripe Integration** and **Dynamic Routing**. These components ensure a seamless user experience while interacting with the marketplace platform.

---

### Components Implemented :

- **Objective:** Display products dynamically in a grid layout fetched from the data source.
- **Features Implemented:**
  - Rendered product details including:
    - Product Name
    - Price
    - Image
    - Fuel
    - Category
    - Capacity

- Grid layout with cards for each product.
- Data fetched dynamically from Sanity CMS or API endpoints.

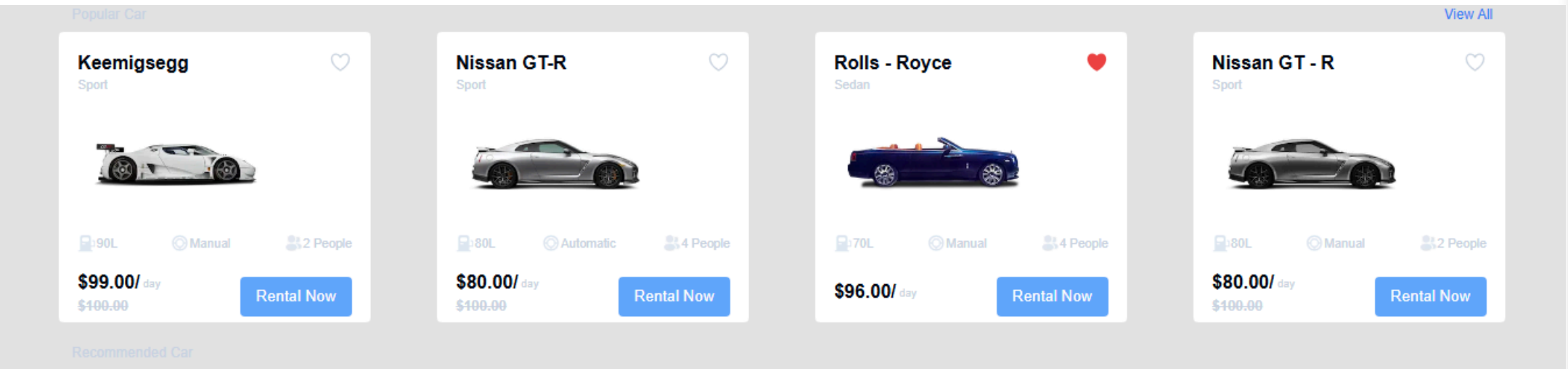
- **Challenges Faced:**

- Managing data loading states.
- Handling errors during API calls.

- **Solutions:**

- Implemented loading indicators and error handling logic.

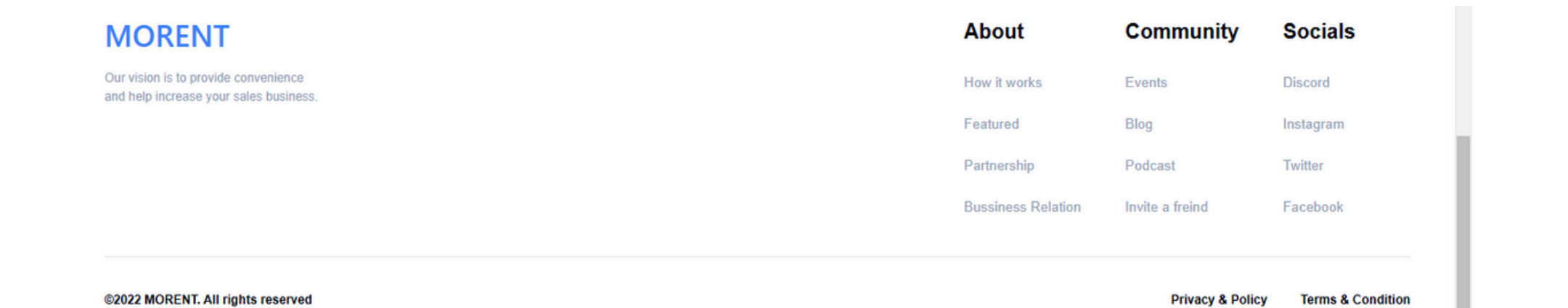
# Product Listing:



