Name : Abdullah

Roll : 00433137

Class : Friday (7 to 10)

Day 5 - Testing, Error Handling, and Backend Integration Refinement

Prepare the marketplace for real-world deployment by testing backend integrations, optimizing performance, implementing error handling, and refining the user experience to handle customer-facing traffic.
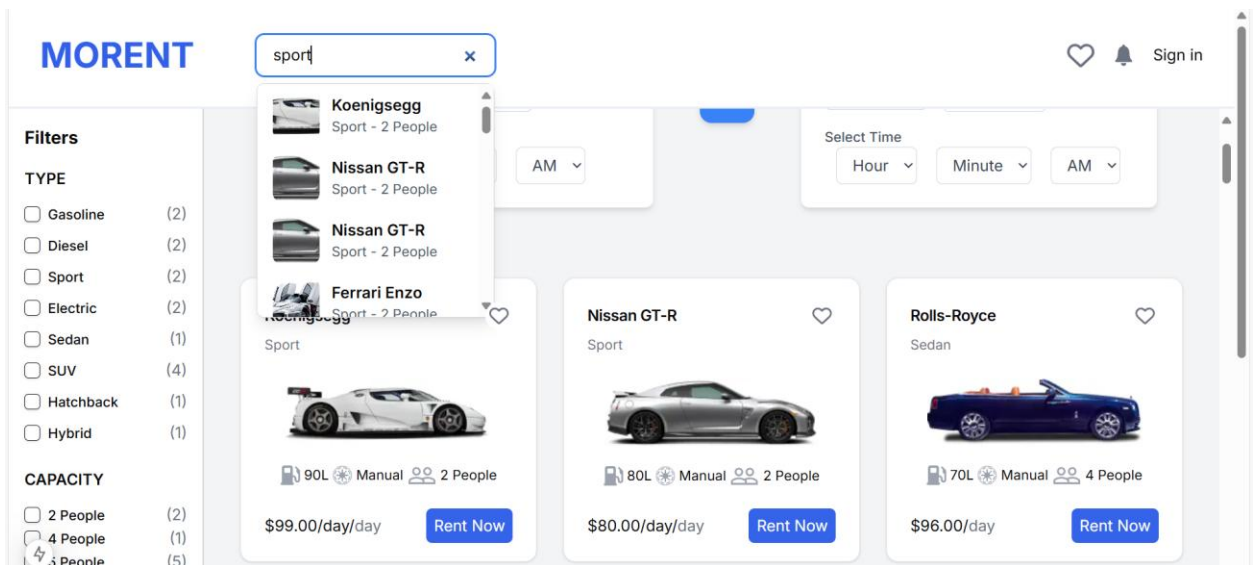
Step # 1: Functional Testing

1. Tested Features:

- **Product Listing**
  - **Test Result**: All products were displayed correctly, including names, prices, and images.
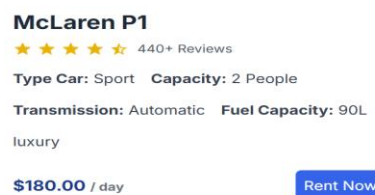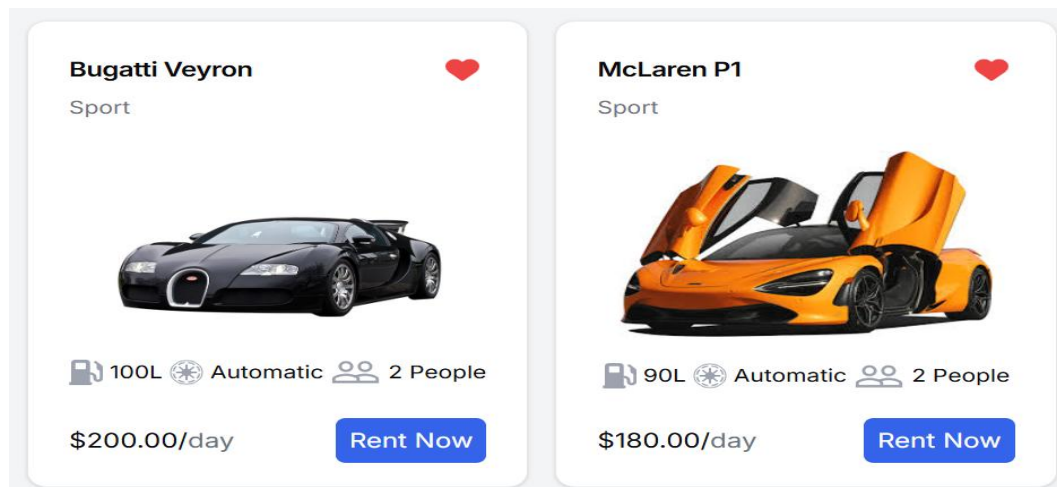 - **Screenshot**:
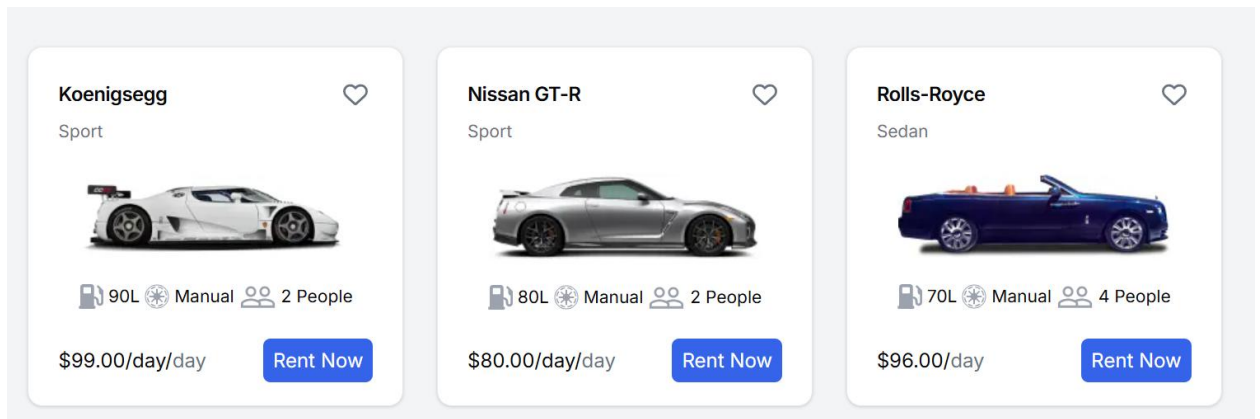
- **Filters and Search**
  - **Test Result**: Filters and search bar returned accurate product results based on selected criteria.
  - **Screenshot**:

