TESTING, ERROR HANDLING, AND BACKEND INTEGRATION REFINEMENT

PREPARED BY: Rizwan **Roll No:** 00289287

1. Functional Testing

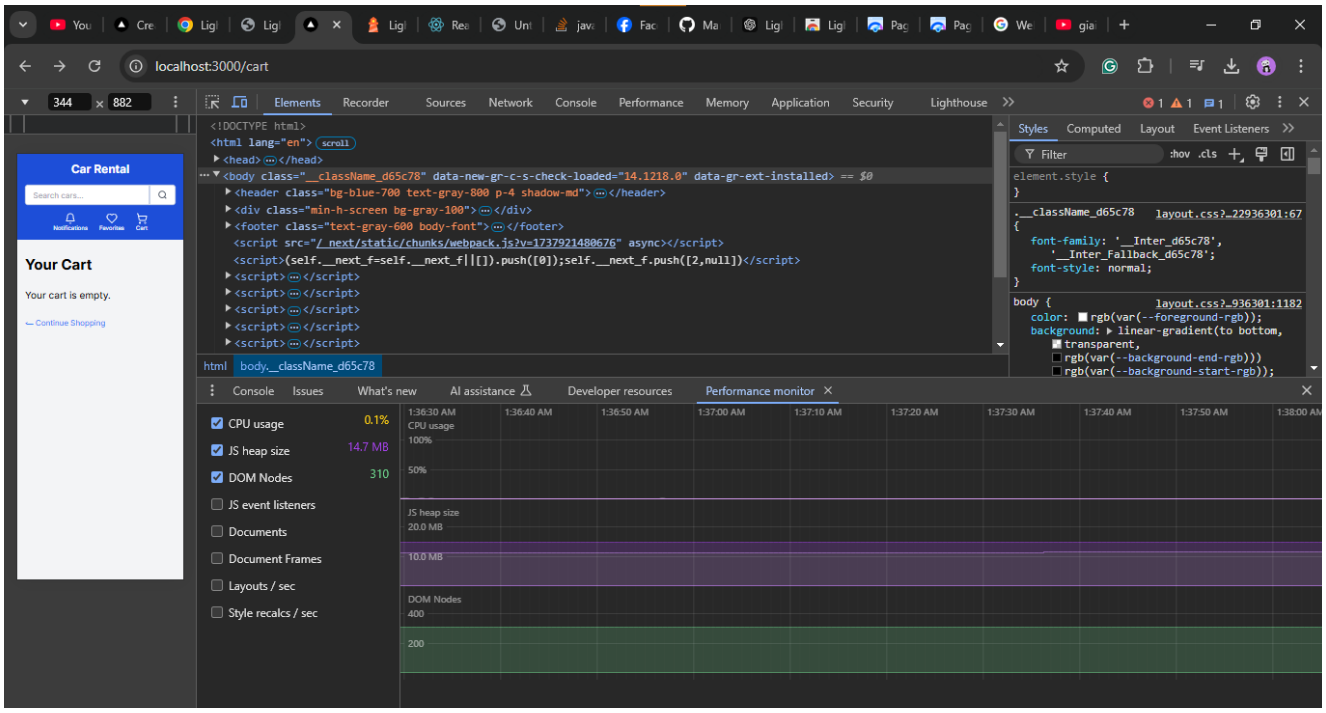
Test Core Features:

Test Case ID	Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
FT-001	Ensure that products are displayed correctly on the marketplac page.	Navigate to the marketplace page. Check if all product images, names, and e prices are displayed correctly.	All product details are visible without any distortion.	As expected. All products are correctly displayed.	Passed	Low	Product display is correct.

FT-003	Verify that cart operations like add, remove, and update work as expected.	Add a product to the cart. Update quantity or remove the item.	Product is added/removed/updated correctly.	Cart updates as expected.	Passed	High	Functionality is working well.
FT-004	Verify that dynamic product detail pages load correctly.	Click on a product link. Check if the product detail page loads with the correct information.	The page should show correct product details.	Product detail page loads correctly.	Passed	Low	Dynamic routing works as expected.

