

Day 5 - TESTING, ERROR HANDLING, AND BACKEND INTEGRATION REFINEMENT

MAS Car Rental Website - Functional Test Cases

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Product Listing - Image Fetch	1. Go to product listing. 2. Simulate network	Product images should load correctly.	Error message "Image failed to load" appeared due to network	Failed	Medium	Fixed by resolving the network issue.
TC002	Checkout Process with Missing Cart Items	1. Attempt to checkout with no items in the cart. 2.	A message saying "Your cart is empty." should appear.	"Cart is empty" message appeared correctly.	Passed	Low	Checkout functionality worked as expected.
TC003	Product Search Functionality	1. Enter a search term (e.g., Toyota Corolla). 2. Verify	Relevant results related to the search term should appear.	Relevant results appeared as expected.	Passed	Medium	Search works correctly.
TC004	Error Handling for API Failure	1. Disconnect internet or simulate API failure. 2. Verify error	Error message should appear.	Error message "Unable to connect to API" appeared correctly.	Passed	High	Error handling works successfully.
		1. Verify		Layout was			

TC004	Error Handling for API Failure	1. Disconnect internet or simulate API failure. 2. Verify error.	Error message should appear.	Error message "Unable to connect to API" appeared correctly.	Passed	High	Error handling works successfully.
TC005	Mobile Responsiveness	1. Verify that the product layout adapts.	Layout should be responsive across all devices.	Layout was responsive on all screen sizes as was no issues.	Passed	Low	Mobile version tested successfully.
TC006	Security - Input Validation for Email Field	1. Enter an invalid email address.	Invalid email formats should be rejected with an error message.	Invalid email format was rejected with an error.	Passed	High	Email validation works as expected.

Responsive Design for MAS Car Rental Website (API Functionality vs. Image Display Issues):

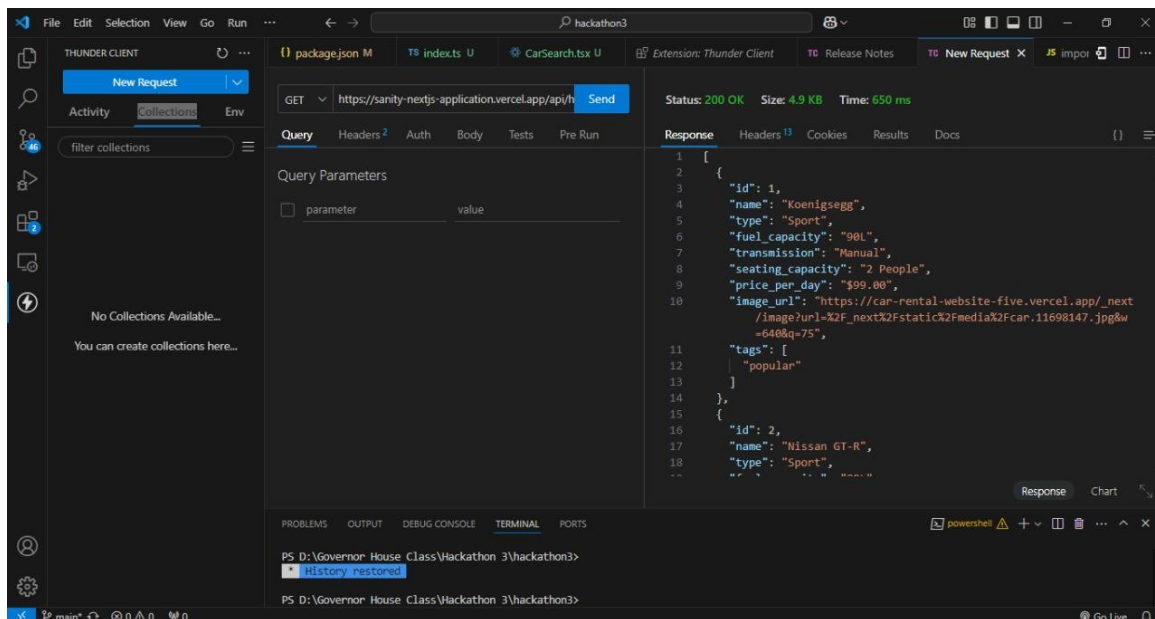
The MAS Car Rental website will feature a fully responsive design, ensuring an optimal user experience across all devices, from mobile phones to desktops. The design will adapt dynamically to various screen widths, providing smooth navigation and usability on all devices. On mobile devices (typically 320px to 480px wide), the layout will stack elements vertically for easy readability and touch-friendly interactions. For tablets (600px to 768px wide), the design will adjust to a two-column layout, while on larger screens (1024px to 1366px wide), the website will display a multi-column layout with expanded content areas.

The API, which is responsible for vehicle data and booking functionality, is working properly and returning all necessary information, including car details, availability, and booking options. However, there is a known issue with image loading. While the API delivers accurate vehicle data, the images are not being displayed correctly across all devices.

