

Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0–49

50–89

90–100

Employees list					
First Name	Last Name	Start Date	Role	Health	Actions
John	Doe	2010-01-01	Manager	Green	Edit
Sarah	Hartley	2010-01-01	Analyst	Green	Edit
Anna	Simpson	2010-01-01	Analyst	Green	Edit
Carlos	Gonzalez	2010-01-01	Analyst	Green	Edit
Samantha	Jones	2010-01-01	Analyst	Green	Edit
Evan	McQueen	2010-01-01	Analyst	Green	Edit
Christopher	Lucas	2010-01-01	Analyst	Green	Edit

METRICS

[Expand view](#)

First Contentful Paint

1.0 s

Total Blocking Time

0 ms

Speed Index

2.2 s

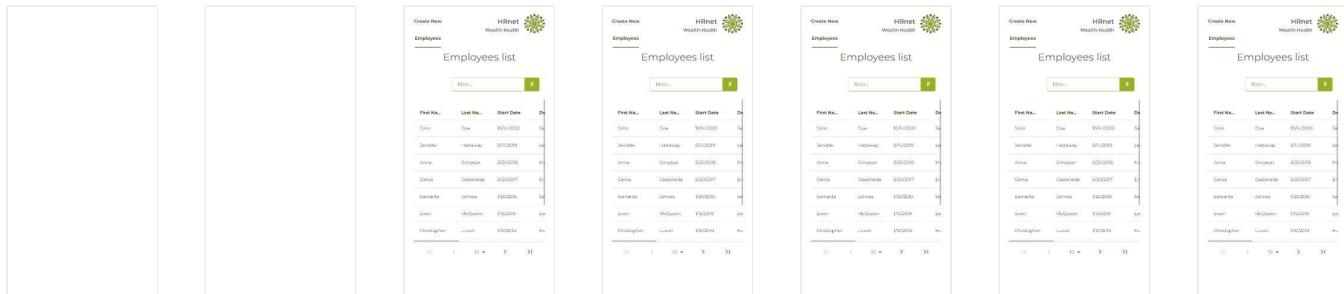
Largest Contentful Paint

1.5 s

Cumulative Layout Shift

0

[View Treemap](#)





Show audits relevant to: All FCP LCP TBT CLS

DIAGNOSTICS

▲ Reduce initial server response time — Root document took 740 ms

Keep the server response time for the main document short because all other requests depend on it. [Learn more about the Time to First Byte metric.](#) [FCP](#) [LCP](#)

URL	Time Spent
vercel.app First Party	740 ms
/employees (hrnet-react-tau.vercel.app)	740 ms

Minify JavaScript — Potential savings of 108 KiB

Minifying JavaScript files can reduce payload sizes and script parse time. [Learn how to minify JavaScript.](#) [FCP](#) [LCP](#)

Show 3rd-party resources (4)

URL	Transfer size	Potential savings
Unattributable	122.9 KiB	94.8 KiB
chrome-extension://cfhdojbkjhnlbpkdaibdccddilifddb/vendor/@eyeo/webext-sdk/content.js	101.7 KiB	83.8 KiB
chrome-extension://cfhdojbkjhnlbpkdaibdccddilifddb/polyfill.js	14.0 KiB	8.9 KiB
chrome-extension://cfhdojbkjhnlbpkdaibdccddilifddb/composer.preload.js	7.2 KiB	2.1 KiB
HeadingsMap Chrome Extension	38.3 KiB	13.2 KiB
chrome-extension://flbjommegcjopdmenkdiocclhjacmbi/content_scripts/utilsService.js	15.8 KiB	5.0 KiB
chrome-extension://flbjommegcjopdmenkdiocclhjacmbi/content_scripts/shortcutsServ	10.1 KiB	3.6 KiB

URL	Transfer size	Potential savings
ce.js chrome-extension://flbjommegcjopdmenkdiocclhjacmbi/content_scripts/treeService.js	6.8 KiB	2.6 KiB
chrome-extension://flbjommegcjopdmenkdiocclhjacmbi/content_scripts/algorithms/HTML5OutlineService.js	5.6 KiB	2.0 KiB

Reduce unused JavaScript — Potential savings of 72 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.](#) LCP

URL	Transfer size	Potential savings
vercel.app First Party	88.2 KiB	72.2 KiB
...chunks/329-564a8aa4414e8a47.js (hrnet-react-tau.vercel.app)	88.2 KiB	72.2 KiB

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

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 delivered to modern browsers, while retaining support for legacy browsers. [Learn how to use modern JavaScript](#) TBT

URL	Potential savings
vercel.app First Party	0.0 KiB
...chunks/329-564a8aa4414e8a47.js (hrnet-react-tau.vercel.app)	0.0 KiB
_next/static/chunks/...4a8aa4414e8a47.js:5	@babel/plugin-transform-classes