# Header:

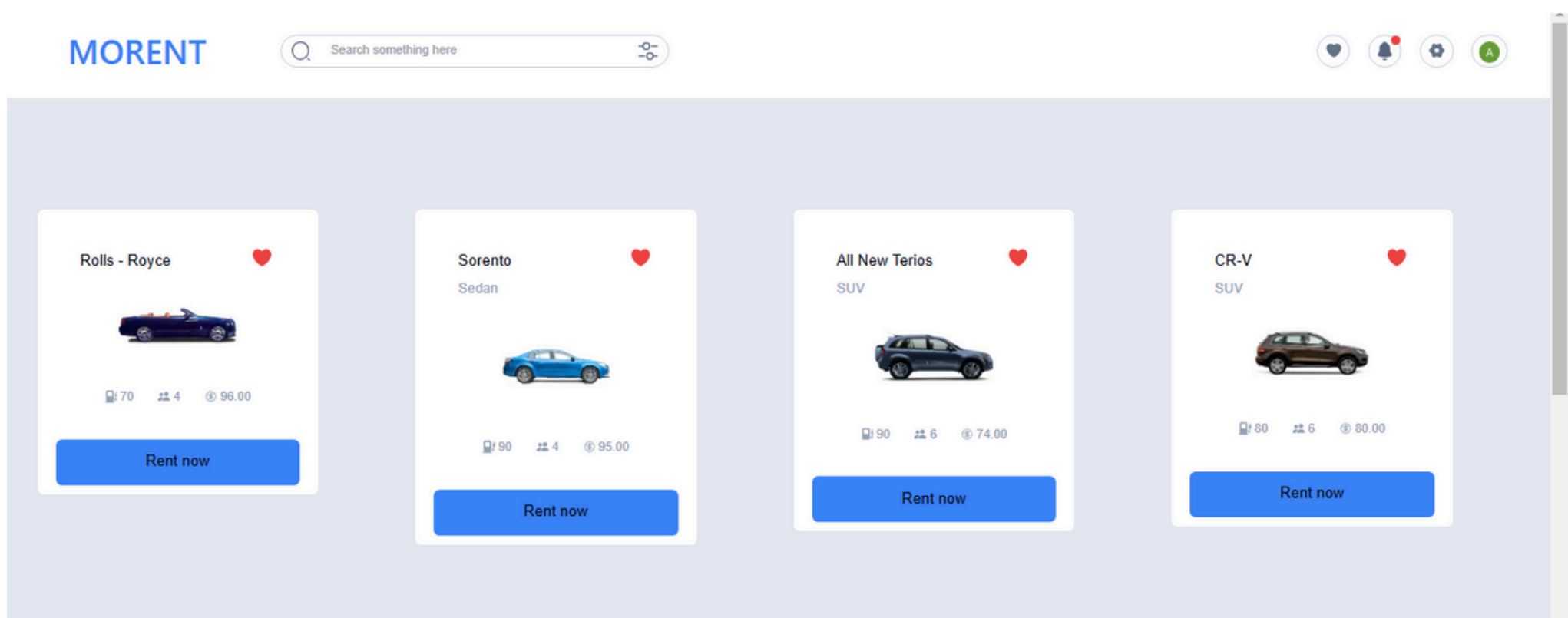


# Footer:



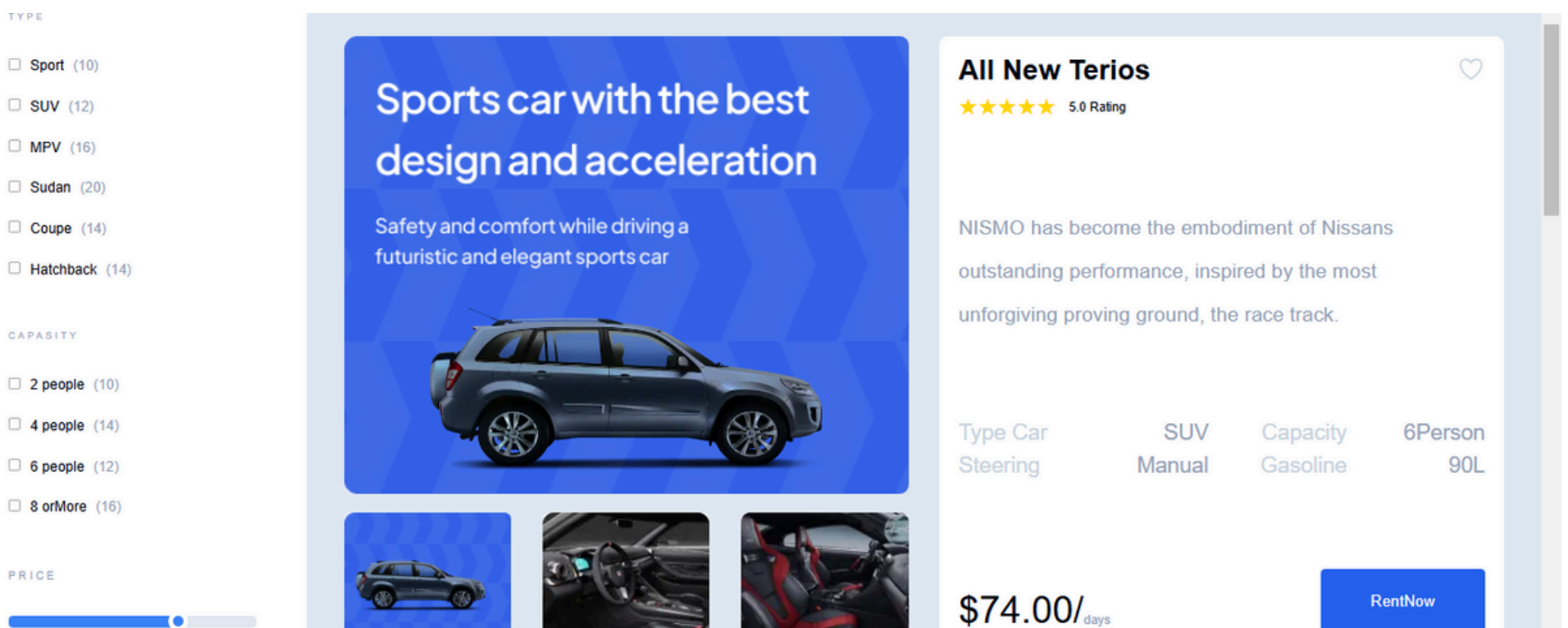
# Add To Cart:

- **Objective:** Allow users to add products to a cart and manage cart items dynamically.
- **Features Implemented:**
  - User can add Cars to cart | favorite.
  - User can remove Cars from cart.
  - The Cart Icon could change at least one car added.
  - Used state management (React Context or useState) to track cart items.
- **Challenges Faced:**
  - Managing cart updates across components.
- **Solutions:**
  - Used a global state management solution (React Context) to ensure.
  - consistent cart updates.

