

Day 5 - Testing and Backend Refinement

Car Rental E-Commerce

Objective:

Test all features of the Car Rental platform to ensure proper functionality, error handling, and backend integration.

Features Verified:

1. Car Listings

- Cars are fetched correctly from the API and displayed on the homepage.
- All details like name, price, and image load without issues.
- Responsiveness verified on desktop and mobile.

2. Car Detail Pages

- Each car opens a dynamic detail page when clicking the "Details" button.
- Correct car information appears with no errors.

3. Wish List

- Clicking the heart icon adds a car to the wish list.
- Cars can be removed from the wish list.
- "Rent Now" starts the rental process without issues.

Error Handling:

- Clear error messages are shown if data fails to load (e.g., " Failed to fetch car data. Please try again later ").
- Tested scenarios like API failure and invalid responses.

Performance Optimization

Lighthouse Report Summary:

Performance: 67

- **FCP:** 1.1 s
- **LCP:** 1.9 s
- **TBT:** 12,100 ms
- **CLS:** 0.071
- **SI:** 3.4 s
- Key Issues: Minimize main-thread work (14.5 s), reduce JavaScript execution time (11.8 s), lazy loading LCP image, unused CSS/JS, and server response time (1,420 ms).

Accessibility: 92

- Issue: Insufficient color contrast.

Best Practices: 100

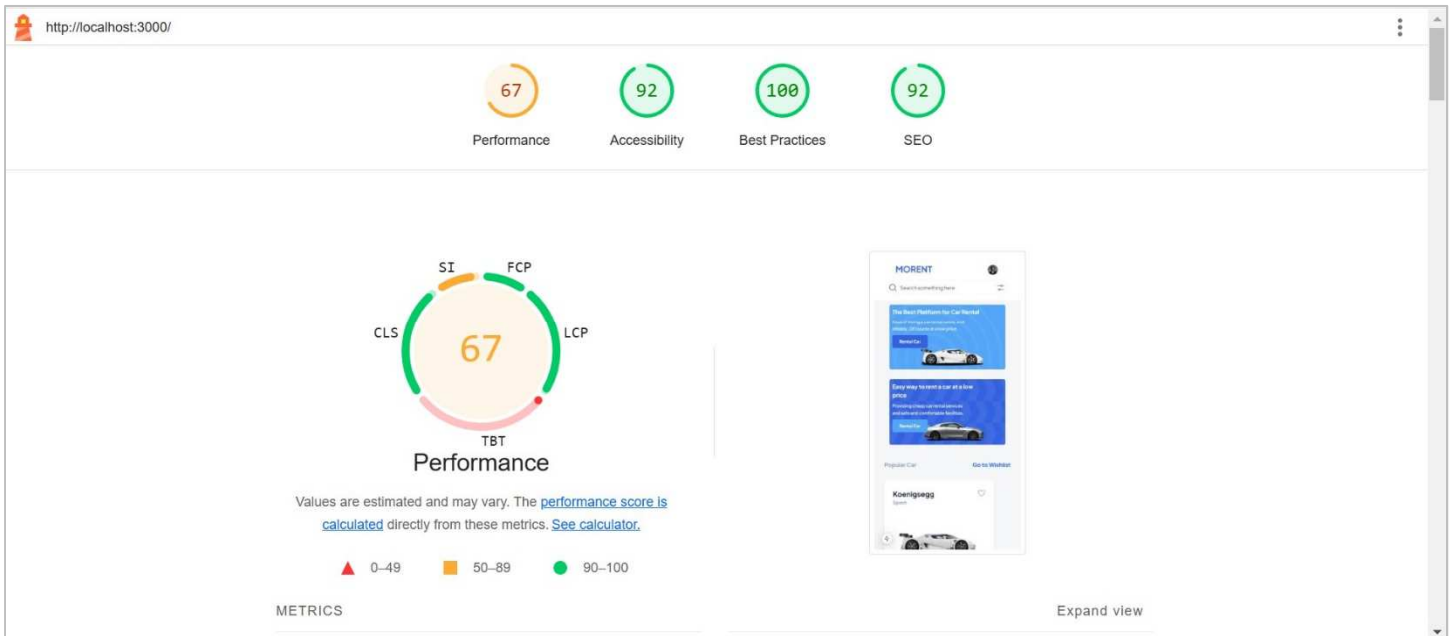
- All checks passed.