○ Avoids an excessive DOM size — 289 elements ^

A large DOM will increase memory usage, cause longer [style calculations](#) and produce costly [layout reflows](#). [Learn how to avoid an excessive DOM size.](#) TBT

Statistic	Element	Value
Total DOM Elements		289
Maximum DOM Depth	div.sc-cwHptR.hmdZnu	11
Maximum Child Elements	body.__className_cce811	13

Minimises main-thread work — 1.3 s

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

Category	Time Spent
Script Evaluation	422 ms
Script Parsing & Compilation	382 ms
Other	206 ms
Style & Layout	189 ms
Parse HTML & CSS	34 ms
Rendering	16 ms
Garbage Collection	4 ms

Avoid long main-thread tasks — 5 long tasks found

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

Show 3rd-party resources (1)

URL	Start Time	Duration
vercel.app <small>First Party</small>		307 ms

URL	Start Time	Duration
/employees (hrnet-react-tau.vercel.app)	795 ms	203 ms
/employees (hrnet-react-tau.vercel.app)	1,199 ms	53 ms
/employees (hrnet-react-tau.vercel.app)	1,098 ms	51 ms
Unattributable		54 ms
Unattributable	998 ms	54 ms
HeadingsMap Chrome Extension		50 ms
chrome-extension://flbjommegcjonpdmenkdiocclhjacmbi/content_scripts/storageService.js	1,149 ms	50 ms
s		

JavaScript execution time — 0.5 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

URL	Total CPU Time	Script Evaluation	Script Parse
vercel.app First Party	781 ms	261 ms	232 ms
/employees (hrnet-react-tau.vercel.app)	582 ms	72 ms	229 ms
...chunks/864-ac640d799c8d11a6.js (hrnet-react-tau.vercel.app)	199 ms	189 ms	4 ms
Unattributable	150 ms	9 ms	0 ms
Unattributable	150 ms	9 ms	0 ms

User Timing marks and measures — 1 user timing

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.](#)

Name	Type	Start Time	Duration
__v3	Mark	0.00 ms	

○ Avoids enormous network payloads — Total size was 294 KiB ^

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

URL	Transfer size
vercel.app First Party	275.3 KiB
...chunks/329-564a8aa4414e8a47.js (hrnet-react-tau.vercel.app)	88.2 KiB
...chunks/fd9d1056-13492bb6c882dac1.js (hrnet-react-tau.vercel.app)	52.0 KiB
...media/e1c529c04de64b40-s.p.woff2 (hrnet-react-tau.vercel.app)	32.6 KiB
...chunks/763-dc77ad14979d6d59.js (hrnet-react-tau.vercel.app)	29.0 KiB
...chunks/864-ac640d799c8d11a6.js (hrnet-react-tau.vercel.app)	28.3 KiB
...chunks/280-527...a97c.js (hrnet-react-tau.vercel.app)	14.1 KiB
...chunks/344-feb06a297....js (hrnet-react-tau.vercel.app)	12.1 KiB
/_next/image?url=... (hrnet-react-tau.vercel.app)	10.9 KiB
...css/d24e364fefafa8b3fb.css (hrnet-react-tau.vercel.app)	4.1 KiB
...app/page-f8721d97a2ace07d.js (hrnet-react-tau.vercel.app)	3.9 KiB

○ Avoid chaining critical requests — 2 chains found ^

The critical request chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources or deferring the download of unnecessary resources to improve page load. [Learn how to avoid chaining critical requests.](#) FCP LCP

Maximum critical path latency: **831.322 ms**

Initial Navigation

/employees (hrnet-react-tau.vercel.app)

...css/6f7f31264299a6de.css (hrnet-react-tau.vercel.app) - **42.628 ms, 2.01 KiB**

...css/636b83c00cf5ecdf.css (hrnet-react-tau.vercel.app) - 77.723 ms, 1.02 KiB

Minimise third-party usage — Third-party code blocked the main thread for 0 ms

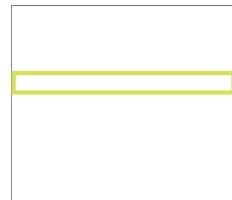
Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading. [Learn how to minimise third-party impact.](#) TBT

Third-party	Transfer size	Main-thread blocking time
HeadingsMap <small>Chrome Extension</small>	16 KiB	0 ms
chrome-extension://flbjommegcjonpdmenkdiocclhjacmbi/_locales/en/messages.json	16 KiB	0 ms

Largest contentful paint element — 1,540 ms

This is the largest contentful element painted within the viewport. [Learn more about the Largest Contentful Paint element \(LCP\)](#)

Element



h2

Phase	% of LCP	Timing
TTFB	48%	740 ms
Load delay	0%	0 ms
Load time	0%	0 ms
Render delay	52%	800 ms

