Hackathon

Day 5 - Report

Prepared By

ABDUL RAHMAN MOIN

Day-5

Testing, Error Handling, and Backend Integration Refinement

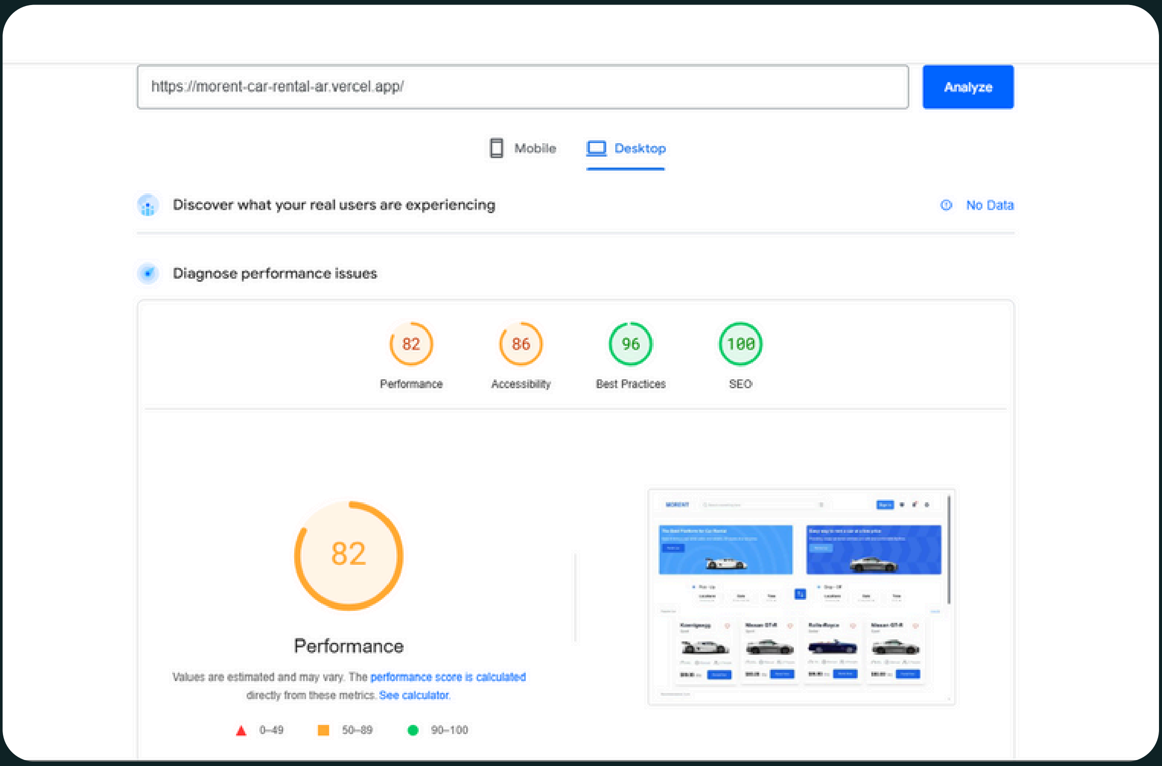
Lighthouse Analysis (Desktop)

The image shows a PageSpeed Insights report for the URL https://morent-car-rental-ar.vercel.app/

Here's a breakdown of the information shown:

Overall Performance:

- Performance: 82/100
- Accessibility: 86/100
- Best Practices: 96/100
- SEO: 100/100



