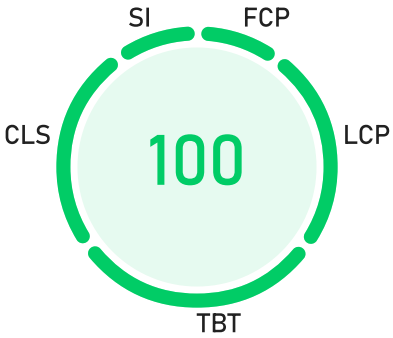
https://proofstrike.io/

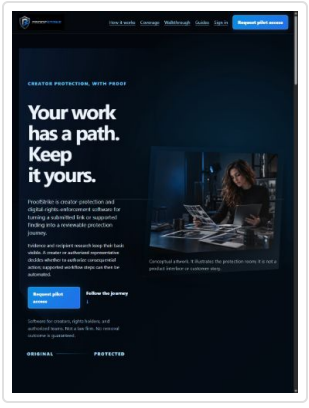
100	100	100	100	2/2
Performance	Accessibility	Best Practices	SEO	Agentic Browsing



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 Largest Contentful Paint

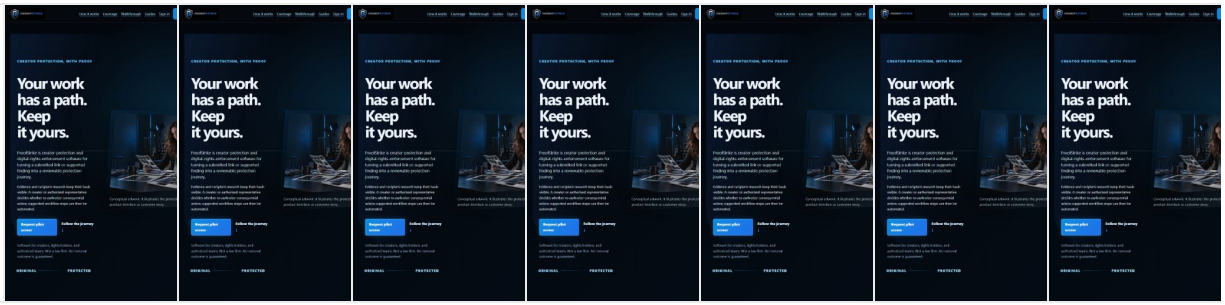
0.5 s

0.5 s

Total Blocking Time
0 ms

Cumulative Layout Shift
0

Speed Index
0.5 s



Show audits relevant to: All FCP LCP

INSIGHTS

▲ Network dependency tree

Avoid chaining critical requests by reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. LCP Unscored

Maximum critical path latency: **414 ms**

Initial Navigation

https://proofstrike.io - **221 ms,** 5.38 KiB

/beacon.min.js/vce70182... (static.cloudflareinsights.com) - **315 ms,** 10.23 KiB

/cdn-cgi/rum (proofstrike.io) - **414 ms,** 0.46 KiB

/assets/site.a5bfa348a740.css (proofstrike.io) - **277 ms,** 5.20 KiB

...cloudflare-static/email-decode.min.js (proofstrike.io) - **269 ms,** 1.04 KiB

Preconnected origins

preconnect hints help the browser establish a connection earlier in the page load, saving time when the first request for that origin is made. The following are the origins that the page preconnected to.

no origins were preconnected

Preconnect candidates

Add preconnect hints to your most important origins, but try to use no more than 4.