- Dynamic Routing (Product Details)

  - Test Result: Individual product pages loaded correctly with accurate details (name, description, price).
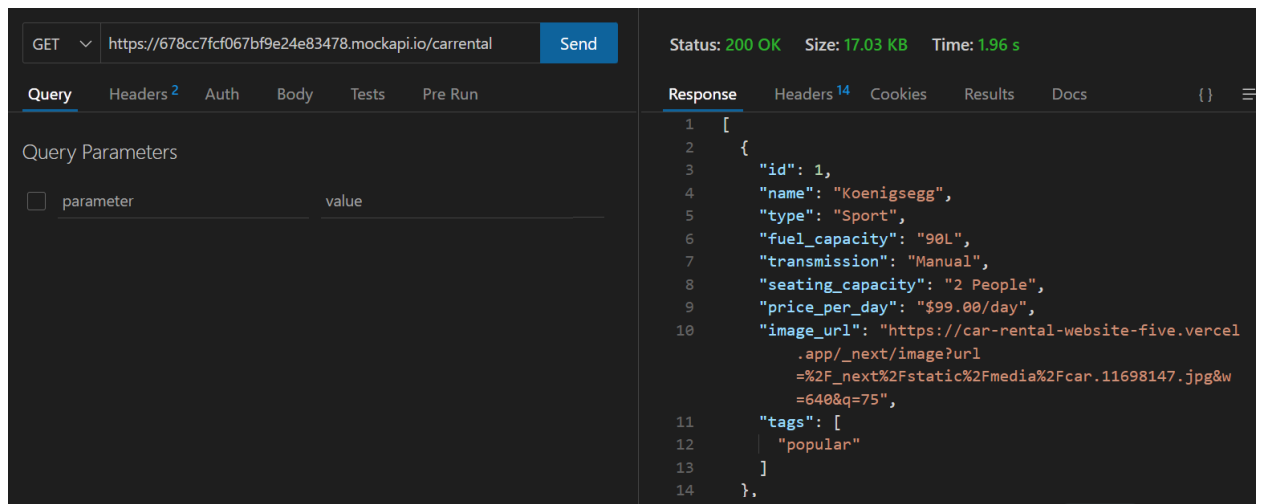 - Screenshot:

2. Testing Tools

- **Postman**
  - ○ **Test Result**: API response was successfully tested, and the expected data was returned for the GET /api/products endpoint.
  - ○ **Screenshot**:

  - ○ **Expected Outcome**: The API should return a valid response with the correct list of products, including fields like id, name, price, and image.
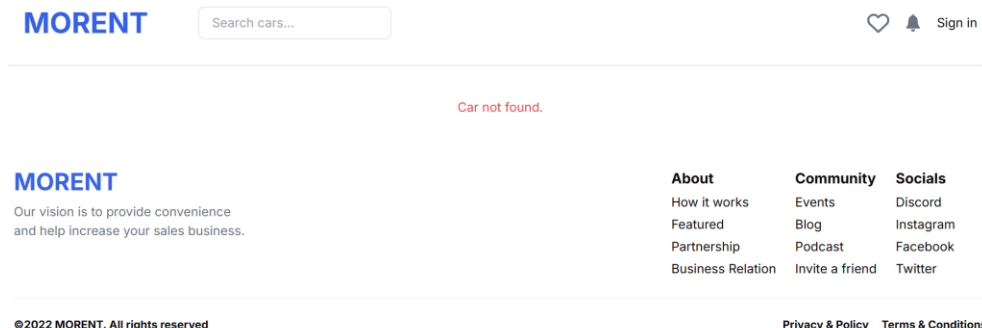
Step # 2: Error Handling

- **Add Error Messages**
  - **Test Result**: API errors were handled successfully using try-catch blocks. Appropriate error messages were shown in case of failure.
  - **Screenshot**:

```
useEffect(() => {
  const fetchCar = async () => {
    try {
      const res = await fetch(
        "https://678cc7fcf067bf9e24e83478.mockapi.io/carrental"
      );
      const allCars: Car[] = await res.json();
      const car = allCars.find((item) => item.id === parseInt(id!, 10));

      if (car) {
        setSelectedCar(car);

        // Exclude the selected car to suggest others
        const suggestions = allCars.filter((item) => item.id !== car.id).slice(0, 12);
        setSuggestedCars(suggestions);
      } else {
        setError("Car not found.");
      }
    } catch (err) {
      console.error("Error fetching car:", err);
      setError(`Failed to fetch car with ID: ${id}`);
    }
  };
```

MORENT   Search cars...   ♡ 🔔 Sign in

Car not found.