https://morent-car-rental-ar.vercel.app/ Analyze

Mobile Desktop

Discover what your real users are experiencing — No Data

Diagnose performance issues

82 Performance · 86 Accessibility · 96 Best Practices · 100 SEO

82

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

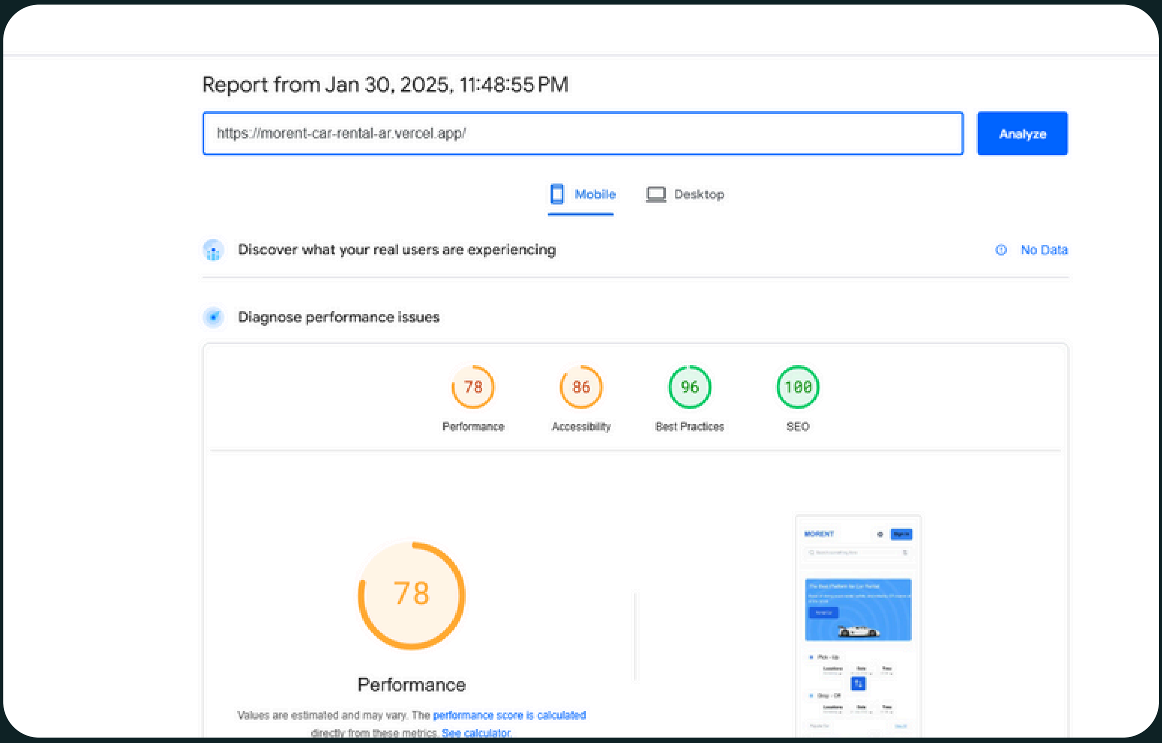
Lighthouse Analysis (Mobile)

The image shows a PageSpeed Insights report for the URL https://morent-car-rental-ar.vercel.app/

Here's a breakdown of the information shown:

Overall Performance:

- Performance: 78/100
- Accessibility: 86/100
- Best Practices: 96/100
- SEO: 100/100



Report from Jan 30, 2025, 11:48:55 PM

https://morent-car-rental-ar.vercel.app/

Discover what your real users are experiencing — No Data

Diagnose performance issues

78 Performance · 86 Accessibility · 96 Best Practices · 100 SEO

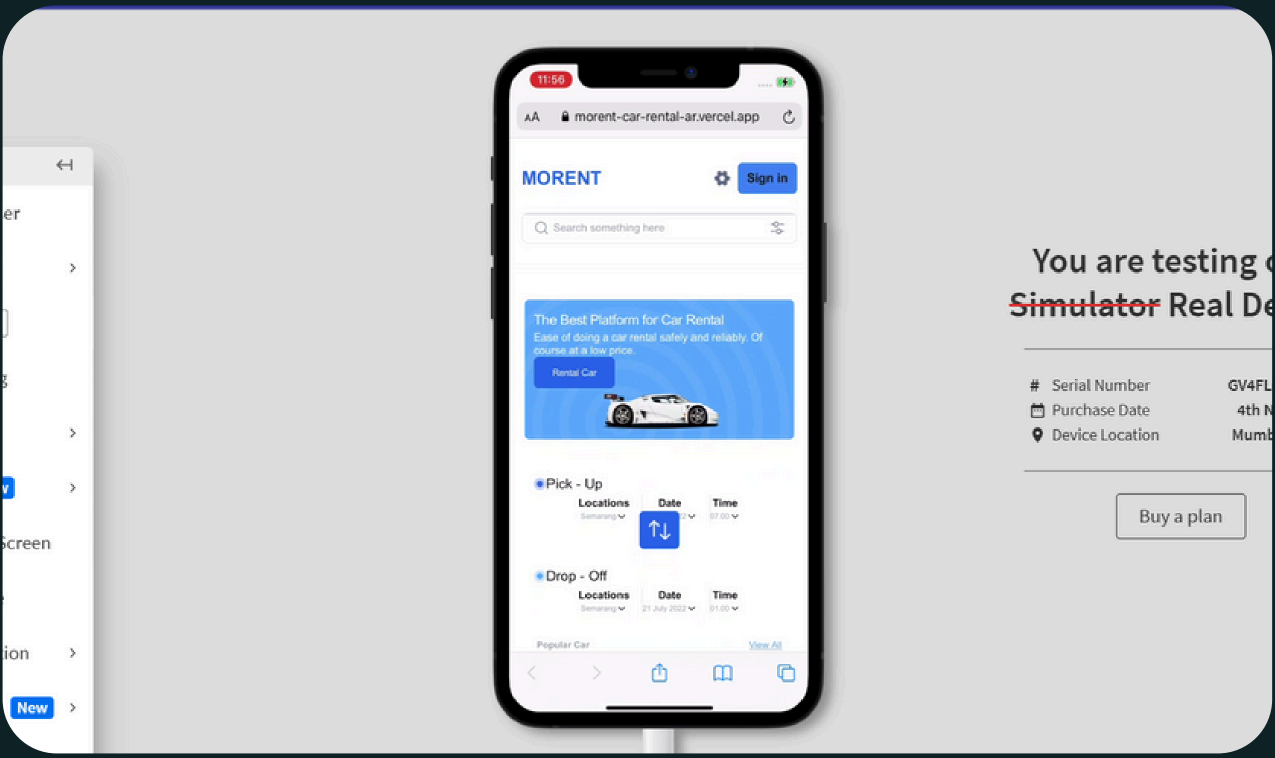
78 Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