SEO: 92

- Issue: Invalid robots.txt file.

Overall Suggestions:

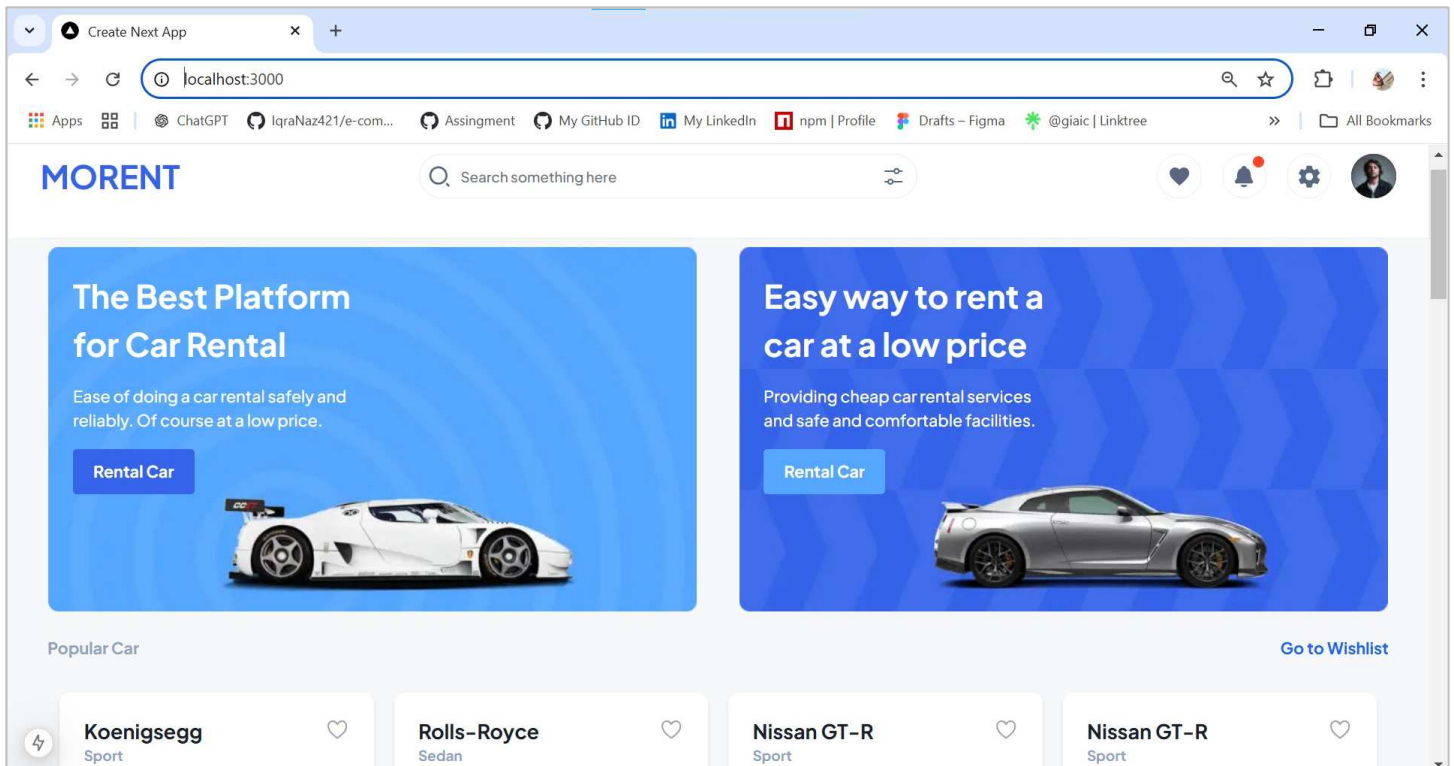
- Optimize server response time and lazy loading for LCP.
- Reduce unused CSS/JS and improve main-thread work.
- Address color contrast issues for better accessibility.
- Fix robots.txt for proper crawling.