MORENT

Our vision is to provide convenience
and help increase your sales business.

About	Community	Socials
How it works	Events	Discord
Featured	Blog	Instagram
Partnership	Podcast	Facebook
Business Relation	Invite a friend	Twitter

©2022 MORENT. All rights reserved   Privacy & Policy   Terms & Conditions
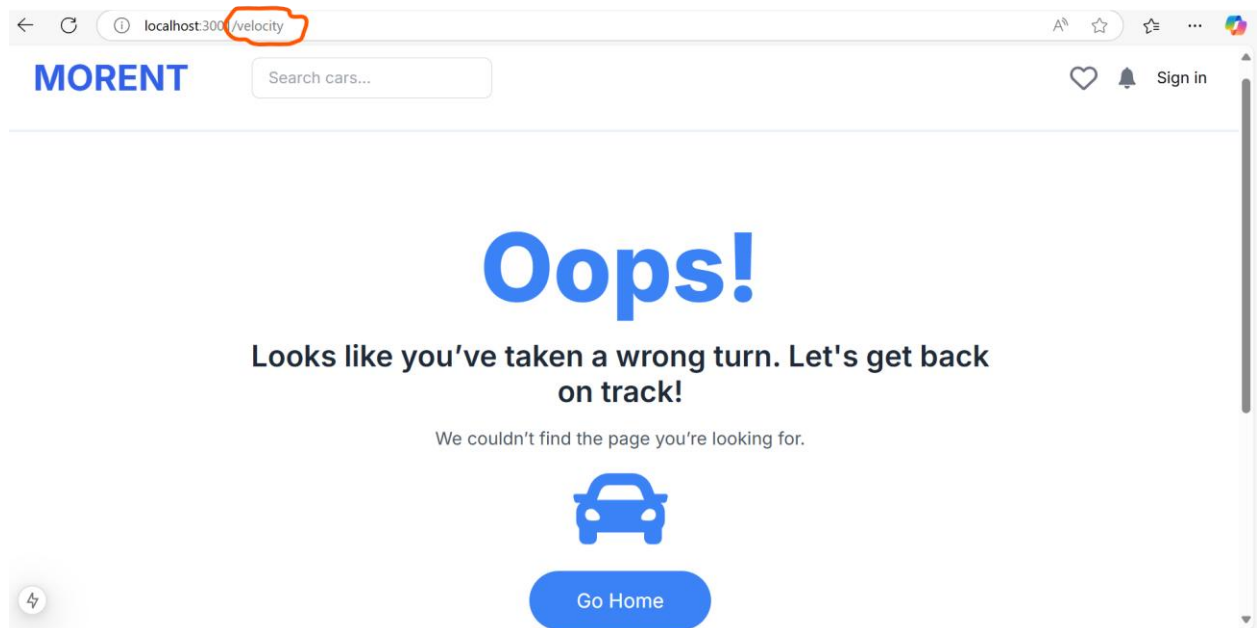
  - **Expected Outcome**: When an error occurs (like a failed API request), a meaningful error message should be shown to the user.

- **Fallback UI**
  - **Test Result**: Alternative content (like "No items found") was displayed when data was unavailable (e.g., empty product list).
  - **Screenshot**:
  - **If ID error**
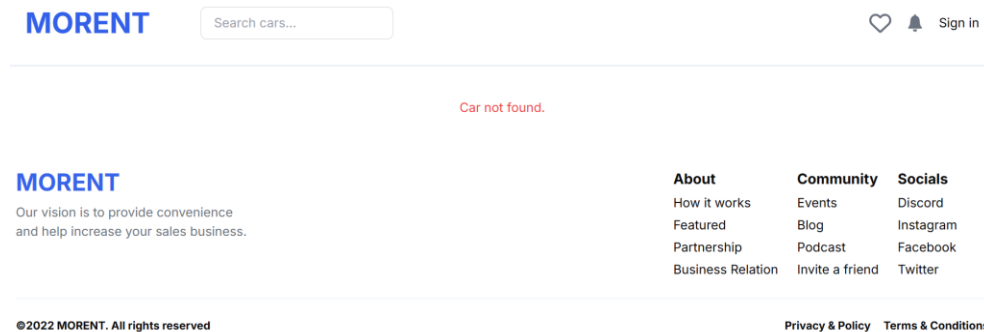
  - **If Page not found error**

