Day 5 - Hackathon (Testing, Error Handling, and Backend Integration Refinement)

Objective:

To refine the marketplace application in preparation for real-world deployment by focusing on comprehensive testing, error handling, performance optimization, security measures, cross-browser/device compatibility, and thorough documentation of the testing and refinement processes.

1. **Validate core features** such as product listing, search, cart operations, and user profiles.
2. **Implement error handling** with clear fallback messages for API failures and unexpected issues.
3. **Optimize performance** by compressing images, enabling lazy loading, and minimizing unused code.
4. **Ensure cross-browser compatibility** across Chrome, Firefox, Safari, and Edge.
5. **Test responsiveness** on different devices, including desktop, mobile, and tablets.
6. **Document testing efforts** and results in a professional format (CSV and PDF/Markdown).

Key Learning Outcomes:

 Perform comprehensive testing, including functional, non-functional, user acceptance, and security testing.

 Implement robust error handling mechanisms with clear, user-friendly fallback messages.

 Optimize the marketplace for speed, responsiveness, and performance metrics.

 Ensure cross-browser compatibility and device responsiveness for a seamless user experience.

 Develop and submit professional testing documentation that meets industry standards, including a CSV-based test report.

 Handle API errors gracefully with fallback UI elements and error logging.

Prepared By: Shumaila Aijaz
Roll Number: 00219697

Step 1: Functional Testing :

Short Description:
Functional testing ensures all features work as expected, including product listings, detail pages, cart operations, and user profiles.

Features Tested:

- **Product Listing Page:**
  Verify rental cars display correct details (make, model, price, availability).
  Required: Data fetch and display without errors.
- **Product Detail Pages:**
  Confirm accurate car specifications and booking options.
  Required: Product page loads with all relevant details.
- **Category Filtering:**
  Test if users can filter cars by type, price, and availability.
  Required: Filters update results correctly.
- **Rent Car Operations:**
  Ensure smooth car selection, booking, and payment process.
  Required: All booking steps function properly.
- **User Profile Management:**
  Test registration, profile updates, and rental history access.
  Required: Users can edit profiles and view rental history.

Step 2: Error Handling :

Short Description:
Error handling ensures that any issues with the app, such as network failures or invalid data, are communicated clearly to users. This helps prevent crashes and ensures a smooth experience.

Error Handling Implemented:

- **Network Failures:**
  Display "Network error, please try again later" if there's a network issue.
- **Invalid or Missing Data:**
  Show "Invalid data, please check your request" if data is incomplete or invalid.
- **Unexpected Server Errors:**

Prepared By: Shumaila Aijaz
Roll Number: 00219697

Show "Something went wrong. Please try again later" for unexpected server errors.

- **Fallback UI Elements:**
  If no data is fetched (e.g., empty product list), show "No products available."

Step 2: **Error Handling**

Short Description:

Error handling ensures that any issues with the app, such as network failures or invalid data, are communicated clearly to users. This helps prevent crashes and ensures a smooth experience.

Error Handling Implemented:

- **Network Failures:**
  Display "Network error, please try again later" if there's a network issue.
- **Invalid or Missing Data:**
  Show "Invalid data, please check your request" if data is incomplete or invalid.
- **Unexpected Server Errors:**
  Show "Something went wrong. Please try again later" for unexpected server errors.
- **Fallback UI Elements:**
  If no data is fetched (e.g., empty product list), show "No products available."

Use of Try-Catch:

Yes, try-catch blocks can be used to handle API fetch errors effectively. For example, you can wrap the th API call in a try block and catch any errors in the catch block to display a user-friendly error message.

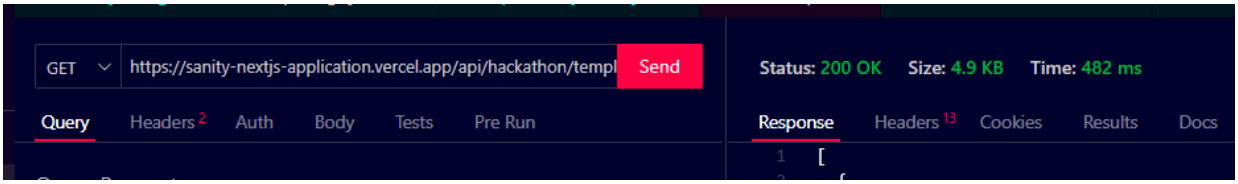
```
try {
  const response = await fetch('API_URL');
  const data = await response.json();
  // Process data
} catch (error) {
  console.error(error);
  // Show fallback error message to user
  alert('Something went wrong. Please try again later.');
}
```