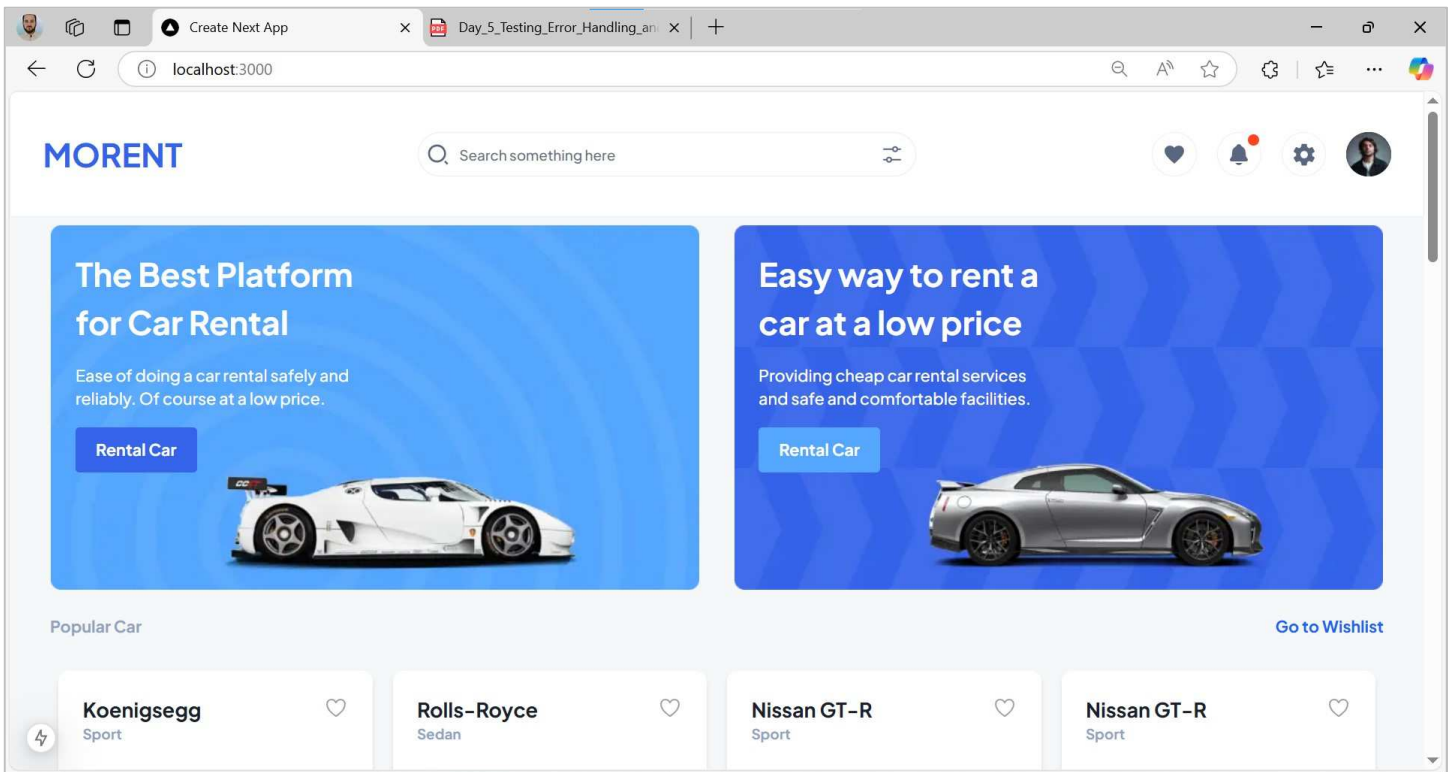
Compatibility:

- Verified functionality on major browsers: Chrome and Edge.
- Responsive design tested on devices including 360px mobile screens.