  - **Expected Outcome**: If the data is not available, a fallback message (e.g., "No items found") should be displayed instead of leaving the screen blank.
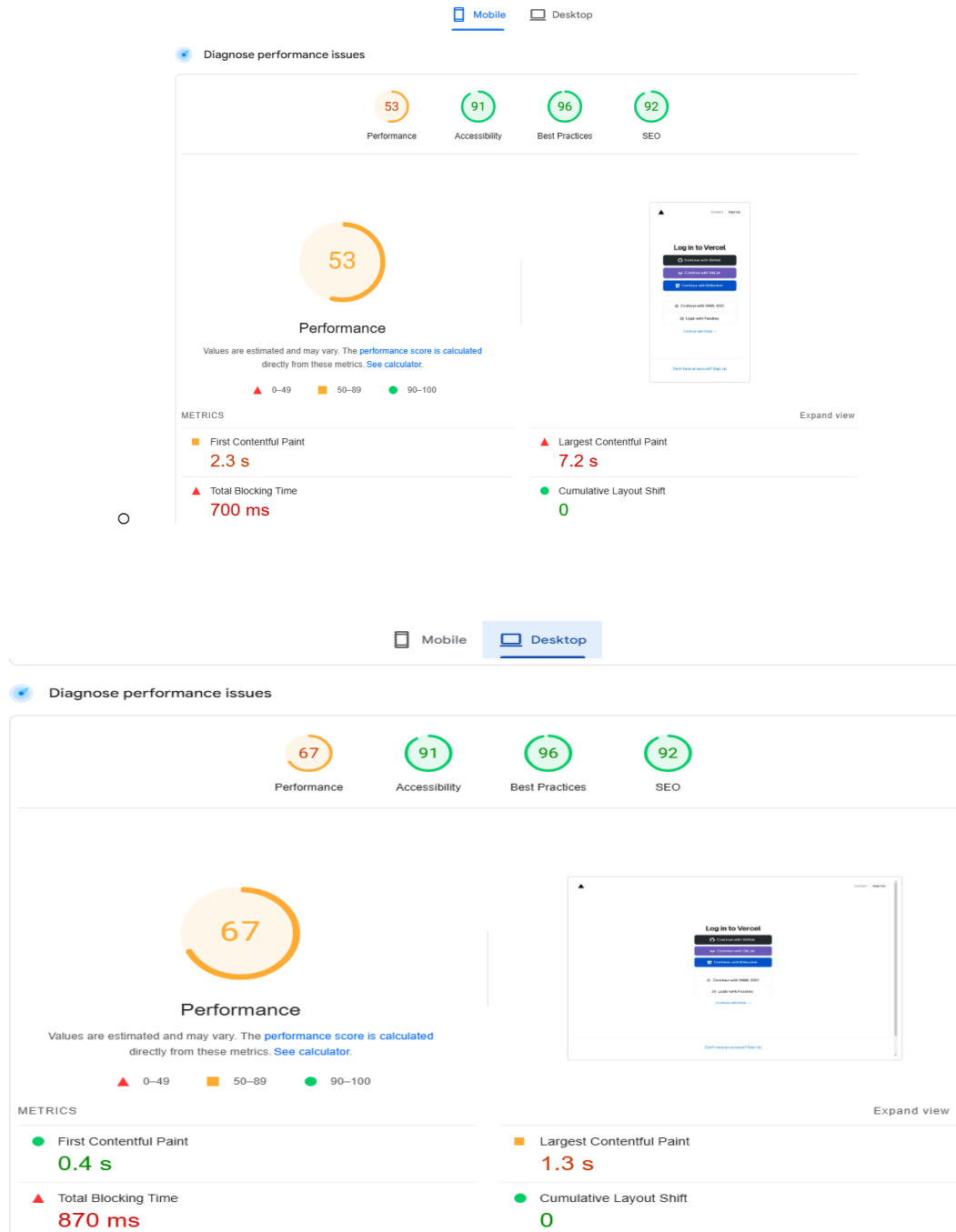
Step 3: Performance Optimization

- **Analyze Performance**
  - **Test Result**: Lighthouse analysis identified and resolved speed issues, including unused CSS, browser caching, and JavaScript optimization.
  - **Screenshot**:
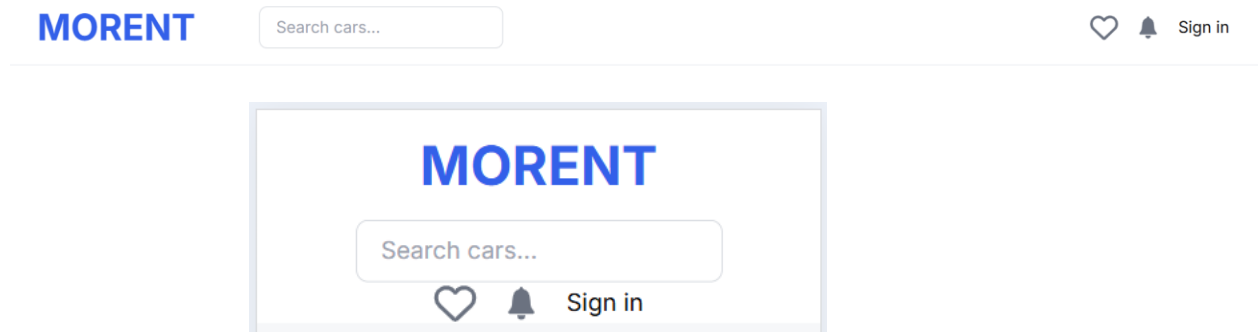
o **Expected Outcome**: The initial load time should be under 2 seconds, ensuring a smooth and quick user experience.
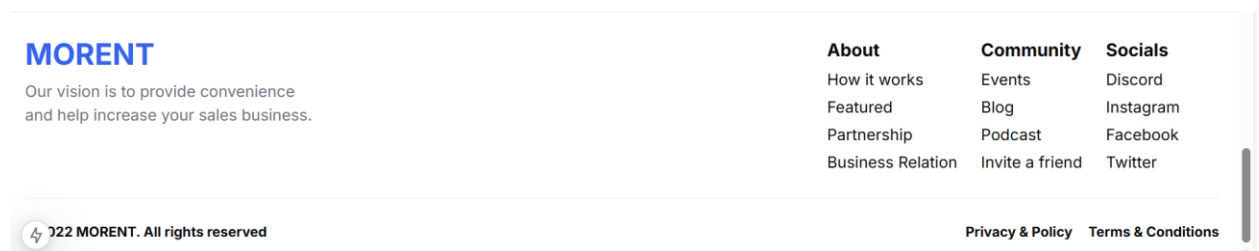
Step 4: Cross-Browser and Device Testing

- **Browser Testing:**
  - **Test Result:** Verified consistent rendering and functionality on Chrome, Firefox , and Edge.
  - **Screenshot:**

→ **Header**:

→**Footer**

MORENT

Our vision is to provide convenience and
help increase your sales business.