No additional origins are good candidates for preconnecting

Use efficient cache lifetimes — Est savings of 4 KiB

A long cache lifetime can speed up repeat visits to your page. Learn more about caching. FCP LCP Unscored

Show 3rd-party resources (1)

Request	Cache TTL	Transfer Size
Cloudflare Utility		**10 KiB**
/beacon.min.js/vce70182... (static.cloudflareinsights.com)	1d	10 KiB
proofstrike.io 1st Party		**1 KiB**
...cloudflare-static/email-decode.min.js (proofstrike.io)	2d 1m 17s	1 KiB

Improve image delivery — Est savings of 136 KiB

Reducing the download time of images can improve the perceived load time of the page and LCP. Learn more about optimizing image size FCP LCP Unscored

	URL	Resource Size	Est Savings
proofstrike.io 1st Party		**221.0 KiB**	**136.4 KiB**
img	/assets/creator-concept.webp (proofstrike.io)	64.9 KiB	50.6 KiB
	This image file is larger than it needs to be (1157x941) for its displayed		50.6 KiB

	URL	Resource Size	Est Savings
	dimensions (653x368). Use responsive images to reduce the image download size.		
img	/assets/original-work-concept.webp (proofstrike.io)	110.4 KiB	46.4 KiB
	Increasing the image compression factor could improve this image's download size.		46.4 KiB
	/assets/proofstrike-lockup.png (proofstrike.io)	45.7 KiB	39.4 KiB
a.brand			
	Using a modern image format (WebP, AVIF) or increasing the image compression could improve this image's download size.		39.4 KiB

Render-blocking requests

Requests are blocking the page's initial render, which may delay LCP. Deferring or inlining can move these network requests out of the critical path. FCP LCP Unscored

URL	Transfer Size	Duration
proofstrike.io 1st Party	**5.2 KiB**	**0 ms**
/assets/site.a5bfa348a740.css (proofstrike.io)	5.2 KiB	

Legacy JavaScript — Est savings of 10 KiB

Polyfills and transforms enable older browsers to use new JavaScript features. However, many aren't necessary for modern browsers. Consider modifying your JavaScript build process to not transpile Baseline features, unless you know you must support older browsers. Learn why most sites can deploy ES6+ code without transpiling FCP LCP Unscored

URL	Wasted bytes
Cloudflare Utility	**10.4 KiB**
/beacon.min.js/vce70182... (static.cloudflareinsights.com)	10.4 KiB
vce701823e78c4301a35…87301792088814092:1 — Array.prototype.at	
vce701823e78c4301a35…87301792088814092:1 — Array.prototype.findLast	
vce701823e78c4301a35…87301792088814092:1 — Array.prototype.includes	

LCP breakdown

Each subpart has specific improvement strategies. Ideally, most of the LCP time should be spent on loading the resources, not within delays. LCP Unscored

Subpart	Duration
Time to first byte	150 ms
Resource load delay	70 ms
Resource load duration	50 ms
Element render delay	80 ms

img

3rd parties

3rd party code can significantly impact load performance. Reduce and defer loading of 3rd party code to prioritize your page's content. Unscored

3rd party	Transfer size	Main thread time
njgehaondchbmjmajphnhlojfnbfokng	**0 KiB**	**15 ms**

3rd party	Transfer size	Main thread time
chrome-extension://njgehaondchbmjmajphnhlojfnbfokng/assets/extra/jquery.js	0 KiB	14 ms
chrome-extension://njgehaondchbmjmajphnhlojfnbfokng/scripts/content.build.min.js	0 KiB	1 ms
chrome-extension://njgehaondchbmjmajphnhlojfnbfokng/scripts/dom.build.min.js	0 KiB	0 ms
Cloudflare Utility	**10 KiB**	**3 ms**
/beacon.min.js/vce70182... (static.cloudflareinsights.com)	10 KiB	3 ms
einnffiilpmgldkapbikhkeicohlaapj	**0 KiB**	**1 ms**
chrome-extension://einnffiilpmgldkapbikhkeicohlaapj/js/content.js	0 KiB	1 ms
hlbhaaegomldlibkeiiifaejlciaifmj	**0 KiB**	**1 ms**
chrome-extension://hlbhaaegomldlibkeiiifaejlciaifmj/js/content.js	0 KiB	1 ms

These insights are also available in the Chrome DevTools Performance Panel - record a trace to view more detailed information.