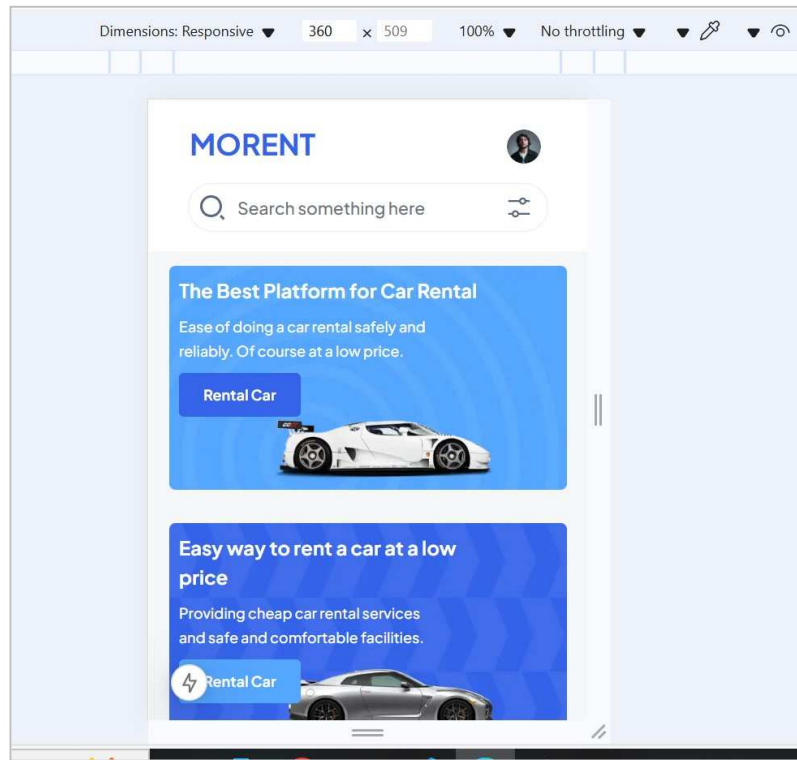
Google Chrome



Microsoft Edge



Responsive



CSV Based Testing Report

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Validate homepage rendering	Load the homepage > Check Hero and Footer	Homepage renders correctly with Hero and Footer	Homepage rendered as expected	Passed	High	Core functionality verified
TC002	Test dynamic car details page	Navigate to a car's detail page using slug	Details page loads with accurate car info	Details page working as expected	Passed	High	Dynamic routing functional
TC003	Check category filtering	Open category page > Select a filter	Cars list updates based on selected filter	Filter functionality working	Passed	Medium	Verified filter accuracy
TC004	Test responsiveness on mobile	Resize browser to mobile size > Verify layout	Layout adapts to smaller screens	Responsive layout verified	Passed	Medium	Works on mobile screens
TC005	Check API integration for cars	Disconnect API > Load cars page	Fallback UI displays an error	Error message displayed	Passed	Medium	Handled gracefully
TC006	Validate navigation links	Click on all navbar links > Verify redirection	Links redirect to the correct pages	Navigation working as intended	Passed	Low	No broken links found
TC007	Verify Tailwind CSS integration	Check class-based styling in components	Tailwind classes applied correctly	Tailwind functional	Passed	Low	Verified styling
TC008	Test checkout confirmation component	Complete a mock checkout > Check confirmation	Displays correct confirmation details	Confirmation shown	Passed	High	Checkout process verified
TC009	Validate wish list functionality	Add a car to the wish list > Verify list	Car is added to the wish list and displayed	Wish list working as expected	Passed	Medium	Wish list verified