About

How it works

Featured

Partnership

Business Relation

Community

Events

Blog

Podcast

Invite a friend

Socials

Discord

Instagram

Facebook

Twitter

➔**Home Page:**

→Pick up / Drop off:

Pick-Up / Drop-Off Form

○ Pick-Up

Enter Location
[Enter location]

Select Date
[dd/mm/y 📅]

Select Time
[Hour ▼] [Minute ▼] [AM ▼]

○ Drop-Off

Enter Location
[Enter loca]

Select Date
[dd/mm, 📅]

Select Time
[Hour ▼] [Minute ▼] [AM ▼]

→**Product Containers:**

Rolls-Royce
SUV

⛽ 70L ❄ Manual 👥 6 People

$72.00/day   **Rent Now**

CR-V
SUV

⛽ 80L ❄ Manual 👥 6 People

$80.00/day   **Rent Now**

All New Terlos
SUV

⛽ 90L ❄ Manual 👥 6 People

$74.00/day   **Rent Now**

Rolls-Royce
Sedan

⛽ 70L ❄ Manual 👥 4 People

$96.00/day   **Rent Now**

Nissan GT-R
Sport

➔Dynamic Pages:

Sports car with the best design and acceleration

Safety and comfort while driving a futuristic and elegant sports car.

Rolls-Royce

★ ★ ★ ★ ☆ 440+ Reviews

Type Car: Sedan

Capacity: 4 People

Transmission: Manual

Fuel Capacity: 70L

popular

$96.00 / day   Rent Now

Sports car with the best design and acceleration

Safety and comfort while driving a futuristic and elegant sports car.

Rolls-Royce

★ ★ ★ ★ ☆ 440+ Reviews

Type Car: Sedan   **Capacity:** 4 People

Transmission: Manual   **Fuel Capacity:** 70L

popular

$96.00 / day   Rent Now

Expected Outcome: All browsers should render the website consistently without layout or functionality issues.

- **Device Testing:**
  - **Test Result:** Responsive design tested using BrowserStack and verified on one physical mobile device.
  - **Screenshot:**

MORENT

Search cars...

♡ 🔔 Sign in

Reviews

Leave a Review

Your Name

Your Review

1 Star ▾

Submit Review

✕

Filters

TYPE

☐ Sport   (2)
☐ Sedan   (1)
☐ SUV   (4)
☐ Hatchback (1)

CAPACITY

☐ 2 People (2)
☐ 4 People (1)
☐ 6 People (5)

People

)/day/day   Rent Now

GT-R   ♡

⚙ Manual 👥 2 People

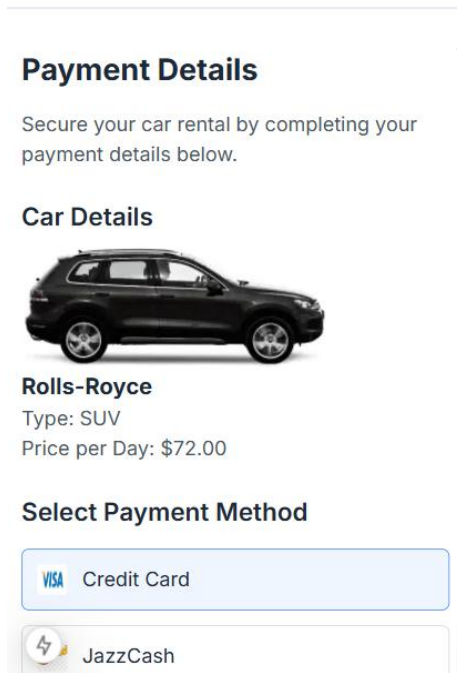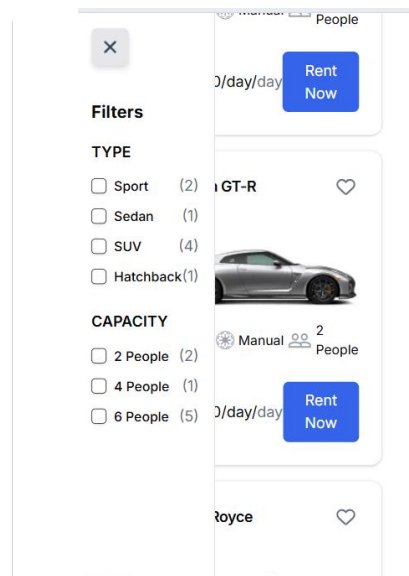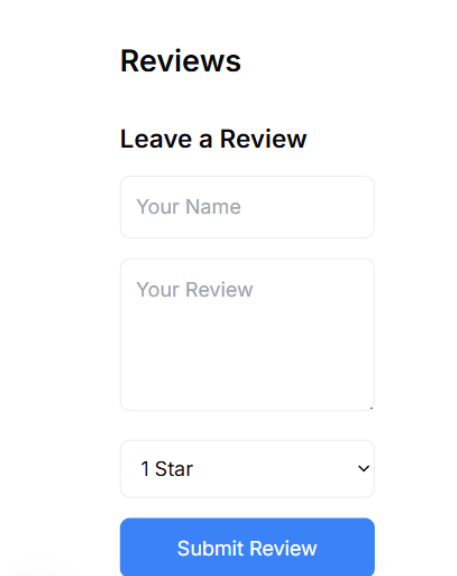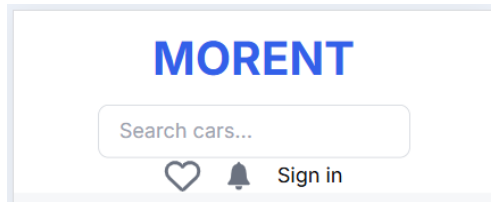)/day/day   Rent Now