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

PASSED AUDITS (26)

Hide

Eliminate render-blocking resources

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn how to eliminate render-blocking resources.](#) [FCP](#) [LCP](#)

Properly size images

Serve images that are appropriately-sized to save mobile data and improve load time. [Learn how to size images.](#)

Defer off-screen images

Consider lazy loading offscreen and hidden images after all critical resources have finished loading to lower Time to Interactive. [Learn how to defer offscreen images.](#)

Minify CSS

Minifying CSS files can reduce network payload sizes. [Learn how to minify CSS.](#) [FCP](#) [LCP](#)

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](#)

Efficiently encode images

Optimised images load faster and consume less mobile data. [Learn how to efficiently encode images.](#)

Serve images in next-gen formats

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.](#)

Enable text compression

Text-based resources should be served with compression (gzip, deflate or brotli) to minimise total network bytes. [Learn more about text compression.](#) [FCP](#) [LCP](#)

Pre-connect to required origins

Consider adding preconnect or dns-prefetch resource hints to establish early connections to important third-party origins. [Learn how to preconnect to required origins.](#) [FCP](#) [LCP](#)

Avoid multiple page redirects

Redirects introduce additional delays before the page can be loaded. [Learn how to avoid page redirects.](#) [FCP](#) [LCP](#)

Pre-load key requests

Consider using `<link rel=preload>` to prioritise fetching resources that are currently requested later in page load. [Learn how to preload key requests.](#) FCP LCP

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](#) LCP

Remove duplicate modules in JavaScript bundles

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

TBT

Preload largest contentful paint image

If the LCP element is dynamically added to the page, you should preload the image in order to improve LCP. [Learn more about preloading LCP elements.](#) LCP

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.](#)

All text remains visible during webfont loads

Leverage the `font-display` CSS feature to ensure that text is user-visible while webfonts are loading. [Learn more about font-display.](#) FCP LCP

Lazy load third-party resources with facades

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn how to defer third-parties with a facade.](#) TBT

Largest contentful paint image was not lazily loaded

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the Largest Contentful Paint. [Learn more about optimal lazy loading.](#) LCP

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 non-composited animations ^

Animations that are not composited can be poor, slow and increase CLS. [Learn how to avoid non-composited animations](#) CLS

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

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimises your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more about using the viewport meta tag](#). TBT

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](#)



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.](#)

○ 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.](#)

○ 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.](#)

○ 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.](#)

○ 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.](#)

○ 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.](#)

○ 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.](#)

○ 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.](#)

○ 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.](#)

○ Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn how to add roles to custom controls.](#)

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

PASSED AUDITS (24)

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](#).

[aria-*] attributes have valid values

^

Assistive technologies, such as 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, such as 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 alternative 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-hidden="true"] elements do not contain focusable descendants

Focusable descendants 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.](#)

[role] values are valid

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more about valid ARIA roles.](#)

Background and foreground colours have a sufficient contrast ratio

Low-contrast text is difficult or impossible for many users to read. [Learn how to provide sufficient colour 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.](#)

[id] attributes on active, focusable elements are unique

All focusable elements must have a unique id to ensure that they're visible to assistive technologies. [Learn how to fix duplicate ids.](#)

<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.](#)

Form elements have associated labels

Labels ensure that form controls are announced properly by assistive technologies, such as screen readers. [Learn more about form element labels.](#)

Links have a discernible name

Link text (and alternative 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.](#)

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.](#)

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.](#)

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.](#)

Values assigned to role="" are valid ARIA roles.

ARIA roles enable assistive technologies to know the role of each element on the web page. If the role values are misspelled, not existing ARIA role values or abstract roles, then the purpose of the element will not be communicated to users of assistive technologies. [Learn more about ARIA roles.](#)

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.](#)

NOT APPLICABLE (37)

[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.](#)

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.](#)

○ Elements with `role="dialog"` or `role="alertdialog"` have accessible names. ^

ARIA dialogue elements without accessible names may prevent screen reader users from discerning the purpose of these elements. [Learn how to make ARIA dialog elements more accessible.](#)

○ 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.](#)

○ 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.](#)

○ 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.](#)

○ Elements with the `role=text` attribute do not have focusable descendants. ^

Adding `role=text` around a text node split by markup enables VoiceOver to treat it as one phrase, but the element's focusable descendants will not be announced. [Learn more about the role=text attribute.](#)

○ 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.](#)

○ 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.](#)

○ 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 labelling treeitem elements.](#)

○ 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.](#)

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

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn how to structure definition lists correctly.](#)

○ 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.](#)

○ 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.](#)

○ 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.](#)

○ `<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.](#)

○ `<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.](#)

○ Input buttons have discernible text. ^

Adding discernable and accessible text to input buttons may help screen reader users to understand the purpose of the input button. [Learn more about input buttons.](#)

