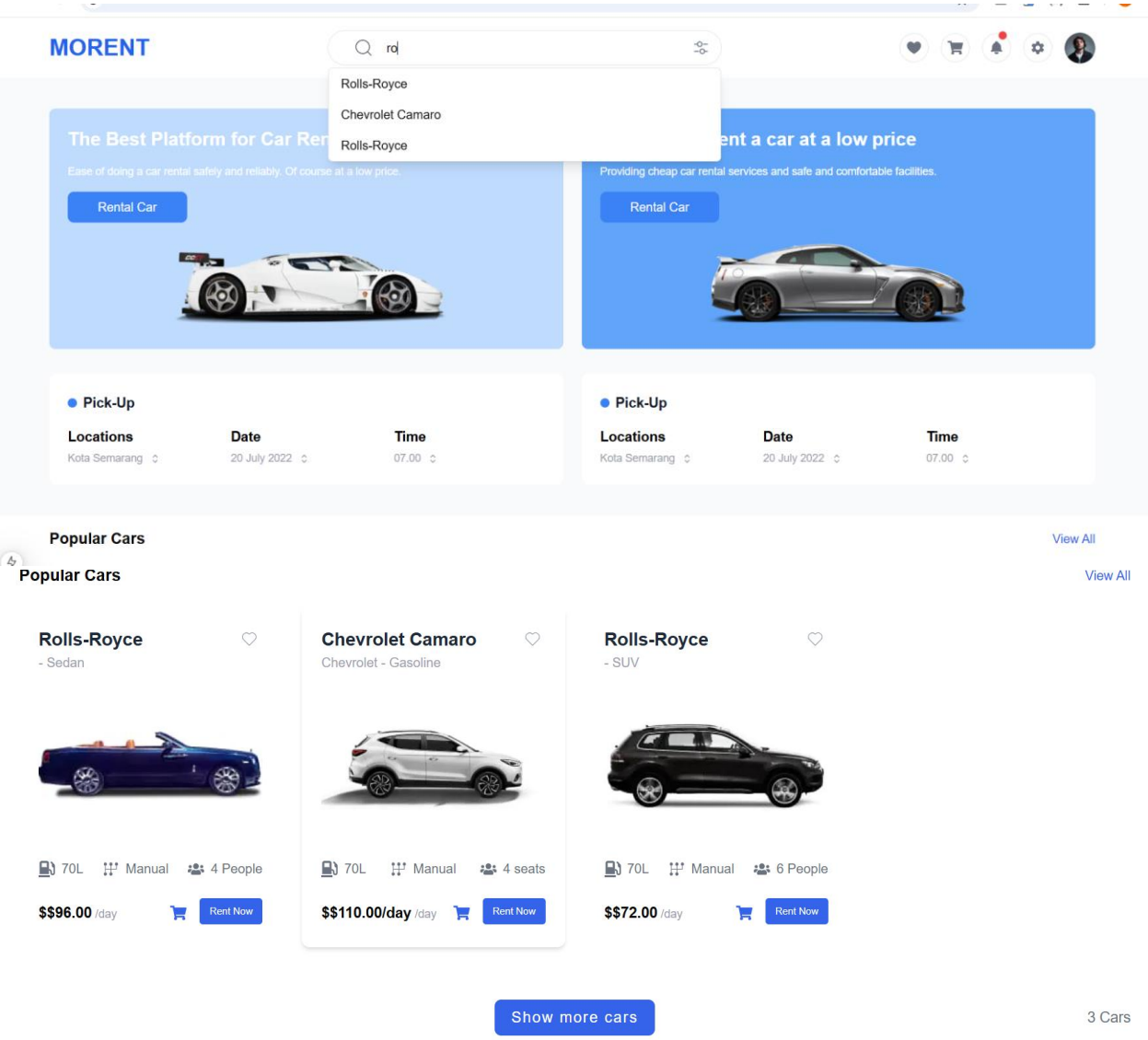
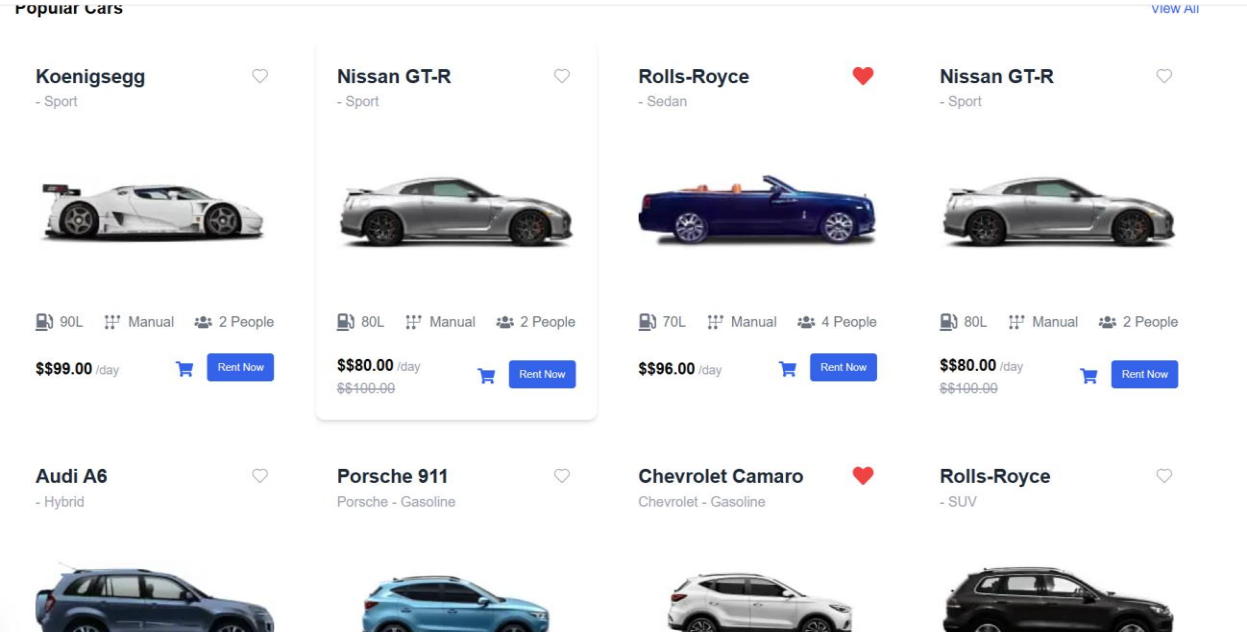
🚀 Hackathon 3 - Task 4: Building Dynamic Frontend Components! 🚀🚀
Hackathon 3 - Task 5: Building Dynamic Frontend Components! 🚀

Day 5 was all about creating dynamic and responsive frontend components for our e-commerce platform.

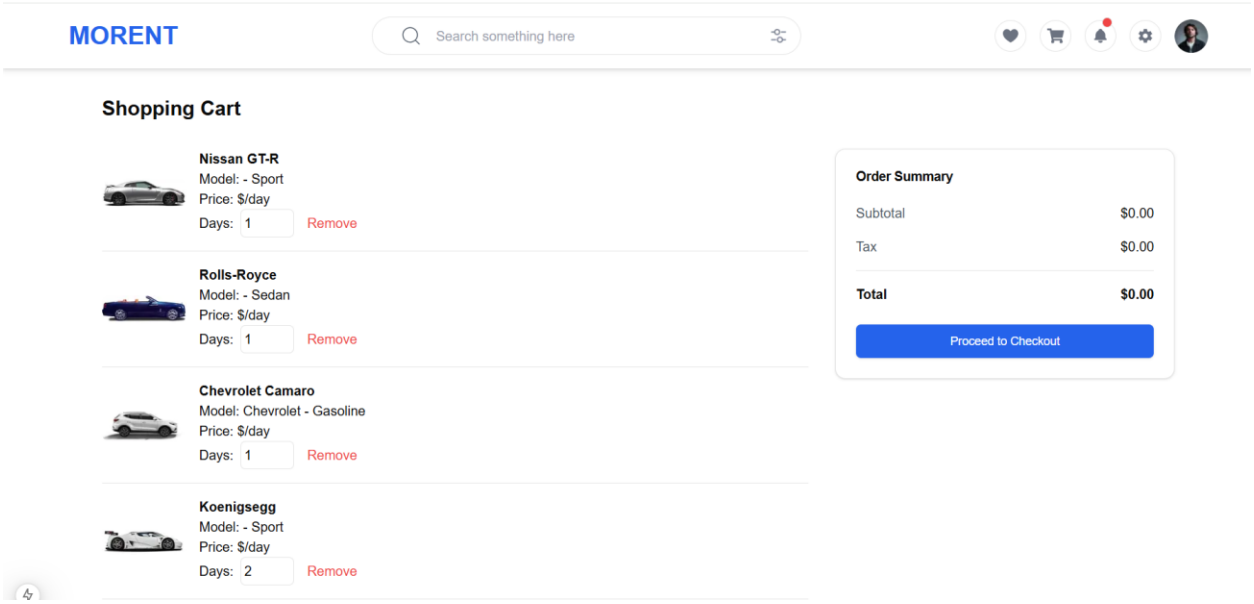


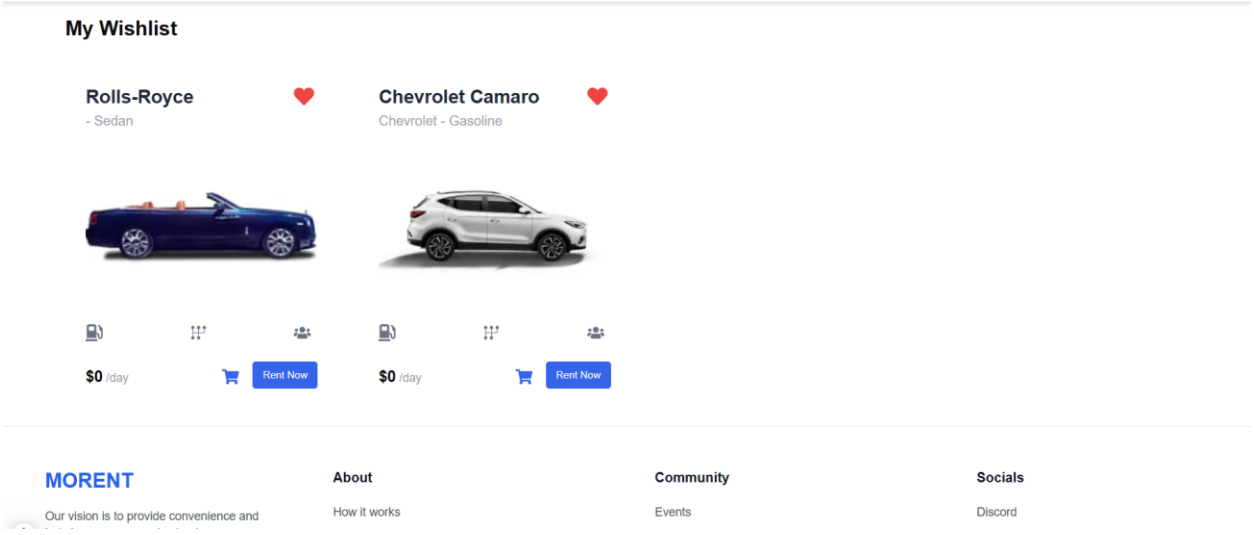


1. Add to Cart Functionality

The Add to Cart feature allows users to select cars they are interested in renting and temporarily save them for booking. Here's how it works:

- Key Features:
  - Users can add a car to the cart directly from the search results or car detail pages by clicking the "Add to Cart" button.
  - A small cart icon appears in the header with a counter showing the number of items in the cart.
  - The cart stores details such as:
    - Car name and model.
    - Rental price.
    - Pickup and drop-off dates.
    - Total rental duration and cost.
- Cart Page:
  - Users can view all selected cars on the cart page.
  - Options to:
    - Update rental dates for individual cars.
    - Remove a car from the cart.





Here's how you can document the features and functionality of your code in a structured and clear format for reference or team collaboration:

Car Rental Website: Filters and Search Functionality

This component (Filters) adds advanced filtering, search, and pagination features for a car rental website. Here's a breakdown of the key features:

---

1. Features

Search and Filters

- Type Filtering: Users can filter cars by type (e.g., Sport, SUV, Sedan, etc.).
- Seating Capacity Filtering: Users can select cars based on seating capacity (e.g., 2, 4, 6 people).
- Price Filtering: A price slider allows users to filter cars within a price range.
- Live Updates: Filters dynamically update results in real-time using React state and hooks.