DIAGNOSTICS

Minify JavaScript — Est savings of 41 KiB

Minifying JavaScript files can reduce payload sizes and script parse time. Learn how to minify JavaScript. FCP LCP Unscored

URL	Transfer Size	Est Savings
Video Downloader PLUS Chrome Extension	**86.3 KiB**	**41.4 KiB**

URL	Transfer Size	Est Savings
chrome-extension://njgehaondchbmjmajphnhlojfnbfokng/assets/extra/jquery.js	86.3 KiB	41.4 KiB

Reduce unused JavaScript — Est savings of 60 KiB

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. Learn how to reduce unused JavaScript. FCP LCP Unscored

URL	Transfer Size	Est Savings
Video Downloader PLUS Chrome Extension	**86.3 KiB**	**59.6 KiB**
chrome-extension://njgehaondchbmjmajphnhlojfnbfokng/assets/extra/jquery.js	86.3 KiB	59.6 KiB

More information about the performance of your application. These numbers don't directly affect the Performance score.

PASSED AUDITS (20) Hide

Layout shift culprits

Layout shifts occur when elements move absent any user interaction. Investigate the causes of layout shifts, such as elements being added, removed, or their fonts changing as the page loads. CLS Unscored

Document request latency

Your first network request is the most important. Reduce its latency by avoiding redirects, ensuring a fast server response, and enabling text compression. FCP LCP Unscored

Avoids redirects

Server responds quickly (observed 147 ms)

Applies text compression

Optimize DOM size

A large DOM can increase the duration of style calculations and layout reflows, impacting page responsiveness. A large DOM will also increase memory usage. Learn how to avoid an excessive DOM size. Unscored

Statistic	Element	Value
Total elements		149
DOM depth	span	7
Most children	main#main	8

Duplicated JavaScript

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

Font display

Consider setting **font-display** to **swap** or **optional** to ensure text is consistently visible. **swap** can be further optimized to mitigate layout shifts with font metric overrides. Unscored

Forced reflow

A forced reflow occurs when JavaScript queries geometric properties (such as **offsetWidth**) after styles have been invalidated by a change to the DOM state. This can result in poor performance. Learn more about forced reflows and possible mitigations. Unscored

INP breakdown

Start investigating how to improve INP by looking at the longest subpart. Unscored

LCP request discovery

Optimize LCP by making the LCP image discoverable from the HTML immediately, and avoiding lazy-loading LCP Unscored

fetchpriority=high applied

Request is discoverable in initial document

LCP resources should not use loading=lazy

img

Modern HTTP

HTTP/2 and HTTP/3 offer many benefits over HTTP/1.1, such as multiplexing. Learn more about using modern HTTP. FCP LCP Unscored

Optimize viewport for mobile

Tap interactions may be delayed by up to 300 ms if the viewport is not optimized for mobile. Unscored

meta

Minify CSS

Minifying CSS files can reduce network payload sizes. Learn how to minify CSS. FCP LCP Unscored

Reduce unused CSS

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. FCP LCP Unscored

Avoids enormous network payloads — Total size was 247 KiB

Large network payloads cost users real money and are highly correlated with long load times. Learn how to reduce payload sizes. Unscored

Show 3rd-party resources (1)

URL	Transfer Size
proofstrike.io 1st Party	**236.4 KiB**
/assets/original-work-concept.webp (proofstrike.io)	110.9 KiB
/assets/creator-concept.webp (proofstrike.io)	65.4 KiB
/assets/proofstrike-lockup.png (proofstrike.io)	46.2 KiB
https://proofstrike.io	5.4 KiB
/assets/site.a5bfa348a740.css (proofstrike.io)	5.2 KiB
/assets/journey.f133f31fe15f.js (proofstrike.io)	1.1 KiB
...cloudflare-static/email-decode.min.js (proofstrike.io)	1.0 KiB
/favicon.ico (proofstrike.io)	0.7 KiB
/cdn-cgi/rum (proofstrike.io)	0.5 KiB
Cloudflare Utility	**10.2 KiB**
/beacon.min.js/vce70182... (static.cloudflareinsights.com)	10.2 KiB

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. Unscored

JavaScript execution time — 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 reduce Javascript execution time. TBT Unscored

URL	Total CPU Time	Script Evaluation	Script Parse
Unattributable	**128 ms**	**1 ms**	**0 ms**
Unattributable	128 ms	1 ms	0 ms
proofstrike.io 1st Party	**79 ms**	**8 ms**	**0 ms**
https://proofstrike.io	79 ms	8 ms	0 ms

Minimizes main-thread work — 0.2 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 Unscored

Category	Time Spent
Other	149 ms
Style & Layout	43 ms
Script Evaluation	22 ms
Script Parsing & Compilation	9 ms
Rendering	7 ms
Parse HTML & CSS	2 ms

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 Unscored

Avoid non-composited animations

Animations which are not composited can be janky and increase CLS. Learn how to avoid non-composited animations CLS Unscored

Image elements 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 Unscored

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 Unscored



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.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

Interactive controls are keyboard focusable

Custom interactive controls are keyboard focusable and display a focus indicator. Learn how to make custom controls focusable. Unscored

Interactive elements indicate their purpose and state

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. Learn how to decorate interactive elements with affordance hints. Unscored

The page has a logical tab order

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. Learn more about logical tab ordering. Unscored

- Visual order on the page follows DOM order

  DOM order matches the visual order, improving navigation for assistive technology. Learn more about DOM and visual ordering. Unscored

- User focus is not accidentally trapped in a region

  A user can tab into and out of any control or region without accidentally trapping their focus. Learn how to avoid focus traps. Unscored

- The user's focus is directed to new content added to the page

  If new content, such as a dialog, is added to the page, the user's focus is directed to it. Learn how to direct focus to new content. Unscored

- HTML5 landmark elements are used to improve navigation

  Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. Learn more about landmark elements. Unscored

- Offscreen content is hidden from assistive technology

  Offscreen content is hidden with display: none or aria-hidden=true. Learn how to properly hide offscreen content. Unscored

- Custom controls have associated labels

  Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. Learn more about custom controls and labels. Unscored

- Custom controls have ARIA roles

  Custom interactive controls have appropriate ARIA roles. Learn how to add roles to custom controls. Unscored

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

PASSED AUDITS (25) Hide

- [aria-*] attributes match their roles

Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. Learn how to match ARIA attributes to their roles.

`[aria-hidden="true"]` is not present on the document `<body>`

Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document `<body>`. Learn how `aria-hidden` affects the document body.

`[role]`s have all required `[aria-*]` attributes

Some ARIA roles have required attributes that describe the state of the element to screen readers. Learn more about roles and required attributes.

Elements with an ARIA `[role]` that require children to contain a specific `[role]` have all required children.

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. Learn more about roles and required children elements.

`[role]`s are contained by their required parent element

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. Learn more about ARIA roles and required parent element.

`[role]` values are valid

ARIA roles must have valid values in order to perform their intended accessibility functions. Learn more about valid ARIA roles.

`[aria-*]` attributes have valid values

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. Learn more about valid values for ARIA attributes.

`[aria-*]` attributes are valid and not misspelled

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. Learn more about valid ARIA attributes.

Buttons have an accessible name

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. Learn how to make buttons more accessible.

Image elements have [alt] attributes

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. Learn more about the alt attribute.

[user-scalable="no"] is not used in the <meta name="viewport"> element and the [maximum-scale] attribute is not less than 5.

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. Learn more about the viewport meta tag.

ARIA attributes are used as specified for the element's role

Some ARIA attributes are only allowed on an element under certain conditions. Learn more about conditional ARIA attributes.

[aria-hidden="true"] elements do not contain focusable descendents

Focusable descendents within an [aria-hidden="true"] element prevent those interactive elements from being available to users of assistive technologies like screen readers. Learn how aria-hidden affects focusable elements.

Elements use only permitted ARIA attributes

Using ARIA attributes in roles where they are prohibited can mean that important information is not communicated to users of assistive technologies. Learn more about prohibited ARIA roles.

Background and foreground colors have a sufficient contrast ratio

Low-contrast text is difficult or impossible for many users to read. Learn how to provide sufficient color contrast.

Document has a <title> element

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. Learn more about document titles.

`<html>` element has a `[lang]` attribute

If a page doesn't specify a `lang` attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. Learn more about the `lang` attribute.

`<html>` element has a valid value for its `[lang]` attribute

Specifying a valid BCP 47 language helps screen readers announce text properly. Learn how to use the `lang` attribute.

Links have a discernible name

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. Learn how to make links accessible.

No element has a `[tabindex]` value greater than 0

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. Learn more about the `tabindex` attribute.

Touch targets have sufficient size and spacing.

Touch targets with sufficient size and spacing help users who may have difficulty targeting small controls to activate the targets. Learn more about touch targets.

Heading elements appear in a sequentially-descending order

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. Learn more about heading order.

Skip links are focusable.

Including a skip link can help users skip to the main content to save time. Learn more about skip links.

Document has a main landmark.

One main landmark helps screen reader users navigate a web page. Learn more about landmarks.

Deprecated ARIA roles were not used

Deprecated ARIA roles may not be processed correctly by assistive technology. Learn more about deprecated ARIA roles.

NOT APPLICABLE (38) Hide

- [accesskey] values are unique

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. Learn more about access keys. Unscored

- button, link, and menuitem elements have accessible names

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn how to make command elements more accessible. Unscored

- Elements with role="dialog" or role="alertdialog" have accessible names.

ARIA dialog elements without accessible names may prevent screen readers users from discerning the purpose of these elements. Learn how to make ARIA dialog elements more accessible. Unscored

- ARIA input fields have accessible names

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more about input field labels. Unscored

- ARIA meter elements have accessible names

When a meter element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn how to name meter elements. Unscored

- ARIA progressbar elements have accessible names

When a **progressbar** element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn how to label **progressbar** elements. Unscored

- Elements with the **role=text** attribute do not have focusable descendents.

Adding **role=text** around a text node split by markup enables VoiceOver to treat it as one phrase, but the element's focusable descendents will not be announced. Learn more about the **role=text** attribute. Unscored

- ARIA toggle fields have accessible names

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more about toggle fields. Unscored

- ARIA **tooltip** elements have accessible names

When a tooltip element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn how to name **tooltip** elements. Unscored

- ARIA **treeitem** elements have accessible names

When a **treeitem** element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more about labeling **treeitem** elements. Unscored

- The page contains a heading, skip link, or landmark region

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. Learn more about bypass blocks. Unscored

- **<dl>**'s contain only properly-ordered **<dt>** and **<dd>** groups, **<script>**, **<template>** or ** ** elements.

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. Learn how to structure definition lists correctly. Unscored

- Definition list items are wrapped in **<dl>** elements

Definition list items (**<dt>** and **<dd>**) must be wrapped in a parent **<dl>** element to ensure that screen readers can properly announce them. Learn how to structure definition lists correctly.

Unscored

- ARIA IDs are unique

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. Learn how to fix duplicate ARIA IDs. Unscored

- No form fields have multiple labels

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. Learn how to use form labels. Unscored

- `<frame>` or `<iframe>` elements have a title

Screen reader users rely on frame titles to describe the contents of frames. Learn more about frame titles. Unscored

- `<html>` element has an `[xml:lang]` attribute with the same base language as the `[lang]` attribute.

If the webpage does not specify a consistent language, then the screen reader might not announce the page's text correctly. Learn more about the `lang` attribute. Unscored

- Input buttons have discernible text.

Adding discernable and accessible text to input buttons may help screen reader users understand the purpose of the input button. Learn more about input buttons. Unscored

- `<input type="image">` elements have `[alt]` text

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. Learn about input image alt text. Unscored

- Form elements have associated labels

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. Learn more about form element labels. Unscored

- Links are distinguishable without relying on color.

Low-contrast text is difficult or impossible for many users to read. Link text that is discernible improves the experience for users with low vision. Learn how to make links distinguishable. Unscored

- Lists contain only <li> elements and script supporting elements (<script> and <template>).

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. Learn more about proper list structure. Unscored

- List items (<li>) are contained within <ul>, <ol> or <menu> parent elements

Screen readers require list items (<li>) to be contained within a parent <ul>, <ol> or <menu> to be announced properly. Learn more about proper list structure. Unscored

- The document does not use <meta http-equiv="refresh">

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. Learn more about the refresh meta tag. Unscored

- <object> elements have alternate text

Screen readers cannot translate non-text content. Adding alternate text to <object> elements helps screen readers convey meaning to users. Learn more about alt text for object elements. Unscored

- Select elements have associated label elements.

Form elements without effective labels can create frustrating experiences for screen reader users. Learn more about the select element. Unscored

- Cells in a <table> element that use the [headers] attribute refer to table cells within the same table.

Screen readers have features to make navigating tables easier. Ensuring <td> cells using the [headers] attribute only refer to other cells in the same table may improve the experience for screen reader users. Learn more about the headers attribute. Unscored

- <th> elements and elements with [role="columnheader"/"rowheader"] have data cells they describe.

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. Learn more

about table headers. Unscored

- [lang] attributes have a valid value

  Specifying a valid BCP 47 language on elements helps ensure that text is pronounced correctly by a screen reader. Learn how to use the lang attribute. Unscored

- <video> elements contain a <track> element with [kind="captions"]

  When a video provides a caption it is easier for deaf and hearing impaired users to access its information. Learn more about video captions. Unscored

- autocomplete attributes are used correctly

  The autocomplete attribute values must be valid and correctly applied for screen readers to function correctly. Learn more about valid autocomplete values. Unscored

- Elements with role="none" or role="presentation" do not have conflicts

  There are certain cases where the semantic role of an element with role="none" or role="presentation" does not resolve to none or presentation. To ensure the element remains removed from the accessibility tree, you should not add any global ARIA attributes to the element or make it focusable. Learn more about presentation role conflict. Unscored

- SVG elements with an img role have an accessible text alternative

  Ensures SVG elements with an img, graphics-document or graphics-symbol role have an accessible text alternative. Learn more about SVG alt text. Unscored

- Tables have different content in the summary attribute and <caption>.

  The summary attribute should describe the table structure, while <caption> should have the onscreen title. Accurate table mark-up helps users of screen readers. Learn more about summary and caption. Unscored

- All heading elements contain content.

  A heading with no content or inaccessible text prevent screen reader users from accessing information on the page's structure. Learn more about headings. Unscored

- Uses ARIA roles only on compatible elements

Many HTML elements can only be assigned certain ARIA roles. Using ARIA roles where they are not allowed can interfere with the accessibility of the web page. Learn more about ARIA roles. Unscored

- Image elements do not have [alt] attributes that are redundant text.

Informative elements should aim for short, descriptive alternative text. Alternative text that is exactly the same as the text adjacent to the link or image is potentially confusing for screen reader users, because the text will be read twice. Learn more about the alt attribute. Unscored

- Identical links have the same purpose.

Links with the same destination should have the same description, to help users understand the link's purpose and decide whether to follow it. Learn more about identical links. Unscored



Best Practices

TRUST AND SAFETY

- Ensure CSP is effective against XSS attacks

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. Learn how to use a CSP to prevent XSS Unscored

Description	Directive	Severity
No CSP found in enforcement mode		High

- Use a strong HSTS policy

Deployment of the HSTS header significantly reduces the risk of downgrading HTTP connections and eavesdropping attacks. A rollout in stages, starting with a low max-age is recommended. Learn more about using a strong HSTS policy. Unscored

Description	Directive	Severity
No HSTS header found		High

Ensure proper origin isolation with COOP

The Cross-Origin-Opener-Policy (COOP) can be used to isolate the top-level window from other documents such as pop-ups. Learn more about deploying the COOP header. Unscored

Description	Directive	Severity
No COOP header found		High

Mitigate clickjacking with XFO or CSP

The **X-Frame-Options** (XFO) header or the **frame-ancestors** directive in the **Content-Security-Policy** (CSP) header control where a page can be embedded. These can mitigate clickjacking attacks by blocking some or all sites from embedding the page. Learn more about mitigating clickjacking. Unscored

Description	Severity
No frame control policy found	High

Mitigate DOM-based XSS with Trusted Types

The **require-trusted-types-for** directive in the **Content-Security-Policy** (CSP) header instructs user agents to control the data passed to DOM XSS sink functions. Learn more about mitigating DOM-based XSS with Trusted Types. Unscored

Description	Severity
No `Content-Security-Policy` header with Trusted Types directive found	High

BROWSER COMPATIBILITY

Baseline Features

Lists web features used on the page and their Baseline status as of 2026-09-17. Learn more about Baseline. Unscored

Show 3rd-party resources (3)

Web-features	Baseline Status	Source
proofstrike.io 1st Party		
page-visibility-state	Limited Availability	(index):2
interaction-contentful-paint	Limited Availability	(index):2
soft-navigations	Limited Availability	(index):2
long-animation-frames	Limited Availability	(index):2
layout-instability	Limited Availability	(index):2
requestidlecallback	Limited Availability	(index):2
zstd	Newly Available (2026-02-11)	(index):-1
largest-contentful-paint	Newly Available (2025-12-12)	(index):2
event-timing	Newly Available (2025-12-12)	(index):2
fetch-priority	Newly Available (2024-10-29)	(index):-1

Web-features	Baseline Status	Source
text-wrap	Newly Available (2024-10-17)	(index):-1
backdrop-filter	Newly Available (2024-09-16)	(index):-1
text-wrap-balance	Newly Available (2024-05-13)	(index):-1
overflow-clip	Widely Available (2022-09-12)	(index):-1
array-findlast	Widely Available (2022-08-23)	(index):2
color-scheme	Widely Available (2022-02-03)	(index):-1
aspect-ratio	Widely Available (2021-09-20)	(index):-1
weak-references	Widely Available (2021-04-26)	(index):2
not	Widely Available (2021-01-21)	(index):-1
clip-path	Widely Available (2021-01-21)	(index):-1
text-underline-offset	Widely Available (2020-11-19)	(index):-1

Web-features	Baseline Status	Source
nullish-coalescing	Widely Available (2020-09-16)	(index):1
webp	Widely Available (2020-09-16)	(index):-1
aborting	Widely Available (2019-03-25)	(index):2
js-modules	Widely Available (2018-05-09)	(index):-1
grid	Widely Available (2017-10-17)	(index):-1
flexbox	Widely Available (2015-09-30)	(index):-1
ch	Widely Available (2015-07-29)	(index):-1
Cloudflare Utility		
ua-client-hints	Limited Availability	vce701823e78c4301a35…87301792088814092:1
Video Downloader PLUS Chrome Extension		
background-clip-text	Limited Availability	jquery.js:5626
slot	Widely Available (2020-01-15)	jquery.js:1175

PASSED AUDITS (13) 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.

Avoids third-party cookies

Third-party cookies may be blocked in some contexts. Learn more about preparing for third-party cookie restrictions.

Allows users to paste into input fields

Preventing input pasting is a bad practice for the UX, and weakens security by blocking password managers.Learn more about user-friendly input fields.

Avoids requesting the geolocation permission on page load

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. Learn more about the geolocation permission. Also consider using the `<geolocation>` element.

Avoids requesting the notification permission on page load

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. Learn more about responsibly getting permission for notifications.

Displays images with correct aspect ratio

Image display dimensions should match natural aspect ratio. Learn more about image aspect ratio.

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.

Page has the HTML doctype

Specifying a doctype prevents the browser from switching to quirks-mode. Learn more about the doctype declaration.

Properly defines charset

A character encoding declaration is required. It can be done with a **<meta>** tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. Learn more about declaring the character encoding.

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. Unscored

NOT APPLICABLE (2) Hide

Redirects HTTP traffic to HTTPS

Make sure that you redirect all HTTP traffic to HTTPS in order to enable secure web features for all your users. Learn more. Unscored

Detected JavaScript libraries

All front-end JavaScript libraries detected on the page. Learn more about this JavaScript library detection diagnostic audit. Unscored



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 the Structured Data Testing Tool to validate structured data. Learn more about Structured Data. Unscored

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

PASSED AUDITS (10) Hide

Page isn't blocked from indexing

Search engines are unable to include your pages in search results if they don't have permission to crawl them. Learn more about crawler directives.

Document has a <title> element

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. Learn more about document titles.

Document has a meta description

Meta descriptions may be included in search results to concisely summarize page content. Learn more about the meta description.

Page has successful HTTP status code

Pages with unsuccessful HTTP status codes may not be indexed properly. Learn more about HTTP status codes.

Links have descriptive text

Descriptive link text helps search engines understand your content. Learn how to make links more accessible.

Links are crawlable

Search engines may use **href** attributes on links to crawl websites. Ensure that the **href** attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. Learn how to make links crawlable

robots.txt is valid

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. Learn more about robots.txt.

Image elements have [alt] attributes

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. Learn more about the **alt** attribute.

Document has a valid **hreflang**

hreflang links tell search engines what version of a page they should list in search results for a given language or region. Learn more about **hreflang**.

Document has a valid **rel=canonical**

Canonical links suggest which URL to show in search results. Learn more about canonical links.

2/2

Agentic Browsing

These checks ensure high-quality, browsable websites for AI agents and validate the correctness of WebMCP integrations. This category is still under development and subject to change.

PASSED AUDITS (2) Hide

Accessibility tree is well-formed — All audits passed

A well-formed accessibility tree helps AI agents to navigate and interact with the page.

Cumulative Layout Shift — 0

Cumulative Layout Shift measures the movement of visible elements within the viewport. Learn more about the Cumulative Layout Shift metric.

NOT APPLICABLE (5) Hide

WebMCP form coverage

Consider adding WebMCP annotations to the forms listed below. This helps AI agents identify and interact with these forms more reliably. Unscored

WebMCP tools registered

Lists the WebMCP tools registered at the time of analysis. Unscored

WebMCP schemas are valid

Valid WebMCP schemas are required for AI agents to understand and interact with tools correctly. Please fix any errors or warnings reported by the browser. Unscored

llms.txt follows recommendations

If your llms.txt file does not follow recommendations, large language models may not be able to understand how you want your website to be crawled or used for training. The llms.txt file should be a Markdown file containing at least one H1 header. Learn more about the llms.txt audit. Unscored

- `ai-catalog.json` schema is valid

Valid **ai-catalog.json** manifests are required for autonomous AI agents and registries to discover and verify your resources. Learn more about the ARD specification. Unscored

Captured at Oct 31, 2026, 3:19 PM EDT Initial page load	Emulated Desktop with Lighthouse 13.5.0 Custom throttling	Single page session Using Chromium 156.0.0.0 with devtools

Generated by **Lighthouse** 13.5.0 | File an issue