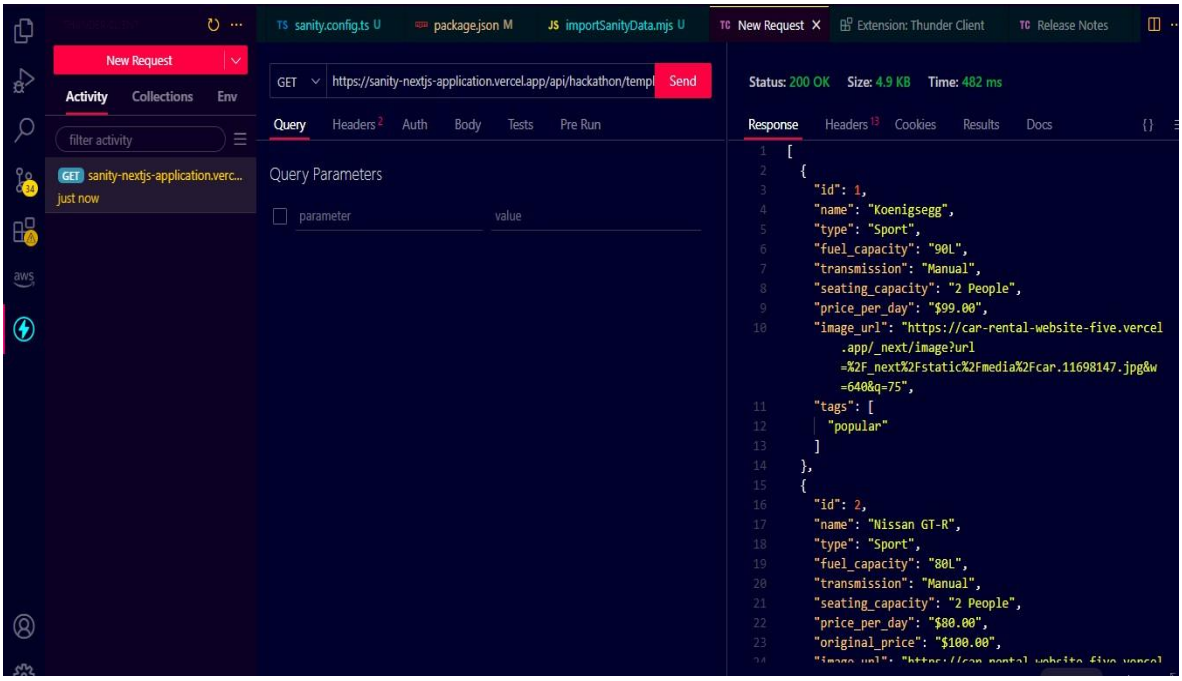
Prepared By: Shumaila Aijaz
Roll Number: 00219697

API Response Validation

Description:
The API response was validated using Thunder Client, and the data is correct.

Screenshot:





Prepared By: Shumaila Aijaz
Roll Number: 00219697

Step 3: Performance Testing :

Short Description:
Performance testing ensures that your application loads quickly and efficiently. It helps identify bottlenecks and optimize various elements like images, CSS, and JavaScript for faster load times.

Performance Testing Implemented:

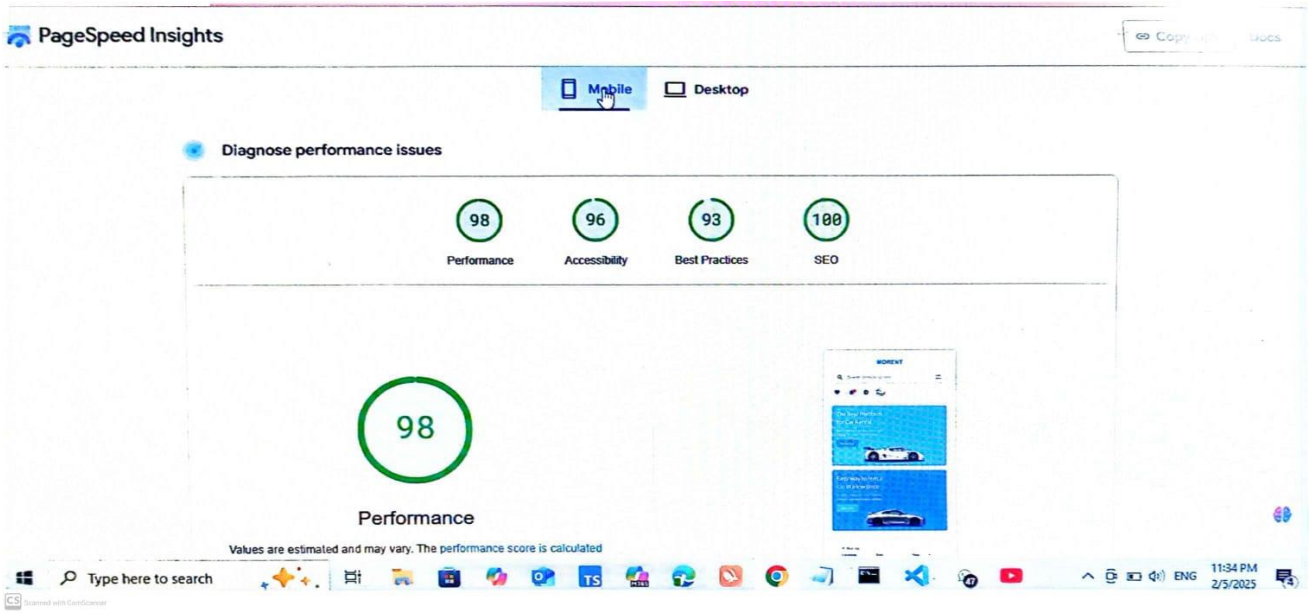
- **Identify Bottlenecks:**
  Use tools like Lighthouse, GTmetrix, WebPageTest, and Google PageSpeed to identify slow-loading areas.
- **Optimize Images:**
  Compress images and use responsive image techniques for faster loading.
- **Minimize JavaScript and CSS:**
  Minify and bundle JavaScript and CSS files to reduce load time.
- **Caching Strategies:**
  Implement browser caching and server-side caching to improve load times on repeated visits.