Paginated Results

- Cars are fetched in batches of 10 using pagination.
- A "Show more cars" button loads additional cars dynamically, enhancing performance for large datasets.

Car Data Fetching

- Car data is fetched from a Sanity CMS backend:
    - Includes car details like name, brand, type, fuel capacity, seating capacity, price per day, and images.

Responsive Design

- A sidebar for filters is visible on larger screens (desktop and tablet).
- Filters and car grids are optimized for smaller devices.

Loading Skeletons

- While fetching data, loading placeholders (skeletons) are displayed for a better user experience.

Error Handling

- Graceful error handling is implemented:
    - Alerts users if there’s an issue fetching cars.
    - Displays a message if no cars match the filters.

---

2. Functional Highlights

Filter Logic

- Filters work independently:
    - Type filter matches car types like "SUV" or "Sedan".
    - Seating capacity filter matches options like "2 People" or "4 People".
    - Price filter converts price strings into numerical values for comparison.

Debounced Price Slider

- Price updates are debounced to improve performance:
    - Prevents excessive state updates while dragging the slider.

Pagination

- Uses a page state to track the current page.
- Loads new data from the backend when the page changes.

Dynamic UI Updates

- The car grid updates in real-time based on active filters without reloading the page.

---

3. Components Used

- ProductCard: Displays individual car details (name, price, image, etc.) in a grid format.
- Filters Sidebar: Provides options for type, seating capacity, and price filtering.

---

4. User Flow

1. Landing Page: Users land on the car rental page, where all available cars are listed.
2. Apply Filters: Users can:
    - Select car type (e.g., Sport, Sedan).
    - Choose seating capacity.
    - Adjust price range using the slider.
3. Show More: If users want to see more cars, they can click the "Show more cars" button to load additional results.
4. View Cars: Cars matching the filters appear instantly, with loading skeletons displayed while data is being fetched.

---

5. Technologies

- Frontend: React with Next.js.

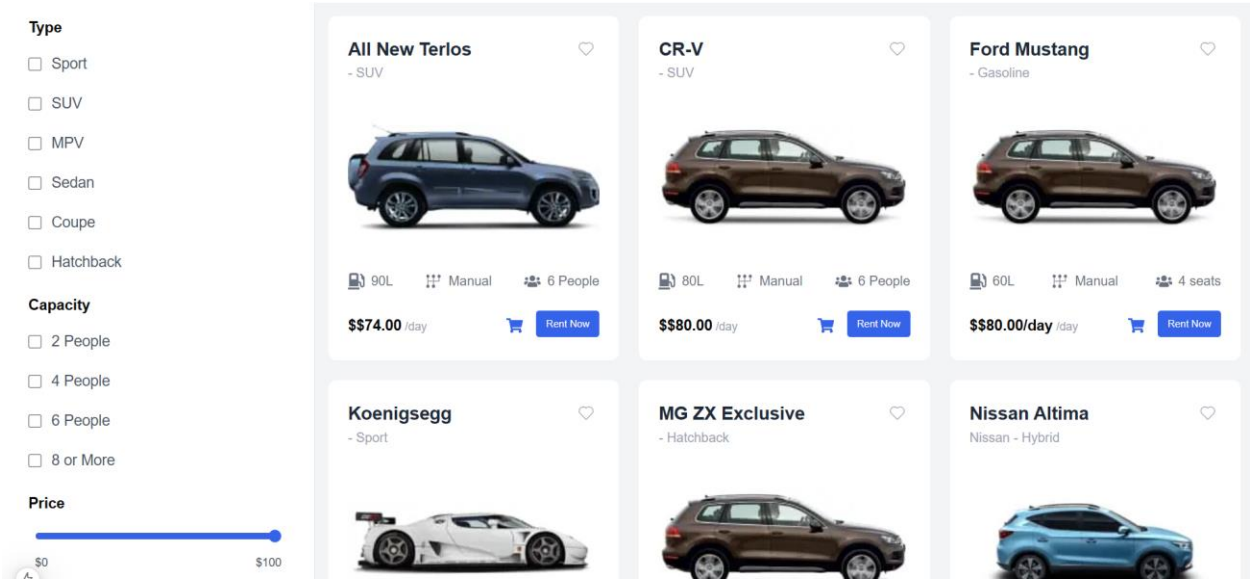
- State Management: React hooks (useState, useEffect).
- Backend: Sanity CMS for car data storage.
- Styling: Tailwind CSS for a responsive and modern design.