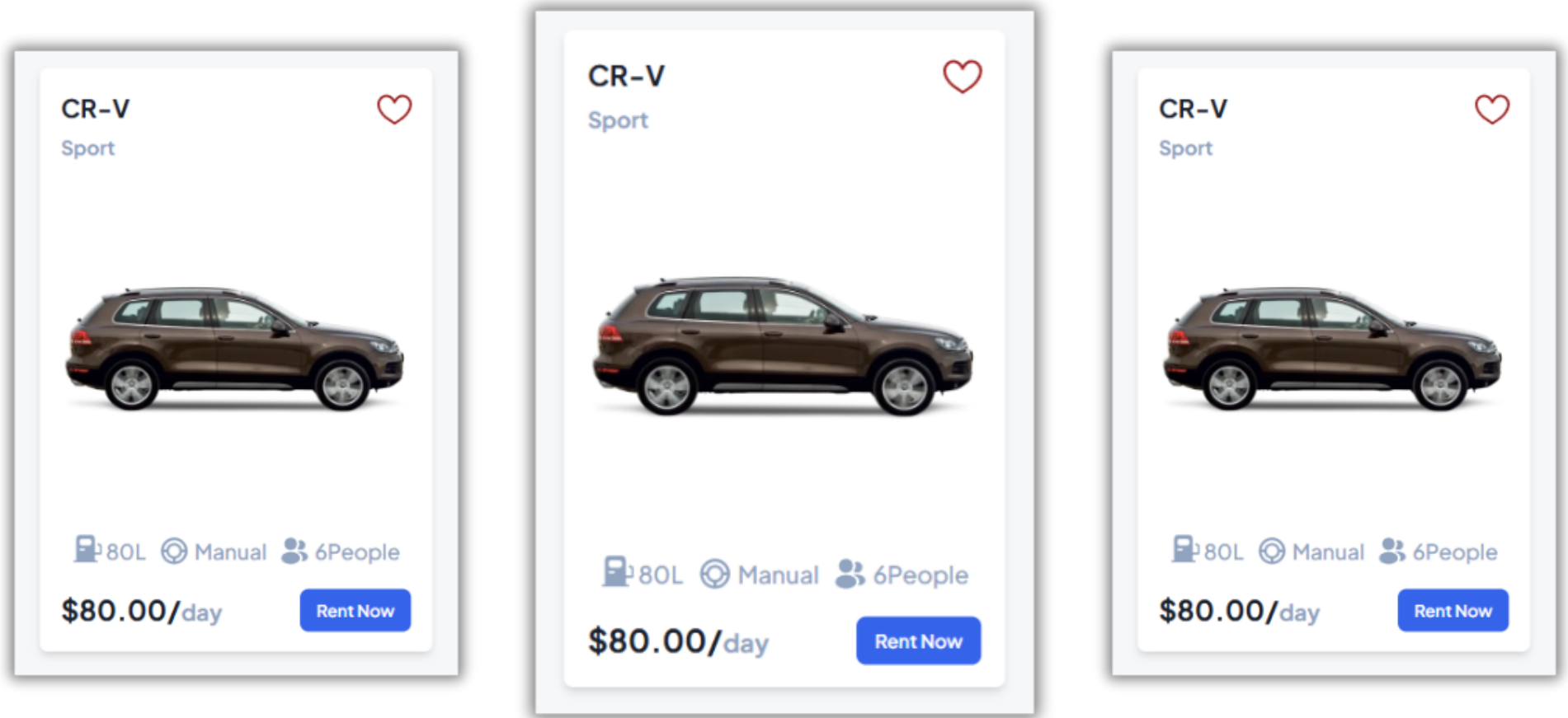


# Dynamic Routing:

- **Objective:** Enable dynamic product detail pages for individual products.
- **Features Implemented:**
  - Utilized Next.js dynamic routing for individual product pages.
  - Rendered product-specific details such as:
    - Product Description
    - Available Live Rating
    - Pricing Details
    - Capacity
  - Integrated proper fallback handling for non-existent product IDs
- **Challenges Faced:**
  - Managing non-existent routes.
- **Solutions:**
  - Handle and Catch errors.
  - No Internet message while internet would not.



# The Eye catching Product Card:



# Location, Date and Time Picker:

- **Objective:** Enable Pickup and DropOff Section
- **Features Implemented:**
  - Introduced Location selector
  - Facilities:
    - Auto set current time.
    - Auto set current date.
    - User can change Location and data with swap button.
    - The Pickup date must be lower than Dropoff.