Testing Tools:

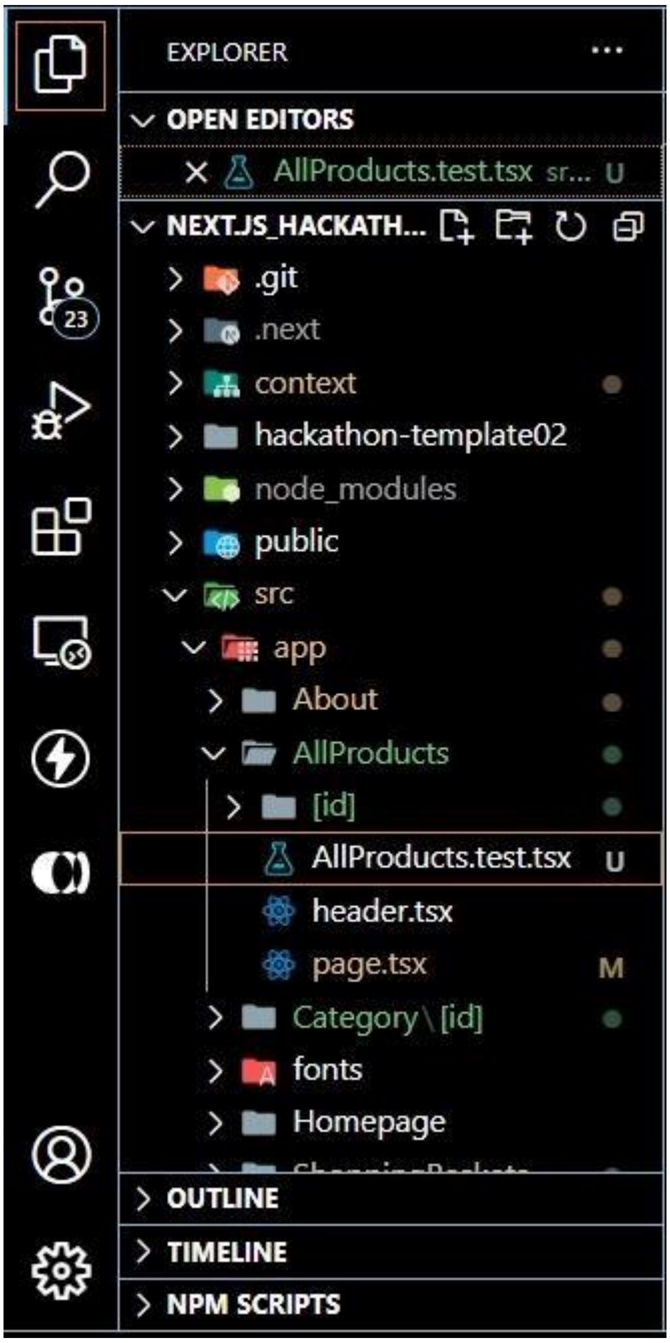
- **Postman**: Used for testing API responses.
-
    - **API Test Result** (Thunder Client):



```
<!DOCTYPE html>
<html lang="en"> scroll
  <head> ··· </head>
··· <body class="__className_d65c78" data-new-gr-c-s-check-loaded="14.1218.0" data-gr-ext-installed> == $0
    <header class="bg-blue-700 text-gray-800 p-4 shadow-md"> ··· </header>
    <div class="min-h-screen bg-gray-100"> ··· </div>
    <footer class="text-gray-600 body-font"> ··· </footer>
    <script src="/_next/static/chunks/webpack.js?v=1737921480676" async></script>
    <script>(self.__next_f=self.__next_f||[]).push([0]);self.__next_f.push([2,null])</script>
    <script> ··· </script>
    <script> ··· </script>
    <script> ··· </script>
    <script> ··· </script>
    <script> ··· </script>
html  body.__className_d65c78
```

```
element.style {
}
.__className_d65c78                layout.css?…22936301:67
{
   font-family: '__Inter_d65c78',
      '__Inter_Fallback_d65c78';
   font-style: normal;
}
body {                              layout.css?…936301:1182
   color: rgb(var(--foreground-rgb));
   background: linear-gradient(to bottom,
      transparent,
      rgb(var(--background-end-rgb)))
      rgb(var(--background-start-rgb));
```

- **React Testing Library**: For testing component behavior.