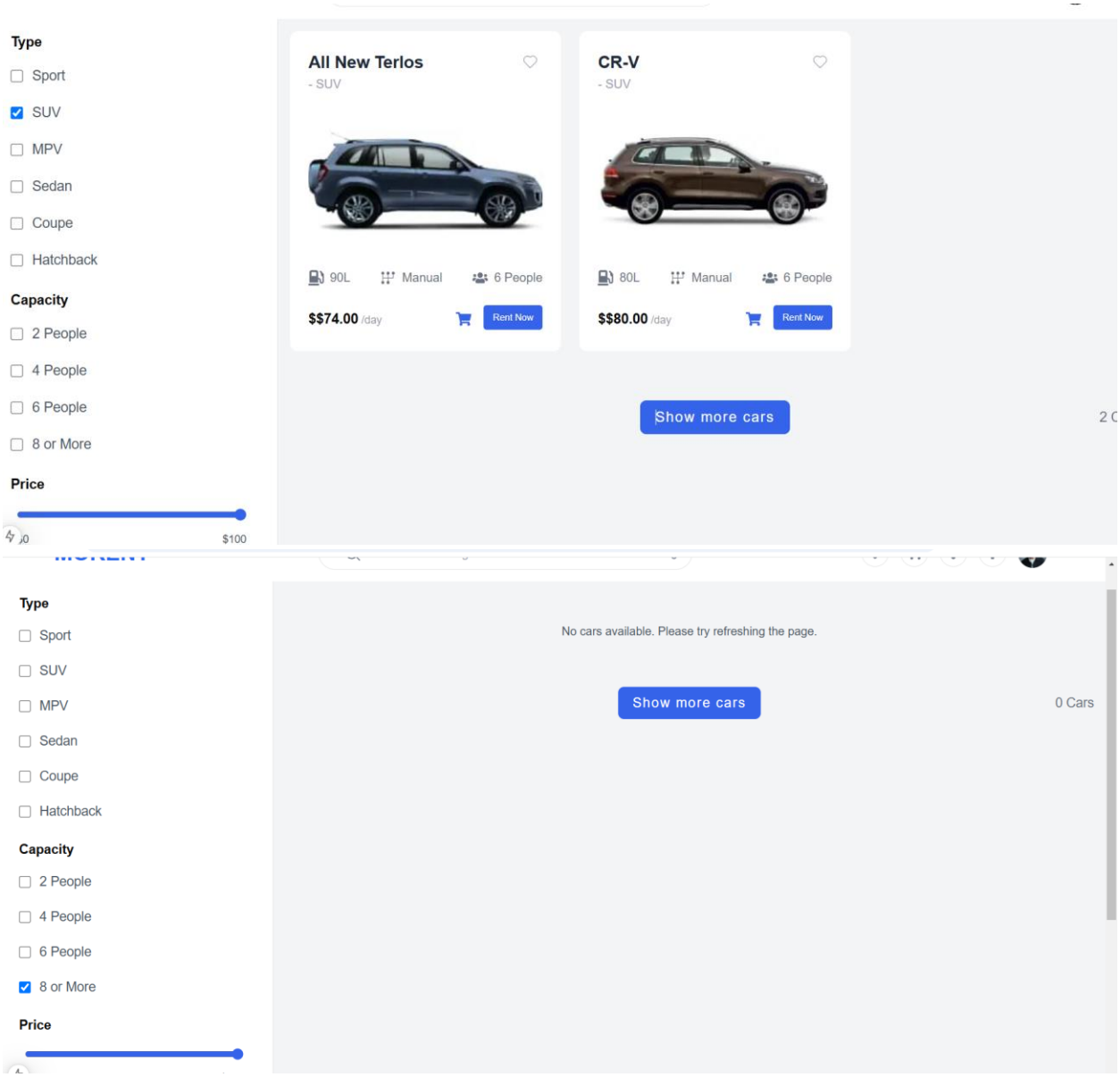
---

6. Advance Features

- Wishlist & Cart Functionality: Add buttons to save cars to a wishlist or cart.
- Advanced Search: Include a search bar to filter cars by name or brand.

- Mobile Filters: Add collapsible filters for smaller screens.

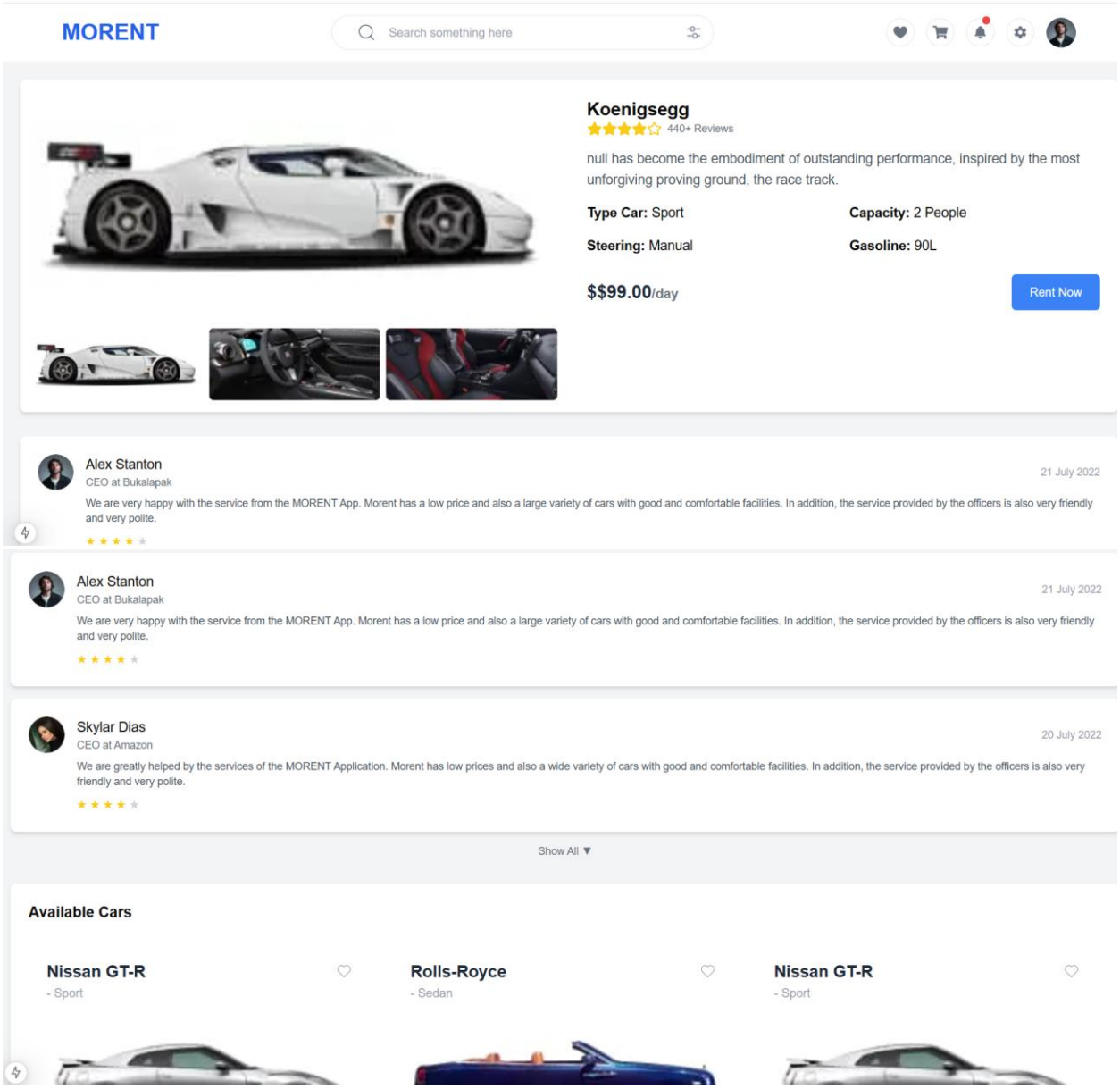




Key Features

1. Car Detail Page:
    - Displays detailed information about a selected car, including:
        - Images (main image, thumbnails, and additional images like interior/seats).
        - Specifications (fuel capacity, transmission type, seating capacity).
        - Pricing with original price (if any) and daily rental rate.
        - Reviews section with ratings and customer feedback.
    - "Rent Now" button leading to a billing page.
2. Reviews Section:

        - Static reviews for demonstration purposes, with:
            - Profile pictures, customer names, roles, dates, and detailed reviews.
            - Star ratings displayed visually.
3. Available Cars Section:
    - Displays a list of other cars available for rent, excluding the currently viewed car.
    - Uses a ProductCard component for rendering individual car details.
4. Car Grid Component:
    - Fetches all cars from Sanity and filters them based on a search query from a SearchContext.
    - Displays filtered cars dynamically using the ProductCard component.
    - Implements loading and error states.
5. Sanity Integration:
    - Fetches car details and other available cars using GROQ queries.
    - Supports dynamic routing by car ID.
6. Error Handling and State Management:
    - Handles API errors and displays appropriate messages.
    - Implements loading states to enhance the user experience.



Rental Payment Form Documentation

Introduction

This documentation provides an overview of the RentalForm and PaymentForm components. It includes details about features, functionality, and a testing plan to ensure all features work as intended.

Project Structure

Components

- RentalForm: Handles user input for billing and rental information.
- PaymentForm: Handles the payment method selection and payment input.

Assets

- bluemercedes.png: Car image used in the rental summary.
- Visa.png, PayPal.png, Bitcoin.png: Icons used for payment options.

Features

RentalForm

1. Billing Information:
    - Input fields for name, phone number, address, and city.
    - Validates required fields.
2. Rental Information:
    - Radio buttons for "Pick-Up" or "Drop-Off" options.
    - Input fields for location, date, and time.
    - Dynamic updates based on the selected option.