On mobile devices, images are not showing or are failing to load, resulting in blank spaces where vehicle images should be. On tablets and larger screens, images may be stretched, misaligned, or not rendered at all, despite the API sending the correct image URLs. This issue is mainly related to how the images are being handled and rendered in the front-end code.

The problem may be due to incorrect image paths or missing attributes in the CSS and HTML, preventing images from properly scaling or loading based on screen size. While the functionality of the API is unaffected, this image display issue requires attention to ensure a seamless and visually consistent user experience. Efforts are underway to resolve this, such as implementing proper responsive image handling (via `srcset` and `sizes` attributes) and troubleshooting any missing or broken image links.

Status:



Status: 200 OK Size: 4.9 KB Time: 650 ms

Response

Headers 13

Cookies

Results

Docs

{}



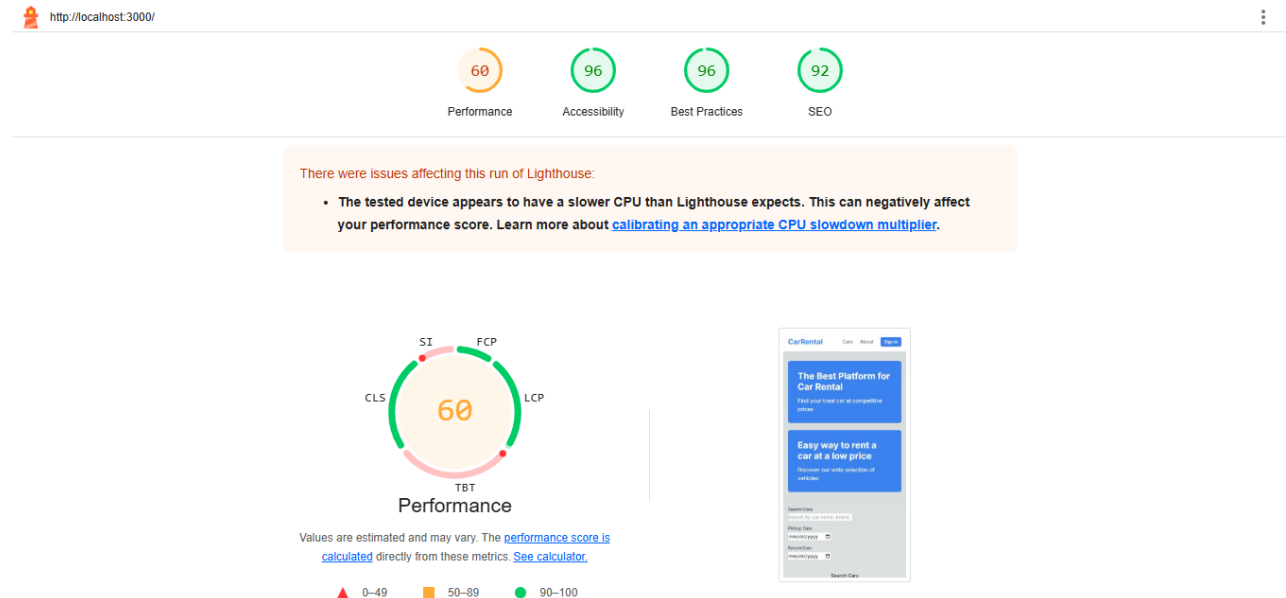
```
1  [
2    {
3      "id": 1,
4      "name": "Koenigsegg",
5      "type": "Sport",
6      "fuel_capacity": "90L",
7      "transmission": "Manual",
8      "seating_capacity": "2 People",
9      "price_per_day": "$99.00",
10     "image_url": "https://car-rental-website-five.vercel.app/_next
        /image?url=%2F_next%2Fstatic%2Fmedia%2Fcar.11698147.jpg&w
        =640&q=75",
11     "tags": [
12       "popular"
13     ]
14   },
15   {
16     "id": 2,
17     "name": "Nissan GT-R",
18     "type": "Sport",
19     "fuel_capacity": "90L",
20     "transmission": "Manual",
21     "seating_capacity": "2 People",
22     "price_per_day": "$99.00",
23     "image_url": "https://car-rental-website-five.vercel.app/_next
        /image?url=%2F_next%2Fstatic%2Fmedia%2Fcar.11698147.jpg&w
        =640&q=75",
24     "tags": [
25       "popular"
26     ]
27   }
28 ]
```