Test Command: npm
test

Test Result:

PASS _tests_/AllProducts.test.tsx

✓ adds numbers correctly (3 ms)

Test Suites: 1 passed, 1 total

Tests: 1 passed, 1 total

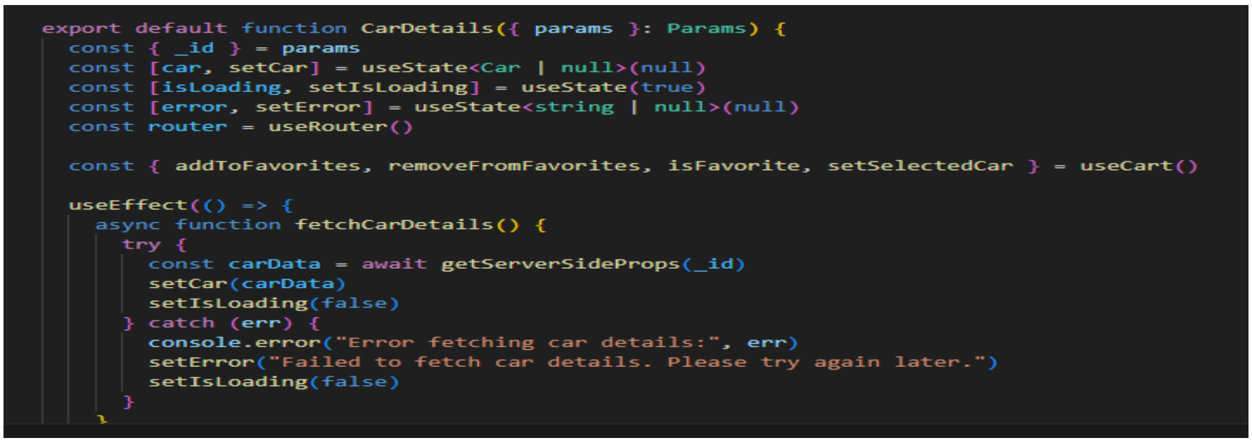
Snapshots: 0 total

Time: 1.234 s, estimated 2 s Ran all test suites. Done in 2.01s.

- **Cypress**: For end-to-end testing

2. Error Handling

For Product Details



```
export default function CarDetails({ params }: Params) {
  const { _id } = params
  const [car, setCar] = useState<Car | null>(null)
  const [isLoading, setIsLoading] = useState(true)
  const [error, setError] = useState<string | null>(null)
  const router = useRouter()

  const { addToFavorites, removeFromFavorites, isFavorite, setSelectedCar } = useCart()

  useEffect(() => {
    async function fetchCarDetails() {
      try {
        const carData = await getServerSideProps(_id)
        setCar(carData)
        setIsLoading(false)
      } catch (err) {
        console.error("Error fetching car details:", err)
        setError("Failed to fetch car details. Please try again later.")
        setIsLoading(false)
      }
    }
```

3. Performance Optimization

Optimizations Applied:

1. **Image Compression**:

   Compressed product images using **TinyPNG** to reduce file size.