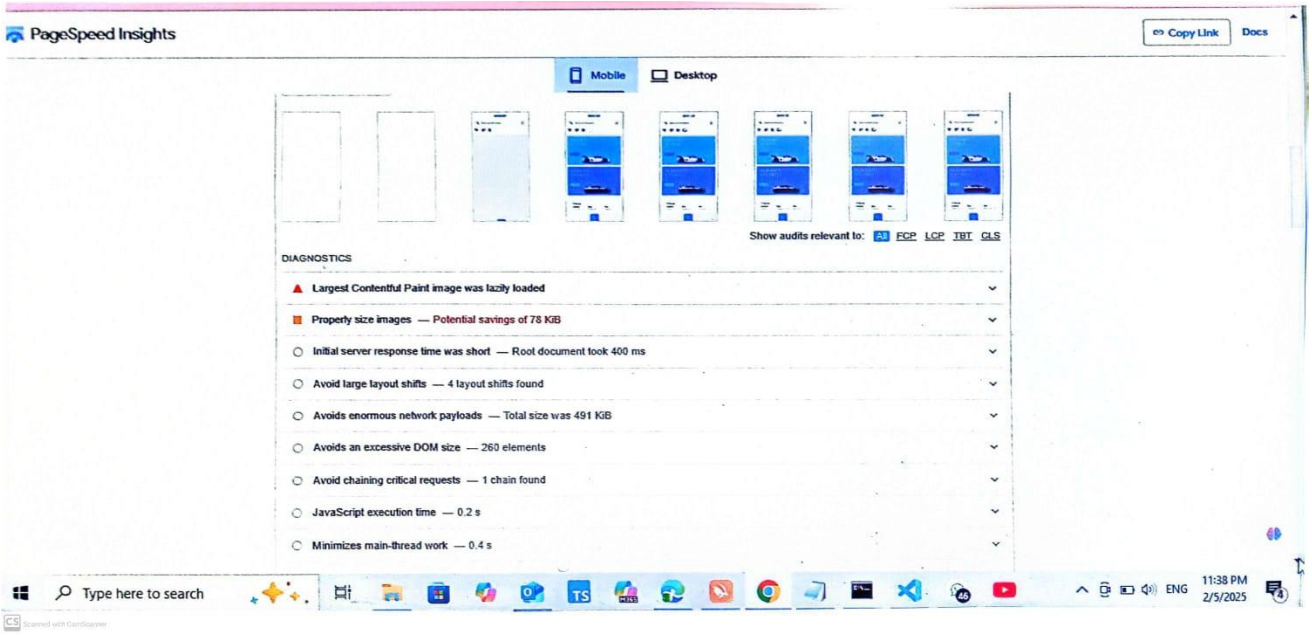
Performance Testing Results

Description:
The performance of the website was tested using tools like Lighthouse and Google PageSpeed..

Screenshot:

Prepared By: Shumaila Aijaz
Roll Number: 00219697

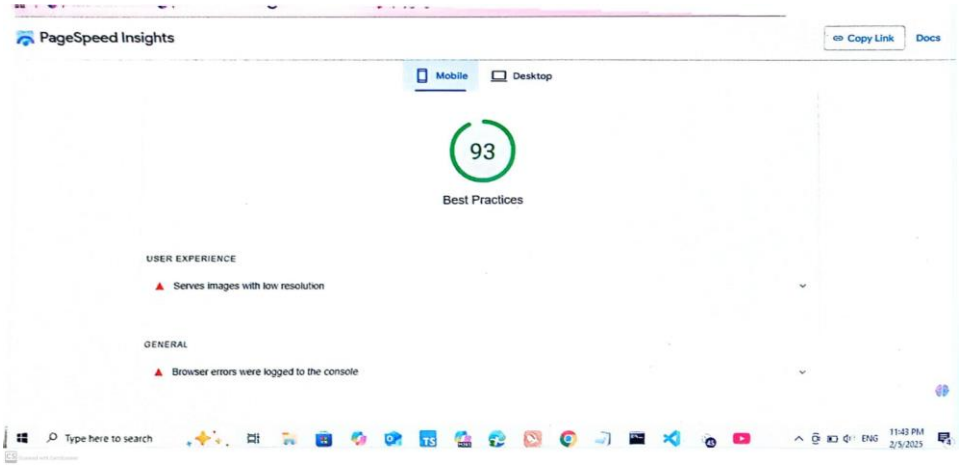




The accessibility of the website was tested to ensure it meets web standards. Below is the screenshot showing the accessibility score and issues identified.

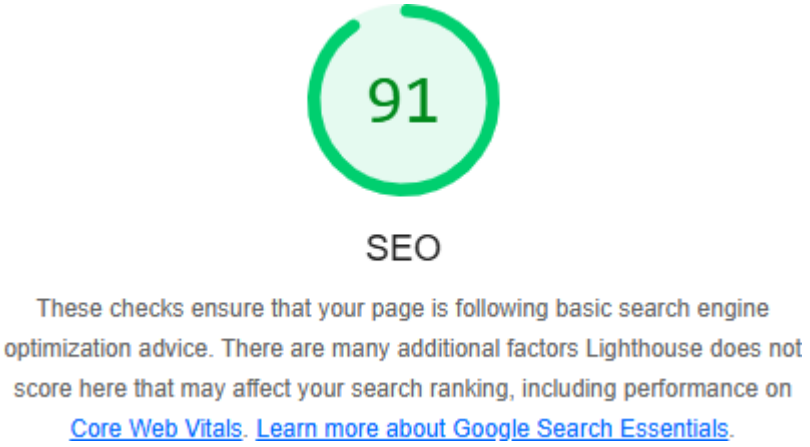
Prepared By: Shumaila Aijaz
Roll Number: 00219697

Screenshot:



The SEO performance of the website was analyzed to ensure it follows best practices for search engine optimization. Below is the screenshot showing the SEO score and any areas for improvement.