○ `<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.](#)

○ Elements with visible text labels have matching accessible names. ^

Visible text labels that do not match the accessible name can result in a confusing experience for screen reader users. [Learn more about accessible names.](#)

- Links are distinguishable without relying on colour.
 - 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](#).
- Lists contain only `` 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](#).
- List items (``) are contained within ``, `` or `<menu>` parent elements
 - Screen readers require list items (``) to be contained within a parent ``, `` or `<menu>` to be announced properly. [Learn more about proper list structure](#).
- 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](#).
- `<object>` elements have alternative text
 - Screen readers cannot translate non-text content. Adding alternative text to `<object>` elements helps screen readers convey meaning to users. [Learn more about alt text for object elements](#).
- Skip links are focusable.
 - Including a skip link can help users skip to the main content to save time. [Learn more about skip links](#).
- 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](#).
- Tables use `<caption>` instead of cells with the `[colspan]` attribute to indicate a caption.
 - Screen readers have features to make navigating tables easier. Ensuring that tables use the actual caption element instead of cells with the `[colspan]` attribute may improve the experience for screen reader users. [Learn more about captions](#).
- `<td>` elements in a large `<table>` have one or more table headers.
 - Screen readers have features to make navigating tables easier. Ensuring that `<td>` elements in a large table (three or more cells in width and height) have an associated table header may improve the experience for screen reader users. [Learn more about table headers](#).

- 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 that `<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.](#)

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

^

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

- `[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.](#)

- `<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.](#)

- All heading elements contain content.

^

A heading with no content or inaccessible text prevents screen reader users from accessing information on the page's structure. [Learn more about headings.](#)

- 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.](#)

- Document has a main landmark.

^

One main landmark helps screen reader users navigate a web page. [Learn more about landmarks.](#)

- 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.](#)



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](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

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](#).

Allows users to paste into input fields

Preventing input pasting is 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](#).

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 maximise 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 1,024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more about declaring the character encoding.](#)

Avoids `unload` event listeners ^

The unload event does not fire reliably and listening for it can prevent browser optimisations 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 to 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.](#)

URL	Map URL
chrome-extension://cfhdojbkjhnlbpkdaibdccddilifddb/vendor/@eyeo/webe xt-sdk/content.js	

chrome-extension://cfhdojbkjhnlbpkdaibdccddilifddb/vendor/@eyeo/webe
xt-sdk/content.js

NOT APPLICABLE (2)

Hide

- Fonts with `font-display: optional` are preloaded ^

Preload optional fonts so that first-time visitors may use them. [Learn more about preloading fonts](#)

- Detected JavaScript libraries ^

All front-end JavaScript libraries detected on the page. [Learn more about this JavaScript library detection diagnostic audit](#).



SEO

These checks ensure that your page is following basic search engine optimisation advice. There are many additional factors that 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](#) and the [Structured Data Linter](#) to validate structured data. [Learn more about structured data](#).

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

PASSED AUDITS (12)

[Hide](#)

- Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimises your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more about using the viewport meta tag](#). TBT

- 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 summarise 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 that more pages of the site can be discovered. Learn how to make links crawlable									
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.									
Image elements have [alt] attributes	^								
Informative elements should aim for short, descriptive alternative 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 uses legible font sizes — 100% legible text	^								
Font sizes less than 12px are too small to be legible and require mobile visitors to 'pinch to zoom' in order to read. Strive to have >60% of page text ≥12px. Learn more about legible font sizes.									
<table border="1"><thead><tr><th>Source</th><th>Selector</th><th>% of page text</th><th>Font size</th></tr></thead><tbody><tr><td>Legible text</td><td></td><td>100.00%</td><td>≥ 12px</td></tr></tbody></table>		Source	Selector	% of page text	Font size	Legible text		100.00%	≥ 12px
Source	Selector	% of page text	Font size						
Legible text		100.00%	≥ 12px						
Document avoids plugins	^								
Search engines can't index plug-in content and many devices restrict plug-ins or don't support them. Learn more about avoiding plugins.									
Tap targets are sized appropriately — 100% appropriately sized tap targets	^								

Interactive elements like buttons and links should be large enough (48 x 48 px) or have enough space around them to be easy enough to tap without overlapping onto other elements. [Learn more about tap targets.](#)

NOT APPLICABLE (2)

Hide

 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](#).

 Document has a valid `rel=canonical` ^

Canonical links suggest which URL to show in search results. [Learn more about canonical links](#).

Captured at 29 Dec 2023,
10:42 CET
Initial page load

Emulated Moto G Power with
Lighthouse 11.2.0
Slow 4G throttling

Single page load
Using Chromium 120.0.0.0 with
devtools

Generated by **Lighthouse** 11.2.0 | [File an issue](#)