3. Rental Summary:
    - Displays car image and details.
    - Shows subtotal, tax, and total rental price.
    - Promo code input field with an "Apply now" button.

PaymentForm

1. Payment Method Selection:
    - Radio buttons for different payment methods: Credit Card, PayPal, Bitcoin.
    - Displays corresponding input fields dynamically based on the selected method.
2. Credit Card Payment:
    - Input fields for card number, expiration date, and CVV.
3. PayPal and Bitcoin:
    - Redirects users to external payment gateways (future enhancement).
4. Security Badge:

- Displays a shield icon to assure users of secure payment.

Testing Plan

Functional Testing

RentalForm

1. Verify all input fields in the Billing Info section accept valid data.
2. Ensure radio buttons in Rental Info switch between Pick-Up and Drop-Off options.
3. Check that location, date, and time fields display correctly for both options.
4. Test the promo code functionality and ensure it applies discounts correctly.

PaymentForm

1. Confirm all payment method radio buttons are selectable.
2. Verify the correct input fields appear for each payment method.
3. Ensure card number, expiration date, and CVV validations are enforced.
4. Test the "Apply now" button functionality.

Usability Testing

1. Check the form layout on different screen sizes (responsive design).
2. Ensure error messages appear for invalid or missing inputs.
3. Validate that the navigation to the dashboard works correctly via the "Apply now" button.

Security Testing

1. Verify that sensitive inputs, such as the credit card number, are masked.
2. Ensure secure HTTPS connections during form submission.

Future Enhancements

1. Payment Gateways: Add integration for PayPal and Bitcoin payments.
2. Dynamic Pricing: Automatically calculate total price based on selected rental dates and duration.
3. Validation: Improve client-side validation with libraries like Formik or React Hook Form.
4. Database Integration: Store rental and payment data in a backend database.

Conclusion

The RentalForm and PaymentForm components are designed to streamline the rental and payment process with an intuitive interface and robust features. The testing plan ensures reliability and a seamless user experience. Future enhancements can further improve functionality and scalability.