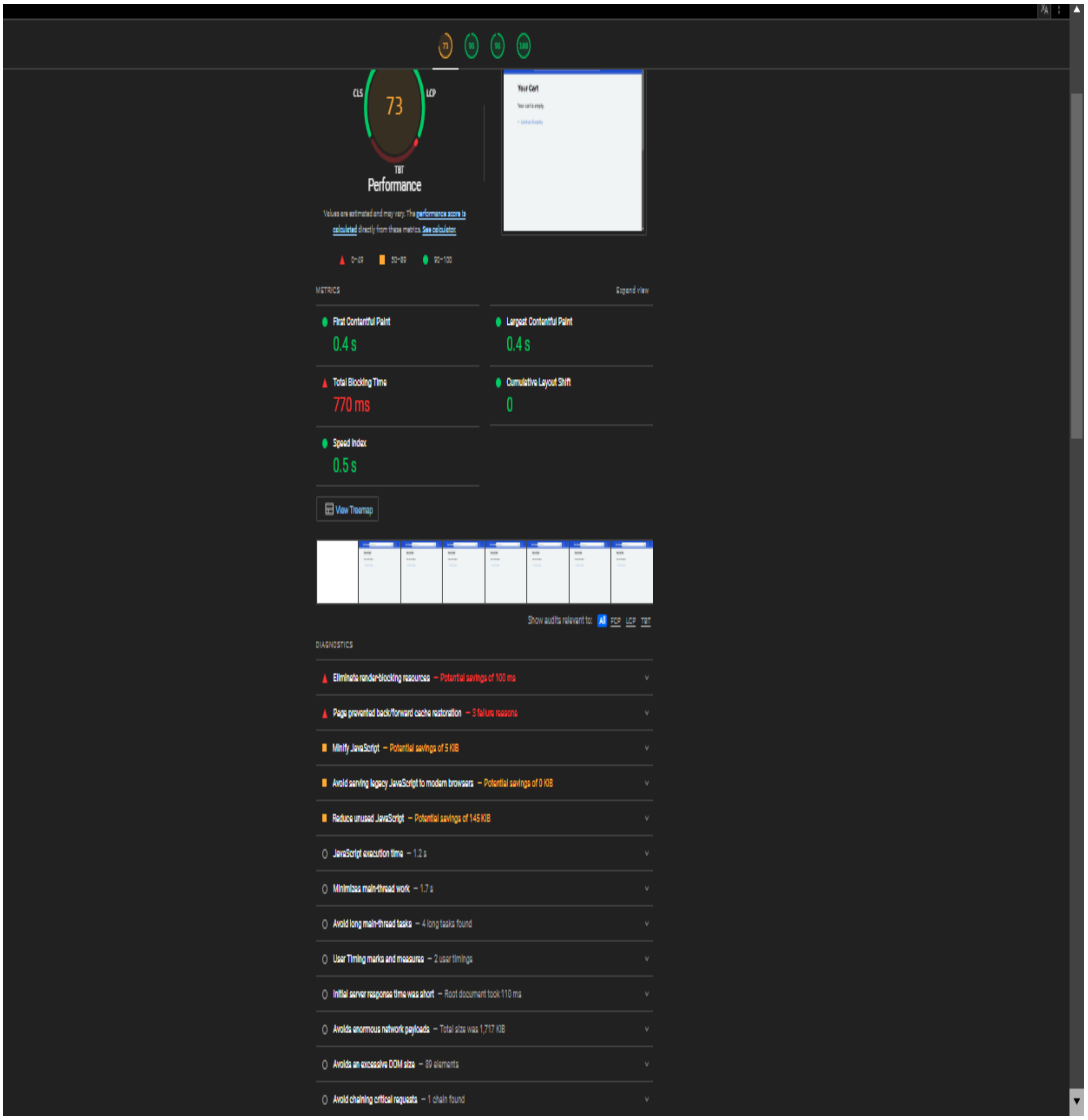
2. **Lazy Loading**:

   Used **lazy loading** for images to load them only when needed, improving page speed.

   Install **next-optimized-images** for better image optimization:

   npm install next-optimized-images

Lighthouse Performance Report:



73

Performance

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

▲ 0–49 ■ 50–89 ● 90–100

METRICS

Expand view

● First Contentful Paint 0.4 s	● Largest Contentful Paint 0.4 s
▲ Total Blocking Time 770 ms	● Cumulative Layout Shift 0
● Speed Index 0.5 s	

Show audits relevant to: All FCP LCP TBT

DIAGNOSTICS

▲ Eliminate render-blocking resources — Potential savings of 100 ms

▲ Page prevented back/forward cache restoration — 3 failure reasons

■ Minify JavaScript — Potential savings of 5 KiB

■ Avoid serving legacy JavaScript to modern browsers — Potential savings of 0 KiB