Testing on BrowserStack

To ensure a smooth user experience across different devices and screen sizes, I conducted comprehensive testing using BrowserStack:

- Tested the responsiveness of the website across multiple screen sizes, including mobile, tablet, and desktop views.
- Verified that key functionalities such as authentication, booking flow, and checkout work consistently across devices.
- Ensured proper layout adjustments and UI rendering across different operating systems and browsers



Test Cases Executed:

- **Product Listing Page Validation:** Ensured that all products were displayed correctly on the product listing page. The test confirmed that items loaded as expected without missing images or incorrect details.

- **API Error Handling:** Simulated API failures to check whether proper error messages and fallback UI were shown. The system handled errors gracefully with informative messages.

- **Categorization Functionality:** Tested category selection by displaying the correct items when filtering. Verified that selecting multiple categories showed the corresponding results, and clearing the selection reset the view.

- **Search Functionality:** Entered search queries and validated that relevant results were displayed dynamically based on user input.

- **Mobile Responsiveness:** Resized the browser and tested on mobile devices to ensure the layout adjusted properly, maintaining usability across different screen sizes.

- **Authentication via Clerk:** Verified login and registration using email/password and Google sign-in. Ensured seamless authentication with appropriate user session handling.

- **Payment Integration via Stripe:** Processed test transactions to confirm successful payments and verify correct handling of success and failure messages.

Performance Optimization Steps Taken

To enhance the marketplace’s performance and ensure a seamless user experience, the following optimizations were implemented:

1. Lazy Loading Implementation:

- Integrated lazy loading for images and heavy assets to prevent unnecessary resource loading on initial page load.
- Reduced the time to first meaningful paint by deferring non-essential scripts.

2. Testing and Performance Analysis:

- Used Google Lighthouse to identify and address performance bottlenecks.

Security Measures Implemented

Ensuring the security of the marketplace was a key priority during development. Various security measures were implemented to protect user data, prevent unauthorized access, and secure transactions. Below are the key security steps taken:

1. Secure API Communication:

- Enforced HTTPS for all API requests to prevent data interception.
- Implemented token-based authentication using Clerk to ensure only authorized users can access sensitive endpoints.

2. User Authentication and Authorization:

- Integrated Clerk authentication with email/password and Google sign-in to ensure secure user login.
- Restricted access to protected pages, such as the booking and payment sections, requiring user authentication before proceeding.

3. Input Validation and Sanitization:

- Ensured proper sanitization of user inputs before processing API requests.

4. Payment Security Measures:

- Used Stripe Payment Gateway, which follows PCI-DSS compliance, to securely process transactions.

5. Error Handling for Security Enhancements:

- Used try-catch blocks to handle unexpected failures gracefully without revealing stack traces.

6. Cross-Origin Resource Sharing (CORS) Protection:

- Configured strict CORS policies to prevent unauthorized third-party access to the Sanity.
- Allowed only trusted domains to interact with Sanity.

Conclusion:

Day 5 was crucial in refining the overall stability and reliability of the marketplace. Comprehensive testing ensured that all core functionalities worked as expected, error handling mechanisms were effective, and performance optimizations significantly improved the user experience. With security measures in place, cross-browser compatibility verified, and seamless user interactions, the platform is now well-prepared for deployment and further scalability.