Royce   ♡

Payment Details

Secure your car rental by completing your payment details below.

Car Details

Rolls-Royce
Type: SUV
Price per Day: $72.00

Select Payment Method

VISA Credit Card

⚡ JazzCash

Oops!

Looks like you've taken a wrong turn. Let's get back on track!

We couldn't find the page you're looking for.

🚗

Go Home

o   **Expected Outcome:** The website should adjust seamlessly across all device sizes and orientations.
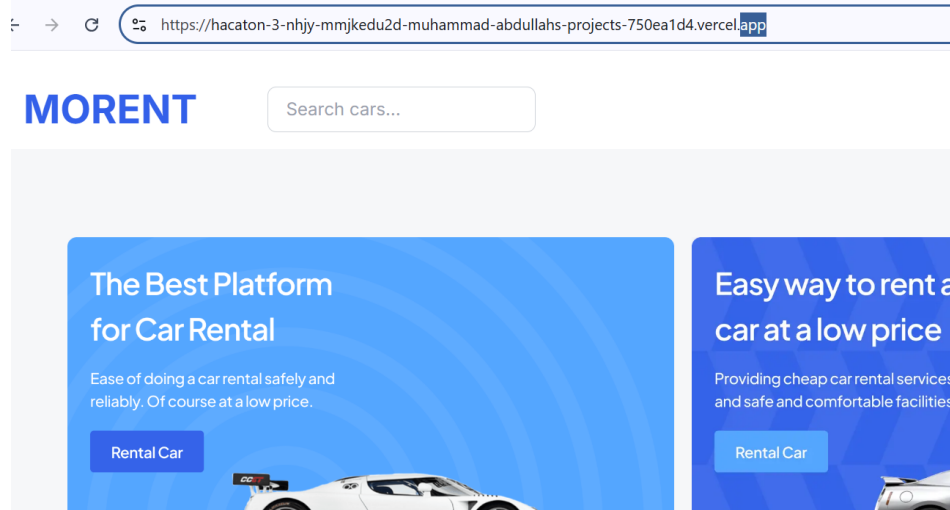
Step 5: Security Testing

1. Input Validation:
   - o Sanitize inputs to prevent SQL injection or XSS attacks.

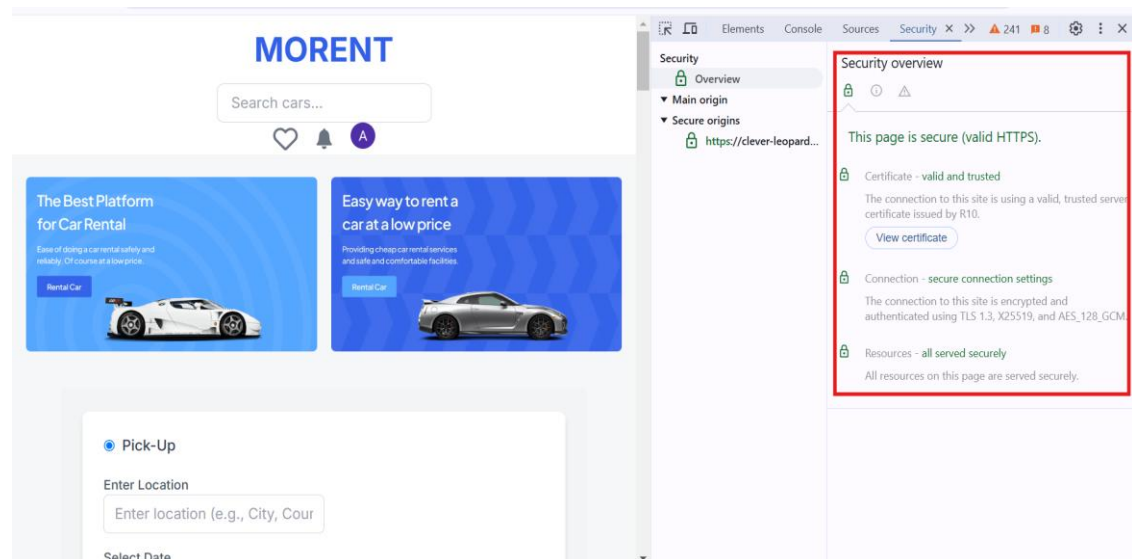   - o Use regular expressions to validate email, phone, and other inputs.