■ Reduce unused JavaScript — Potential savings of 145 KiB

○ JavaScript execution time — 1.2 s

○ Minimizes main-thread work — 1.7 s

○ Avoid long main-thread tasks — 4 long tasks found

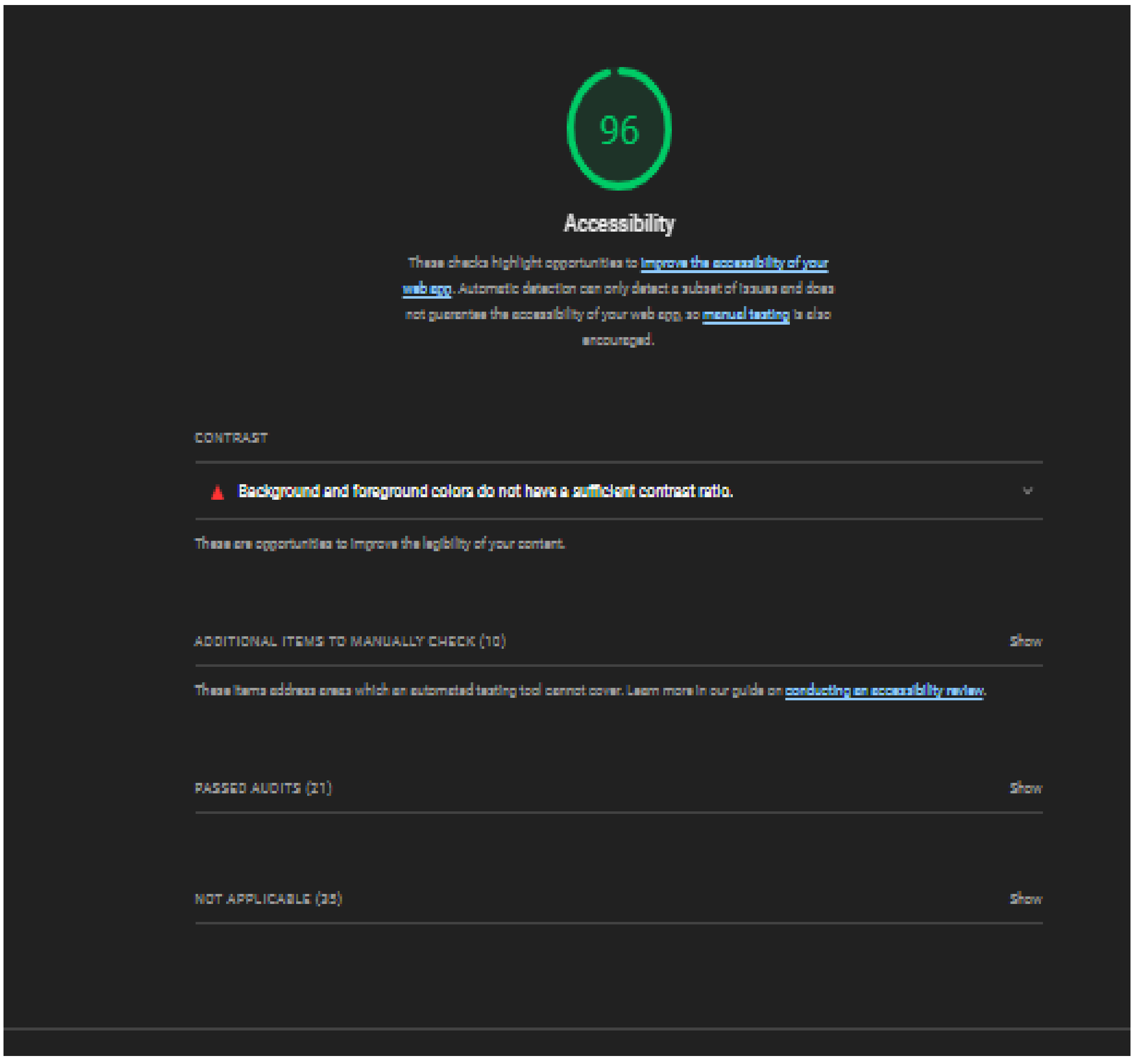
○ User Timing marks and measures — 2 user timings

○ Initial server response time was short — Root document took 110 ms

○ Avoids enormous network payloads — Total size was 1,717 KiB

○ Avoids an excessive DOM size — 89 elements

○ Avoid chaining critical requests — 1 chain found


96

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

CONTRAST

▲ Background and foreground colors do not have a sufficient contrast ratio.

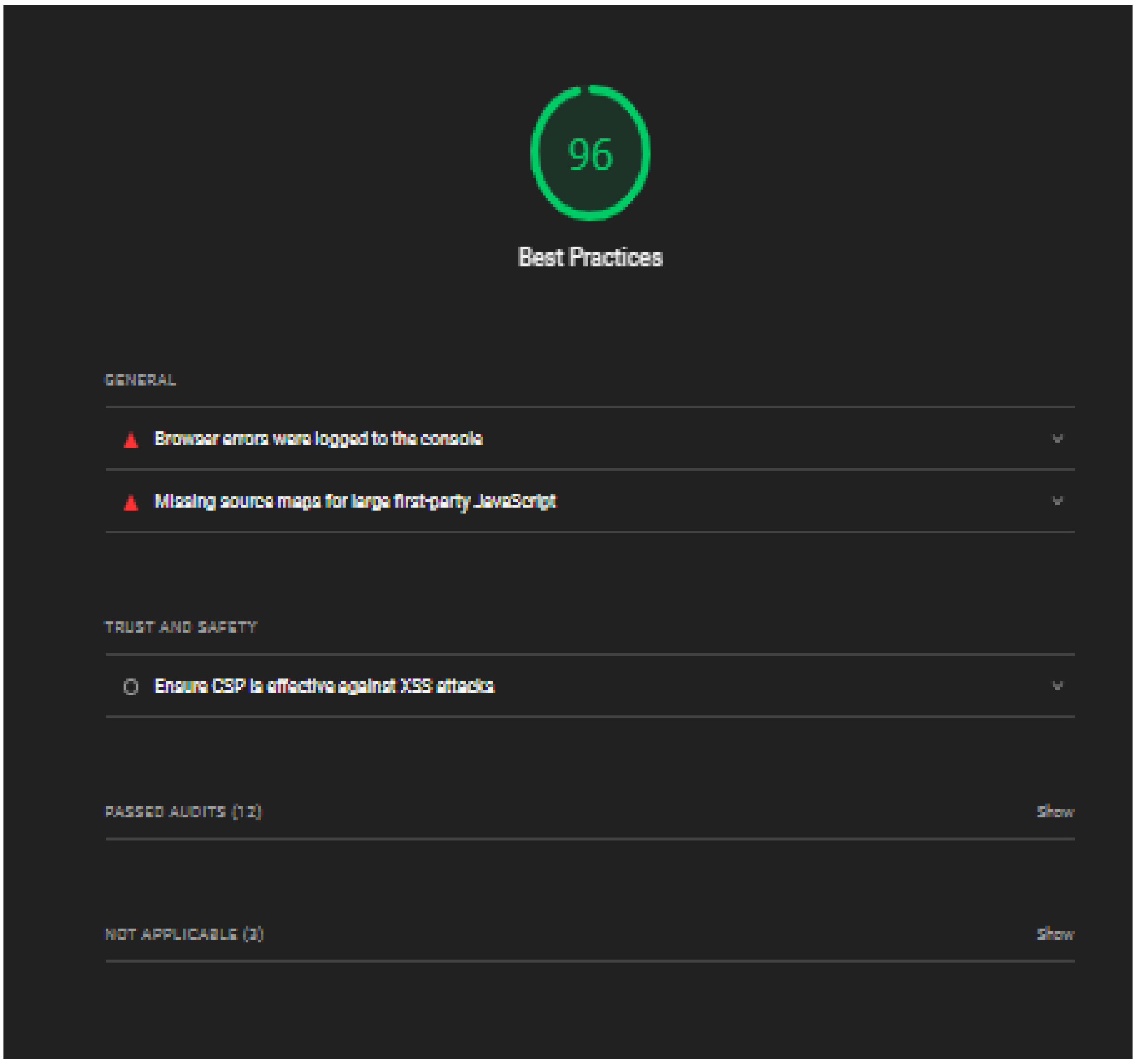
These are opportunities to improve the legibility of your content.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (21) Show

NOT APPLICABLE (35) Show



96

Best Practices

GENERAL

▲ Browser errors were logged to the console

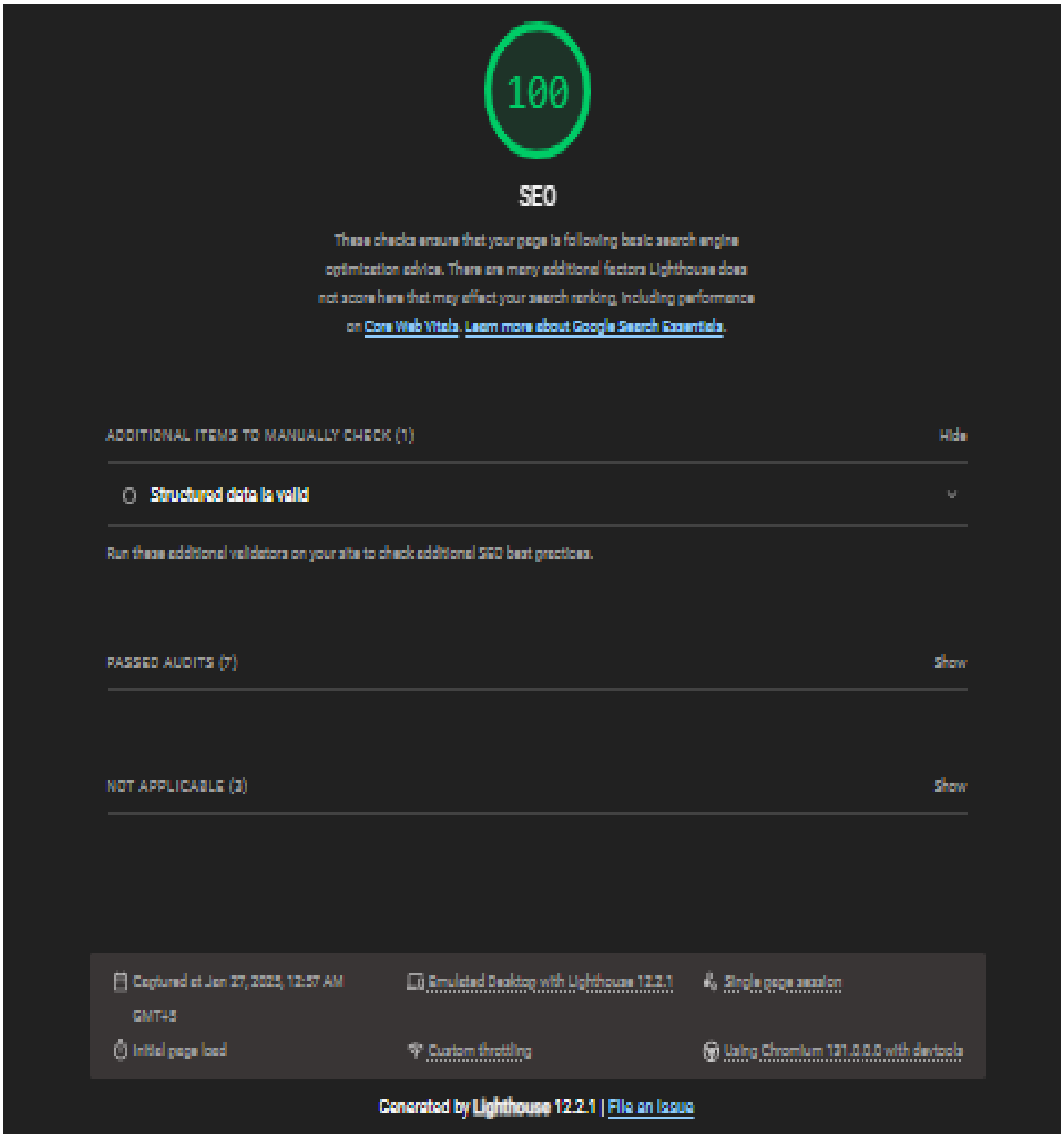
▲ Missing source maps for large first-party JavaScript

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks

PASSED AUDITS (13) Show

NOT APPLICABLE (3) Show



100

SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on Core Web Vitals. Learn more about Google Search Essentials.

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Hide

Structured data is valid

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (7) Show

NOT APPLICABLE (3) Show

Captured at Jan 27, 2025, 12:57 AM GMT+5

Emulated Desktop with Lighthouse 12.2.1

Single page session

Initial page load

Custom throttling

Using Chromium 131.0.0.0 with devtools

Generated by Lighthouse 12.2.1 | File an issue

5. Security Testing

Test Case ID	Description	Test Steps	Expected Result	Actual Result	Status	Severity	Tools Used

ST-001	Secure API communicati on and input validation	- Check if API calls are made over HTTPS. - Test inputs for SQL injection or XSS vulnerabilities.	All API calls over HTTPS and inputs properly validated.	All API requests are secure, and inputs are sanitized.	Passed	High	OWASP ZAP

- **Remarks:** Security testing completed successfully with no vulnerabilities detected.
- **Tools Used:**
    - **OWASP ZAP**: For vulnerability scanning.

6. User Acceptance Testing (UAT)

Test Case ID	Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
UAT-001	Test browsing products and checkout	1. Browse products. 2. Add to cart. 3. Proceed to checkout.	Seamless browsing and checkout process.	Browsing and checkout successful.	Passed	Low	No issues. Ready for release.

Checklist for Day 5

CheckList Items	Status
Functional Testing	✔

Error Handling	✔
Performance Optimization	✔
Cross-Browser and Device Testing	✔
Security Testing	✔
Documentation	✔
Final Review	✔