Pick - Up

Location

Karachi

Date

02/07/2025

Time

10:58 PM

Drop - Off

Location

Hyderabad

Date

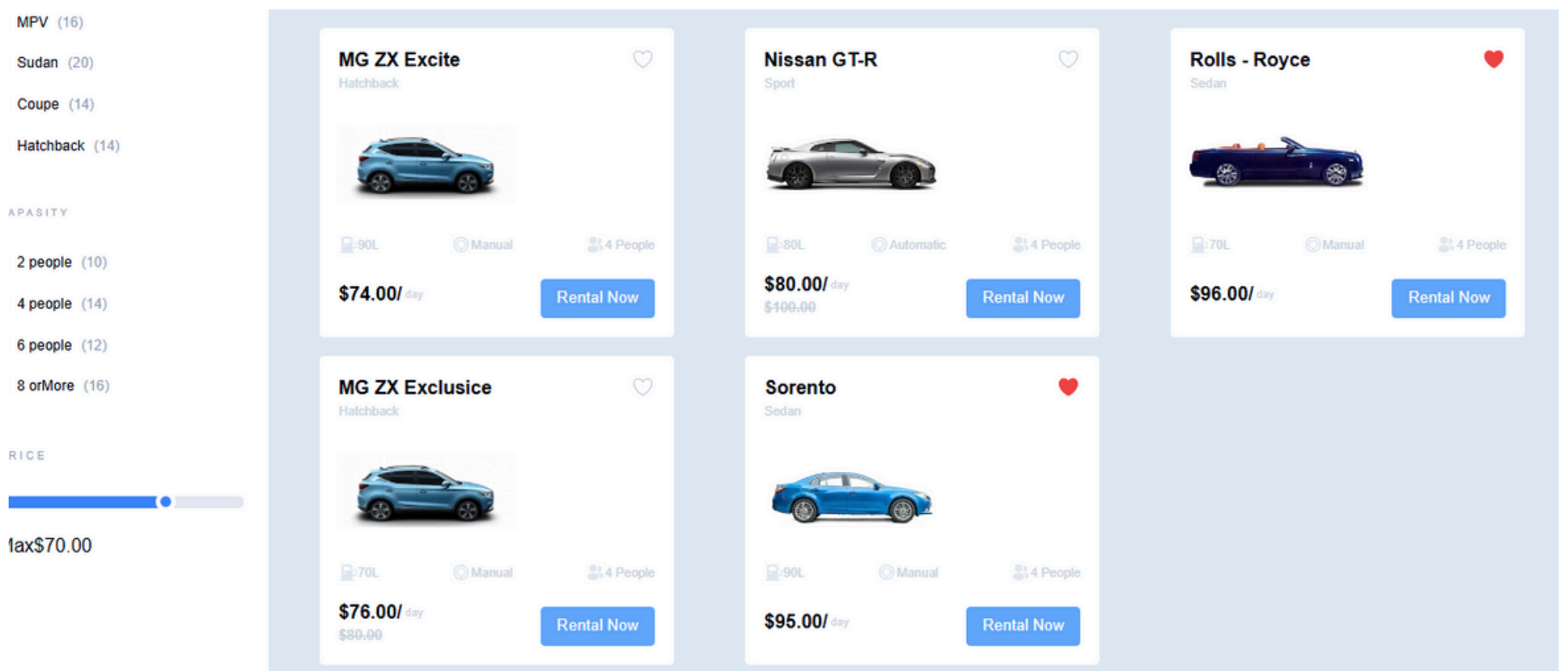
02/08/2025

Time

10:58 PM

# Category Tag:

- **Objective:** Here are multiple choices in Category Tag
- **Features Implemented:**
  - Select any type to see it's data,
  - Select any capacity checkbox to see it's data.
  - And handle Price Selector range to filter Cars by all types
- **Challenges Faced:**
  - Managing all data to multiple pages.
- **Solutions:**
  - Used a global state management solution (React Context) to ensure.



# Stripe Payment Integration:

Completely secure Payment Receiving with your own User Interface.

←

→

↺

📄

localhost:3000/billing/mg-zx-exclusice

Payment Method

Please enter your payment method

Step 3 of 4

🇺🇸

Card

📄

Amazon Pay

💰

Cash App Pay

🔒

Secure, 1-click checkout with Link

▼

Card number

1234 1234 1234 1234

VISA

MasterCard

Amex

Discover

Expiration date

MM / YY

Security code

CVC

📄

Country

Pakistan

▼

Confirmation

We are getting to the end,just few click and your rental is ready!

Step 4 of 4

☐ I agree with sending an Marketing and newsletter emails. No spam, promissed!

☐ I agree with our terms and conditions and privacy policy.

Rent Now

1. **Key Learning Outcomes:** Dynamic Data Handling:  
Successfully fetched and displayed data from external APIs or CMS.
  2. **State Management:** Applied effective state management techniques to handle cart functionality.
  3. **Responsive Design:** Ensured that components are mobile-friendly and adhere to UI/UX best practices.
  4. **Dynamic Routing:** Learned how to utilize Next.js routing effectively for a seamless user experience.
- 

## Challenges and Resolutions:

Challenge	Resolution
<ul style="list-style-type: none"><li>• API errors during fetching</li><li>• Product Listing</li><li>• cart updates consistency</li><li>• Dynamic route handling</li><li>• Category Tag for filter Cars</li><li>• Location, Date and Time Selector</li><li>• Stripe Integration</li><li>• UI/UX Responsive and attractive</li></ul>	<ul style="list-style-type: none"><li>• Added error boundary and retry mechanisms</li><li>• Product Listing</li><li>• Used React Context for global state</li><li>• Implemented fallback logic</li><li>• Dropoff date will not grater than Pickup</li><li>• Stripe will not accept payment without checked our terms and conditions.</li></ul>

---