2. Secure API Communication:

   - o Ensure API calls are made over HTTPS.
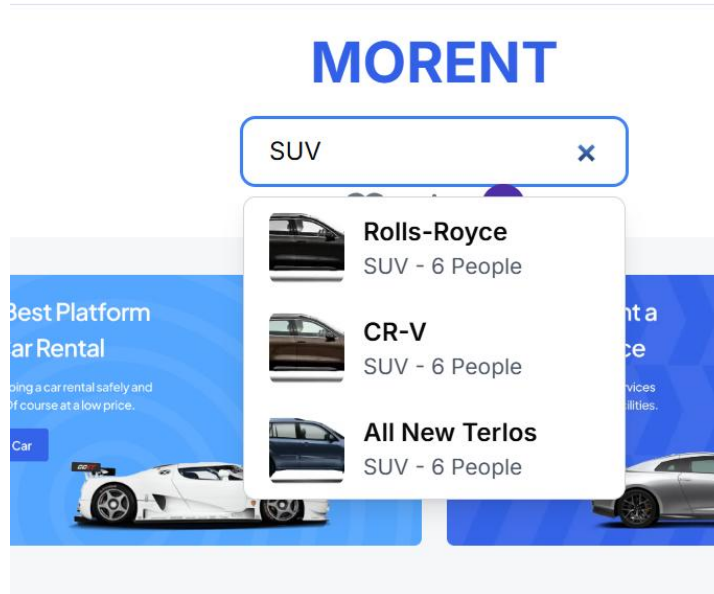

3. Testing Tools:

   By google

Step 6: User Acceptance Testing (UAT)

1. Simulate Real-World Usage:

   o Perform tasks like browsing products, Rent a car, and checking out.

      Filter car by searchbar with car name & brand also

Easily track order   filter cars by seats &
brand

OrderSummary

Thank you for your payment! Your order has been successfully processed.

Car Details

Car Name: Koenigsegg
Type: Sport
Price per Day: $99.00/day

Payment Details

Payment Method: JazzCash
Mobile Number: +923282513725
Account Holder Name: abdullah

Billing Details

Full Name: cccccccccccccd
Email: abdullahmuhammad20080409@gmail.com
Address: 232322222222222
City: 434333334444444
Zip Code: 2222

[Go Back to Home]   [Track Order]   [Download as PDF]
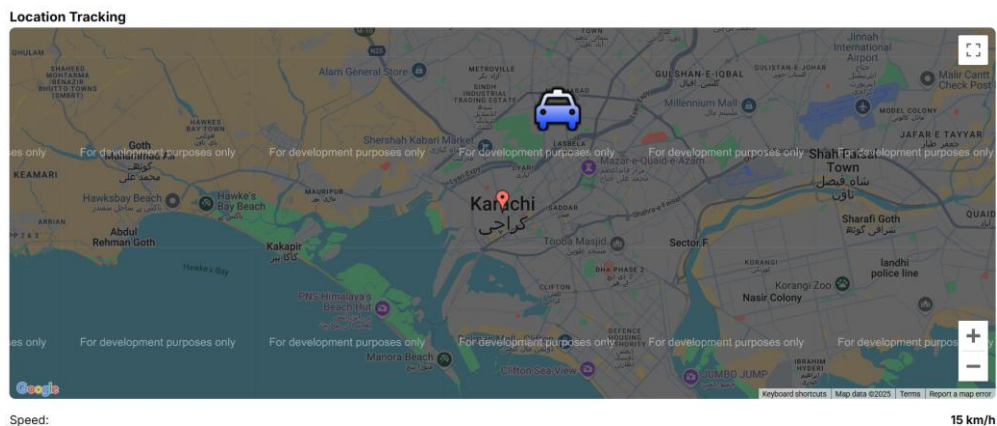
Filters

TYPE

☐ Gasoline	(2)
☐ Diesel	(2)
☐ Sport	(2)
☐ Electric	(2)
☐ Sedan	(1)
☐ SUV	(4)
☐ Hatchback	(1)
☐ Hybrid	(1)

CAPACITY

☐ 2 People	(2)
☐ 4 People	(1)
☐ 5 People	(5)
☐ 6 People	(6)

Easily track car

Location Tracking

Speed: 15 km/h

2. Feedback Collection from Friends and Family:

So, I had my family and friends test out my car rental website and got some feedback. Here's what they had to say:

- o **Navigation**: Everyone said the navigation was pretty easy. They found it simple to browse cars and search through categories.
- o **Design**: The logo and overall design got positive feedback. A lot of people said the website looks clean and modern, and it's user-friendly.
- o **Functionality**: A few people mentioned that the checkout process was smooth, but some noticed it took a bit of time to load the cart page. It's just a minor thing.

o **Suggestions**: One suggestion was to add more images to the car details, so customers get a better idea of what they're renting.
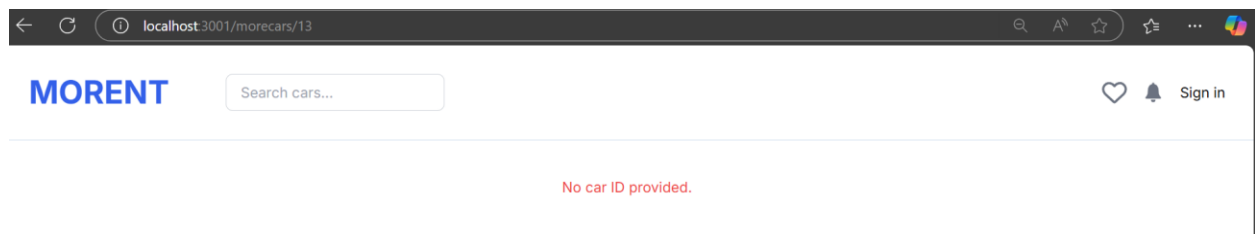
Overall, the feedback was pretty good, but I think there are still a few things that can be improved. I'll be doing some more testing before moving to the next step!