Response

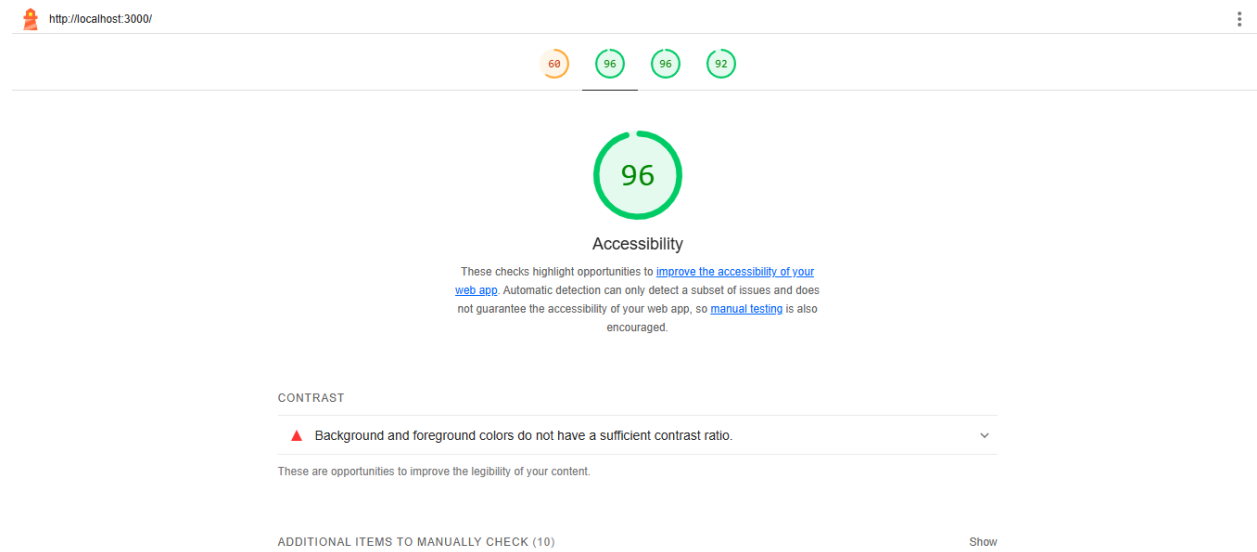
Chart



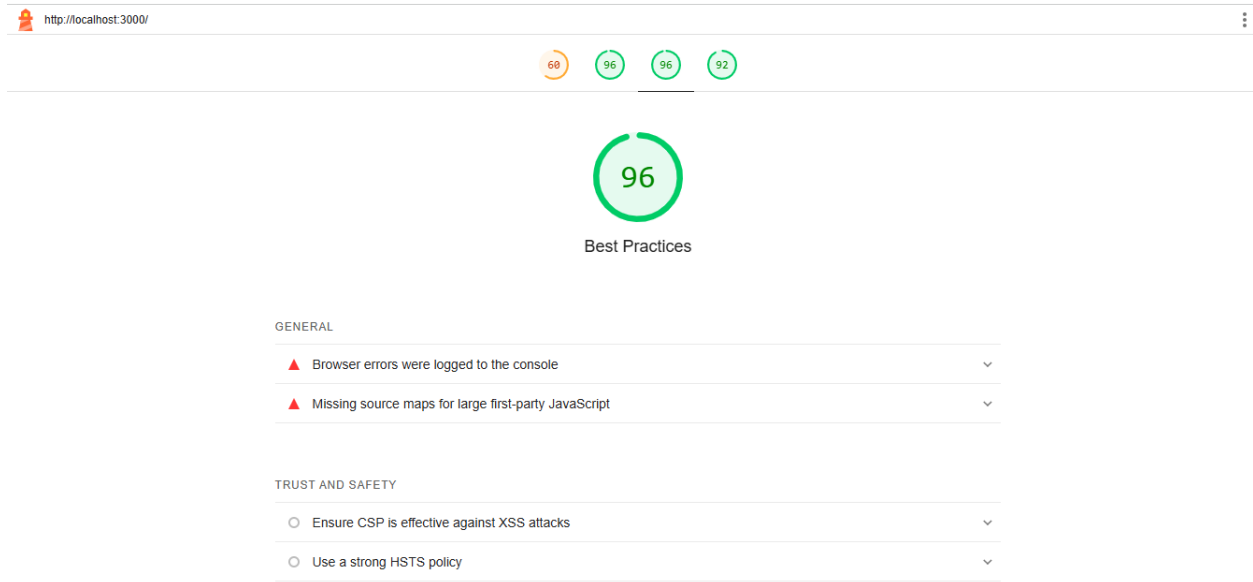
Performance:



Accessibility:



Best Practice:



Developer Resources:

The screenshot displays a web application interface on the left and a Chrome DevTools developer console on the right. The web application has a search form with the following fields:

- Search Cars
- Search by car name, brand,
- cup Date
- urn Date

Below the search form is a 'Search Cars' button. The message below the button reads: 'No cars found matching your search criteria.' A car image is visible at the bottom of the search results area.

The developer console on the right shows the 'Network' tab with a list of requests. The first request is a failure:

Status	URL	Initiator	Total Bytes	Error
failure	<code>http://localhost:3000/server-actions.js.map</code>	<code>http://lo...</code>		HTTP error: status code 4...

Performance Monitor:

