

10/11

7/8

Performance

Best Practices

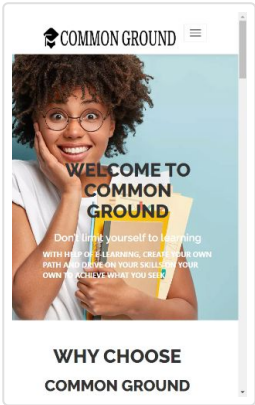
10/11

Performance

 0–49

50–89


90–100



METRICS

Expand view

<div>Total Blocking Time</div> <div>0 ms</div>	<div>Cumulative Layout Shift</div> <div>0</div>
<div>Interaction to Next Paint</div> <div>--</div>	

 [View Treemap](#)

[View Trace](#)

Show audits relevant to:


All

TBT

CLS

INP

DIAGNOSTICS

 Image elements do not have explicit [width](#) and [height](#)

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn how to set image dimensions](#)

CLS

URL	
img	...media/why-choose-us.5ec7e3b....png (common-ground.netlify.app)

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (25)

Hide

<div><div></div><div>Properly size images</div><div></div></div>
<div>Serve images that are appropriately-sized to save cellular data and improve load time. Learn how to size images.</div>
<div><div></div><div>Minify CSS</div><div></div></div>
<div>Minifying CSS files can reduce network payload sizes. Learn how to minify CSS.</div>
<div><div></div><div>Minify JavaScript</div><div></div></div>
<div>Minifying JavaScript files can reduce payload sizes and script parse time. Learn how to minify JavaScript.</div>
<div><div></div><div>Reduce unused CSS</div><div></div></div>
<div>Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. Learn how to reduce unused CSS.</div>
<div><div></div><div>Reduce unused JavaScript</div><div></div></div>
<div>Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. Learn how to reduce unused JavaScript.</div>
<div><div></div><div>Efficiently encode images</div><div></div></div>
<div>Optimized images load faster and consume less cellular data. Learn how to efficiently encode images.</div>
<div><div></div><div>Serve images in next-gen formats</div><div></div></div>
<div>Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. Learn more about modern image formats.</div>
<div><div></div><div>Enable text compression</div><div></div></div>
<div>Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. Learn more about text compression.</div>

Use HTTP/2



HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more about HTTP/2.](#)

☐ Use video formats for animated content

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more about efficient video formats](#)

☐ Remove duplicate modules in JavaScript bundles

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity.

TBT

☐ Avoid serving legacy JavaScript to modern browsers

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. [Learn how to use modern JavaScript](#) TBT

Avoids enormous network payloads — Total size was 0 KiB



Large network payloads cost users real money and are highly correlated with long load times. [Learn how to reduce payload sizes.](#)

Uses efficient cache policy on static assets — 0 resources found



A long cache lifetime can speed up repeat visits to your page. [Learn more about efficient cache policies.](#)

☐ User Timing marks and measures

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. [Learn more about User Timing marks.](#)

JavaScript execution time



Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn how to reduce Javascript execution time.](#) TBT

Minimizes main-thread work — 0.0 s



Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. [Learn how to minimize main-thread work](#) TBT

Category	Time Spent
Script Evaluation	2 ms
Other	2 ms

Minimize third-party usage

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading. [Learn how to minimize third-party impact.](#) TBT

Avoid large layout shifts

These DOM elements contribute most to the CLS of the page. [Learn how to improve CLS](#) CLS

Uses passive listeners to improve scrolling performance

Consider marking your touch and wheel event listeners as passive to improve your page's scroll performance. [Learn more about adopting passive event listeners.](#)

Avoids `document.write()`

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. [Learn how to avoid document.write\(\)](#).

Avoid long main-thread tasks

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay. [Learn how to avoid long main-thread tasks](#) TBT

Avoid non-composited animations

Animations which are not composited can be janky and increase CLS. [Learn how to avoid non-composited animations](#) CLS

Minimizes work during key interaction

This is the thread-blocking work occurring during the Interaction to Next Paint measurement. [Learn more about the Interaction to Next Paint metric.](#) INP

Page didn't prevent back/forward cache restoration

Many navigations are performed by going back to a previous page, or forwards again. The back/forward cache (bfcache) can speed up these return navigations. [Learn more about the bfcache](#)

about:blank

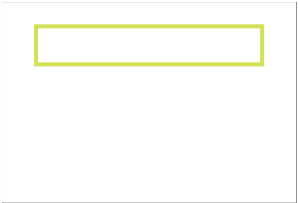
4/6

Best Practices

USER EXPERIENCE

▲ Displays images with incorrect aspect ratio

Image display dimensions should match natural aspect ratio. [Learn more about image aspect ratio.](#)

	URL	Aspect Ratio (Displayed)	Aspect Ratio (Actual)
 img	...media/Logo.97b8712....png (common-ground.netlify.app)	254 x 40 (6.35)	2000 x 186 (10.75)

PASSED AUDITS (7)

Hide

Uses HTTPS

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more about HTTPS.](#)

Avoids deprecated APIs

Deprecated APIs will eventually be removed from the browser. [Learn more about deprecated APIs.](#)

Serves images with appropriate resolution

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn how to provide responsive images.](#)

Avoids `unload` event listeners

The unload event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use pagehide or visibilitychange events instead. Learn more about unload event listeners	
No browser errors logged to the console	^
Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. Learn more about this errors in console diagnostic audit	
No issues in the Issues panel in Chrome Devtools	^
Issues logged to the Issues panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.	
Page has valid source maps	^
Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. Learn more about source maps .	

NOT APPLICABLE (1)

Hide

<input type="radio"/> Fonts with font-display: optional are preloaded	^
Preload optional fonts so first-time visitors may use them. Learn more about preloading fonts	

Captured at Dec 2, 2023, 6:51 PM GMT+5:30

Initial page load

Emulated Desktop with Lighthouse 11.1.0

Custom throttling

Single page load

Using Chromium 119.0.0.0 with devtools