Step 7: Documentation Updates
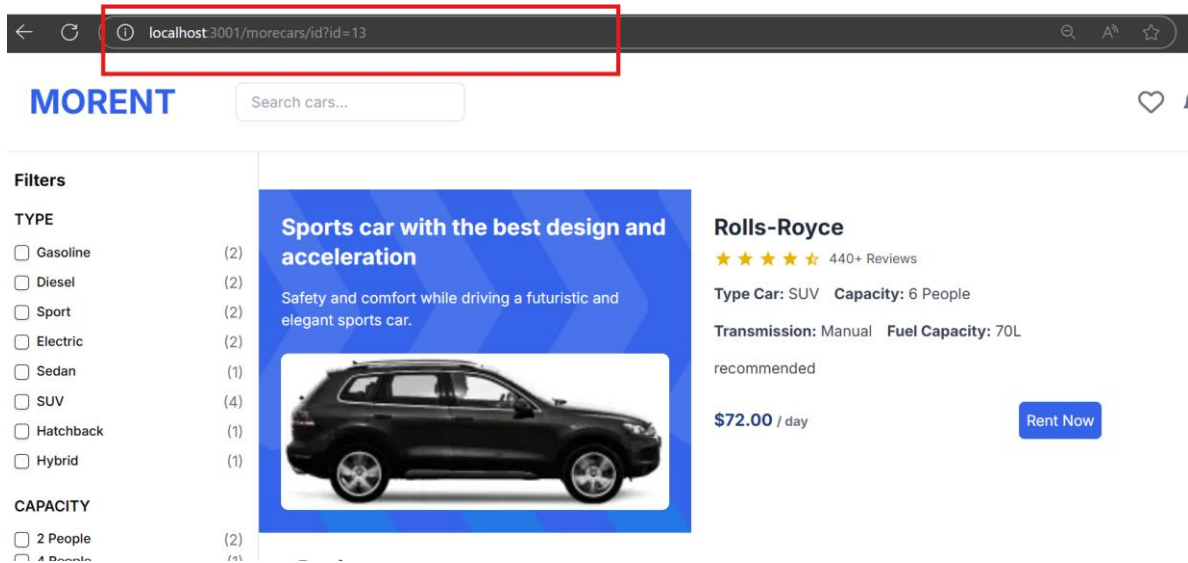
o Problem with the URL:

The link might not have worked because its path couldn't be found, which is why the "ID not found" error appeared.

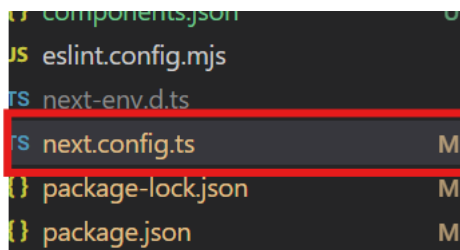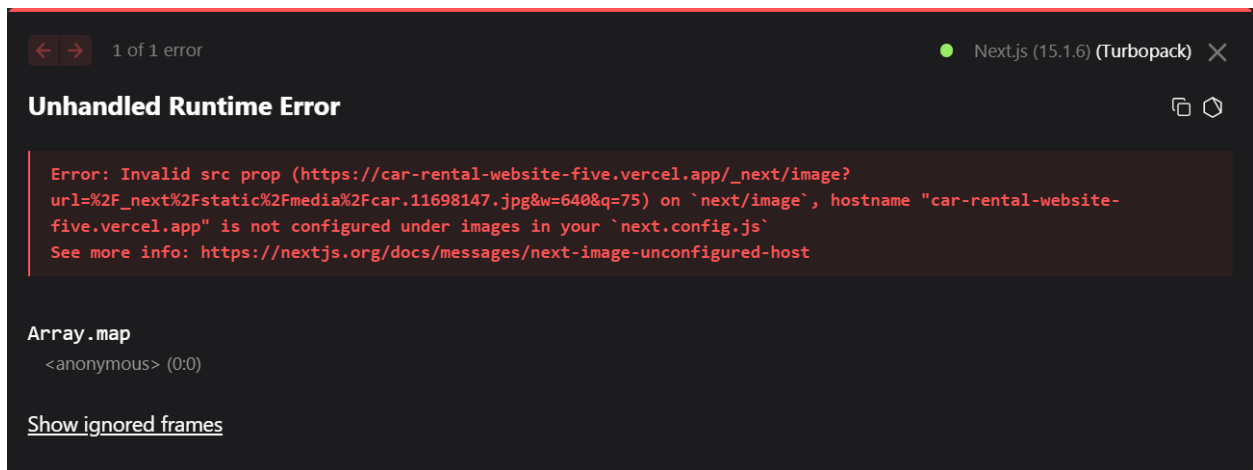<Link href={`/morecars/${car.id}`} className="bg-[#3563e9] p--2 text-white rounded-md">Rent Now</Link>

Then I wrote it this way, and it started working.

<Link href={`/morecars/id?id=${car.id}`} className="bg-[#3563e9] p-2 text-white rounded-md">Rent Now</Link>

o **Image Src Configuration Error :**

This error occurs because the external image domain is not configured in the next.config.ts file, causing Next.js to reject the image source.

```
1    import type { NextConfig } from "next";
2
3    const nextConfig: NextConfig = {
4      images: {
5        domains: ['car-rental-website-five.vercel.app'],
6      },
7    };
8
9    export default nextConfig;
```