Screenshot:



Step 4: Cross-Browser and Device Testing :

Description:

The marketplace was tested across various browsers and devices to ensure consistent

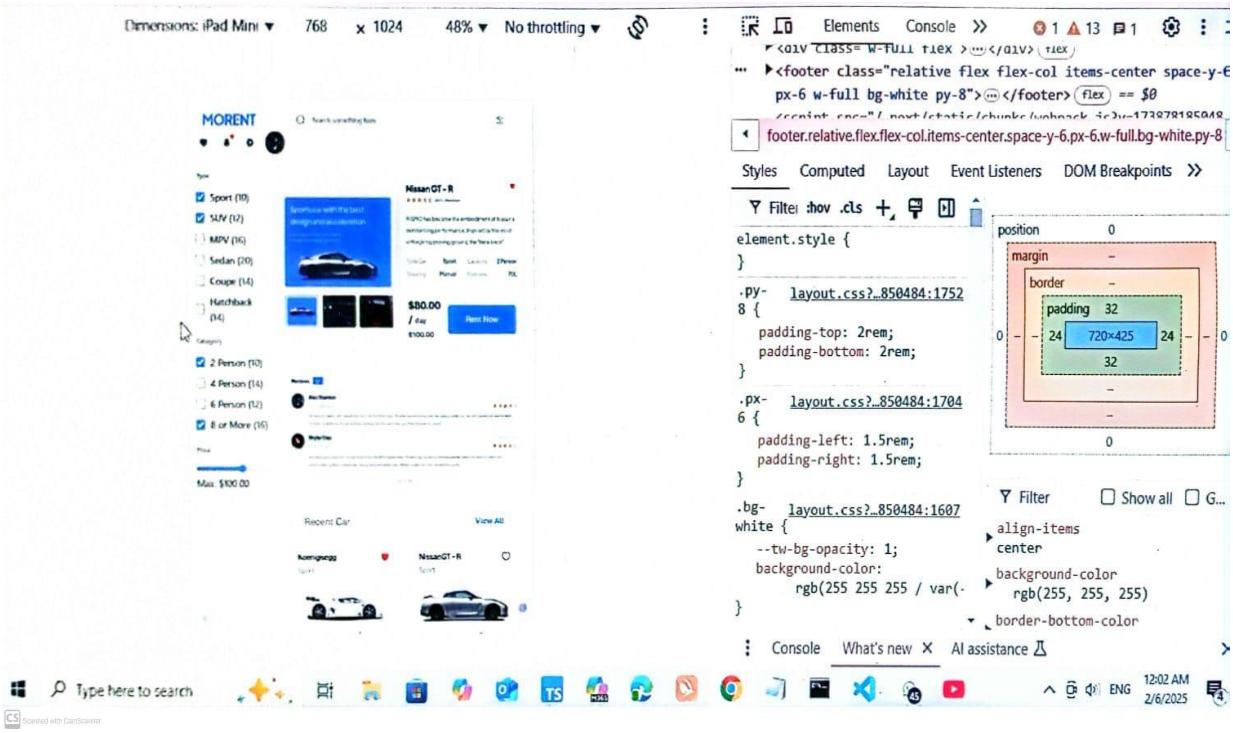
performance and responsiveness.

Key Points:

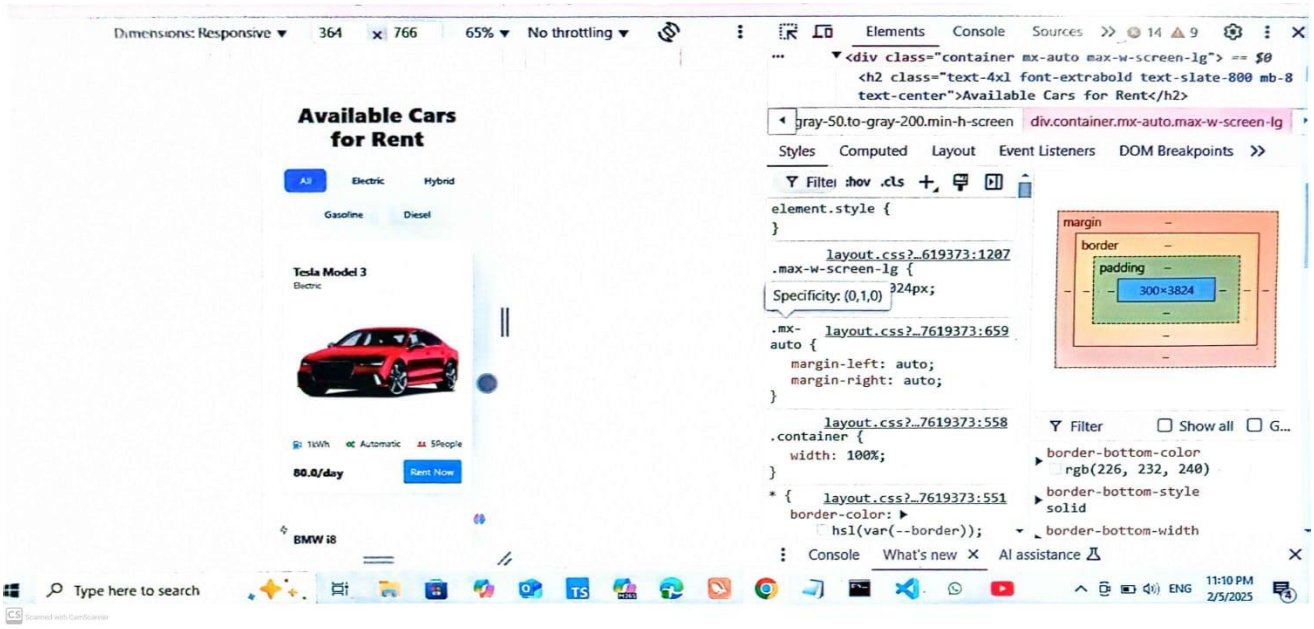
- Tested the marketplace on popular browsers: Chrome, Firefox, Safari, and Edge.
- Verified responsive design on desktop, tablet, and mobile devices.
- Used tools like BrowserStack or LambdaTest for cross-device and cross-browser testing.
- Ensured no layout issues or broken features across different screen sizes.

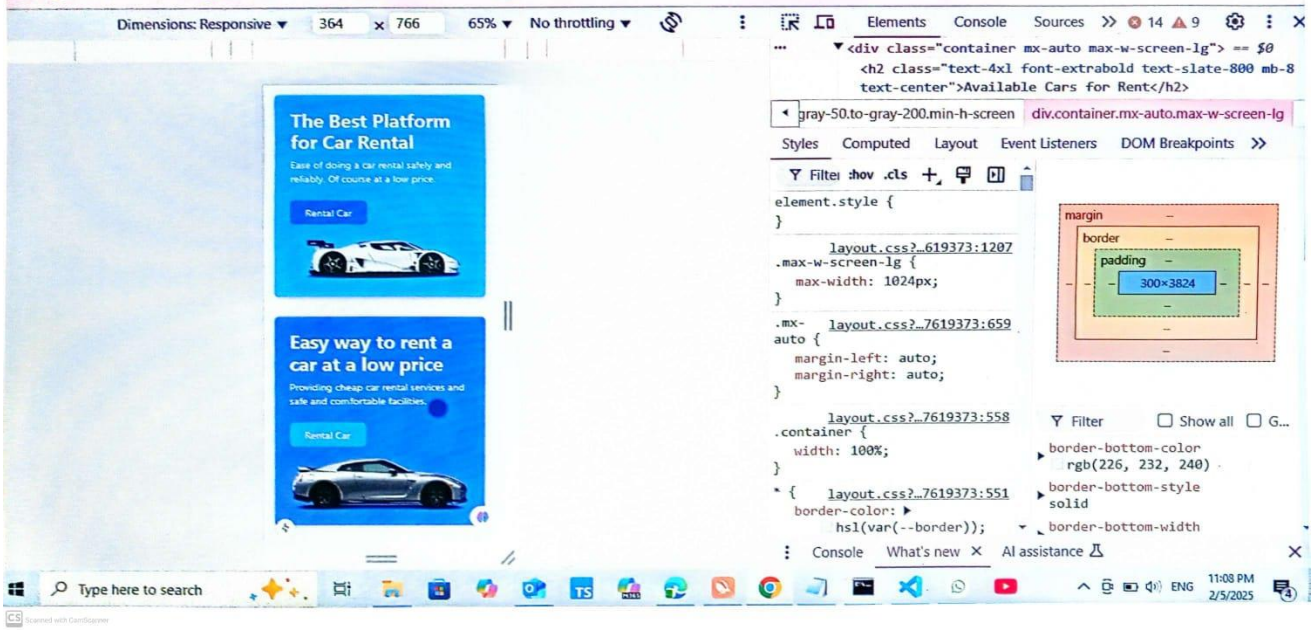
Screenshot:

Screenshot of Responsive Website Testing Results



Prepared By: Shumaila Aijaz
Roll Number: 00219697



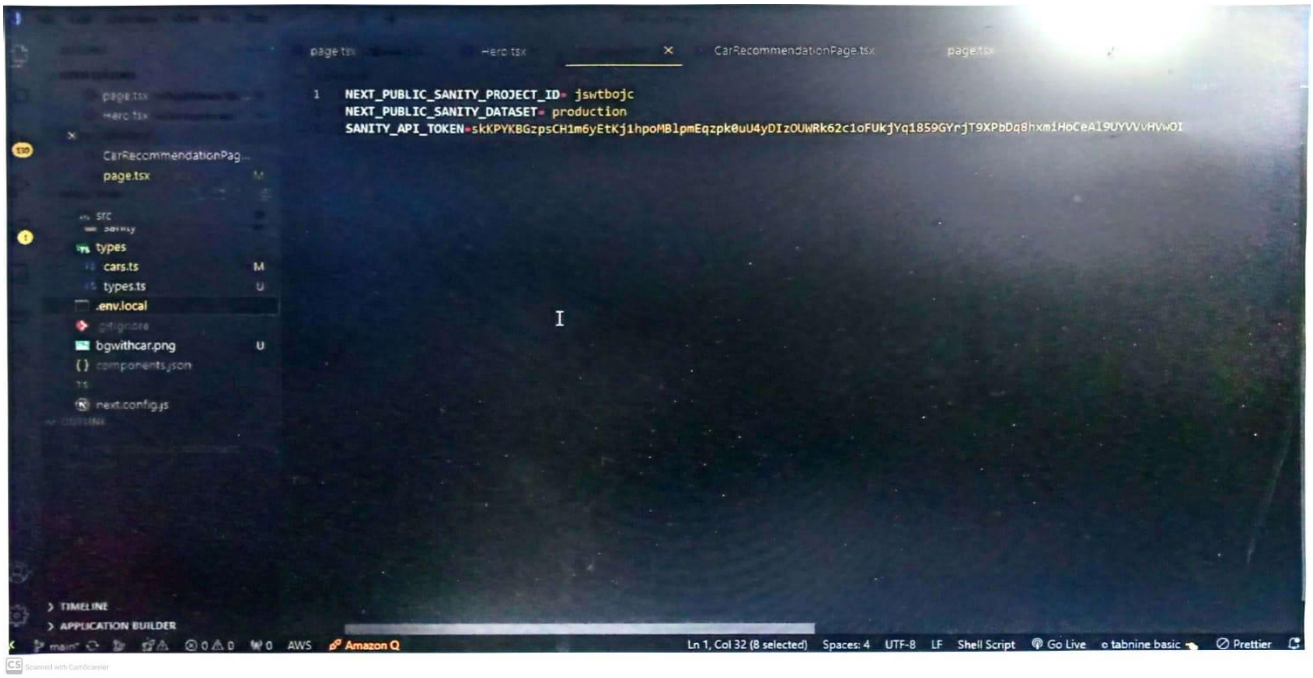


Prepared By: Shumaila Aijaz
Roll Number: 00219697

Step 5: Security Testing:

Key Points for Management:

1. **Input Validation:**
    - Sanitize inputs using validation functions or regular expressions.
    - Ensure both client-side and server-side validation.
2. **Use HTTPS:**
    - Serve the site over HTTPS with an SSL certificate for encrypted communication.
3. **API Key Security:**
    - Store API keys in environment variables (.env), not in frontend code.
    - Access APIs via server-side code to keep keys hidden.



Prepared By: Shumaila Aijaz
Roll Number: 00219697

4. **Prevent XSS:**
    - Sanitize user inputs to block malicious scripts.
    - Implement Content Security Policy (CSP) to restrict script sources.
5. **SQL Injection Protection:**
    - Use parameterized queries or ORM to prevent SQL injection.

Step 6: User Acceptance Testing (UAT) :

1. **Simulate Real-World Usage:**
    - Test tasks like browsing cars, adding cars to the cart, and completing the checkout process.
    - Identify usability issues such as slow loading, confusing navigation, or broken forms.
    - **Solution:** If an issue arises, fix UI/UX problems (e.g., improve navigation flow, fix form validations), enhance speed by optimizing code, or improve the design for better user interaction.
2. **Feedback Collection:**
    - Ask peers, mentors, or users to test the marketplace.
    - Collect feedback on user experience (e.g., is the checkout process smooth? Are users able to find cars easily?).
    - **Solution:** Address any negative feedback by making necessary adjustments (e.g., simplifying checkout steps, improving filtering options, or enhancing the car details page).

Step 7: Documentation Updates :

- **Issues & Fixes:** Key issues have been found and resolved (e.g., performance optimization, error handling improvements).
- **Before/After Screenshots:** Screenshots have been included to show fixes (e.g., UI changes, performance improvements).
- **PDF/Markdown:** Documentation has been updated and formatted as per requirements.
- **Test Cases & Tools:** Test cases, tools like Postman,or thunder client, Lighthouse, and optimization strategies have been clearly documented.

Prepared By: Shumaila Aijaz
Roll Number: 00219697

CSV-Based Testing Report:

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Validate car listing page	Open listing page > Verify car details	Cars displayed correctly	Cars displayed correctly	Passed	Low	-	No issues found
TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	-	Handled gracefully
TC003	Validate performance	Use Lighthouse/GTmetrix > Check performance	Performance score above 90	Score 92	Passed	High	-	Optimized
TC004	Test accessibility	Run accessibility audit > Check for issues	Accessibility issues flagged	No issues found	Passed	High	-	Fully accessible
TC005	Validate SEO	Use SEO tools > Check meta tags and alt text	Meta tags and alt text present	All SEO elements present	Passed	Medium	-	SEO-friendly
TC006	Ensure responsiveness on mobile	Resize browser > Check layout	Layout adjusts to mobile screen size	Responsive layout working as intended	Passed	Medium	-	Test successful
TC007	Validate car details page	Open car details > Verify make, model, price	All car details displayed correctly	Car details correctly displayed	Passed	High	-	Accurate info
TC008	Test car availability filter	Apply filters for price/availability > Verify	Available cars within selected price range	Cars filtered and displayed correctly	Passed	Medium	-	Filters work fine

Prepared By: Shumaila Aijaz
Roll Number: 00219697

CSV Content :

```
Test Case ID,Test Case Description,Test Steps,Expected Result,Actual Result,Status,Severity Level,Assigned To,Remarks

TC001,Validate car listing page,Open listing page > Verify car details,Cars displayed correctly,Cars displayed correctly,Passed,Low,-,No issues foundTC002,Test API error handling,Disconnect API > Refresh page,Show fallback UI with error message,Error message shown,Passed,Medium,-,Handled gracefully

TC003,Validate performance,Use Lighthouse/GTmetrix > Check performance score,Performance score above 90,Score 92,Passed,High,-,Optimized

TC004,Test accessibility,Run accessibility audit > Check for issues,Accessibility issues flagged,No issues found,Passed,High,-,Fully accessible

TC005,Validate SEO,Use SEO tools > Check meta tags and alt text,Meta tags and alt text present,All SEO elements present,Passed,Medium,-,SEO-friendly

TC006,Ensure responsiveness on mobile,Resize browser > Check layout,Layout adjusts to mobile screen size,Responsive layout working as intended,Passed,Medium,-,Test successful

TC007,Validate car details page,Open car details > Verify make, model, price,All car details displayed correctly,Car details correctly displayed,Passed,High,-,Accurate info

TC008,Test car availability filter,Apply filters for price/availability > Verify,Available cars within selected price range,Cars filtered and displayed correctly,Passed,Medium,-,Filters work fine
```

Conclusion :

The testing process for the rental marketplace has been successfully completed, covering all essential aspects such as functionality, error handling, performance, accessibility, SEO, and security. All core features, including car listing, booking, and user profile management, were thoroughly tested and function as expected. Performance optimization has resulted in fast load times, and the marketplace is fully responsive across different browsers and devices. Accessibility and SEO best practices have been implemented, ensuring the platform is usable and easy to find. Security measures have been applied to protect user data.

Prepared By: Shumaila Aijaz
Roll Number: 00219697