



Performance

Accessibility

Best Practices

SEO



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

[Collapse view](#)

- First Contentful Paint

1.0 s

First Contentful Paint marks the time at which the first text or image is painted. [Learn more about the First Contentful Paint metric.](#)

- ! Largest Contentful Paint

Error!

NO_LCP

! Total Blocking Time

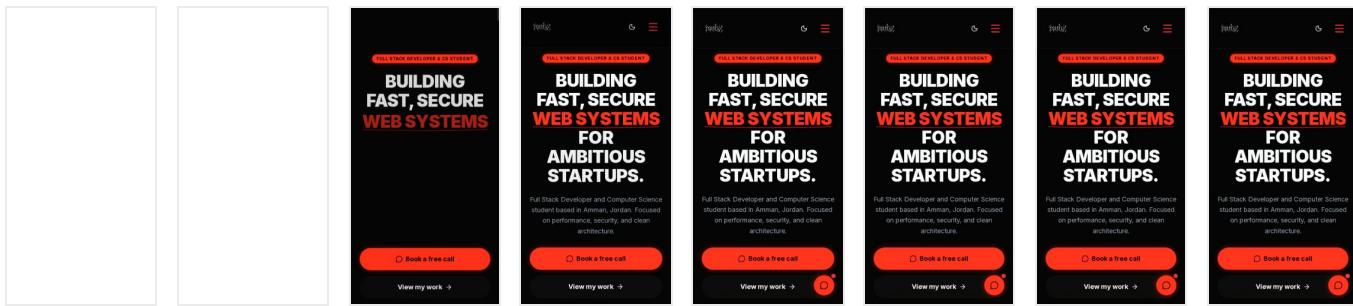
! 97 100 100

Movement of visible elements within the viewport. [Learn more about the Cumulative Layout Shift metric.](#)

● Speed Index

2.9 s

Speed Index shows how quickly the contents of a page are visibly populated. [Learn more about the Speed Index metric.](#)



Show audits relevant to: [All](#) [LCP](#) [TBT](#)

INSIGHTS

▲ 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

Top function call

Total reflow time

...chunks/506b0b0b5578a47f.js:2:8527 (mubx.dev)

60 ms

Source

Total reflow time

[unattributed]

58 ms

...chunks/3ab36bfd939cb062.js:1:4074 (mubx.dev)

58 ms

...chunks/506b0b0b5578a47f.js:1:85626 (mubx.dev)

2 ms

▲ Improve image delivery — Est savings of 85 KiB

Reducing the download time of images can improve the perceived load time of the page and LCP.

	!	97	100	100		
URL					Size	Est Savings
mubx.dev 1st Party					87.2 KiB	85.1 KiB
	MUBX - High Perfor manc e Web					
Systems & E-commerce						
		/_next/image? url=%2Ficon.png&w=96&q=75 (mubx.dev)		87.2 KiB	85.1 KiB	
	Using a modern image format (WebP, AVIF) or increasing the image compression could improve this image's download size.				61.1 KiB	
	This image file is larger than it needs to be (400x400) for its displayed dimensions (112x112). Use responsive images to reduce the image download size.				80.4 KiB	

▲ Legacy JavaScript — Est savings of 14 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](#) Unscored

URL**Wasted bytes**
! 97 100 100

...chunks/4b2c135c89b53e2a.js:1:6069 (mubx.dev)	Array.prototype.at
...chunks/4b2c135c89b53e2a.js:1:5457 (mubx.dev)	Array.prototype.flat
...chunks/4b2c135c89b53e2a.js:1:5570 (mubx.dev)	Array.prototype.flatMap
...chunks/4b2c135c89b53e2a.js:1:5946 (mubx.dev)	Object.fromEntries
...chunks/4b2c135c89b53e2a.js:1:6204 (mubx.dev)	Object.hasOwnProperty
...chunks/4b2c135c89b53e2a.js:1:5199 (mubx.dev)	String.prototype.trimEnd
...chunks/4b2c135c89b53e2a.js:1:5114 (mubx.dev)	String.prototype.trimStart

▲ 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. LCP Unscored

URL**Transfer Size****Duration**

mubx.dev	1st Party	15.1 KiB	0 ms
...chunks/ac180c698...a.css	(mubx.dev)	15.1 KiB	

○ 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

1,320

DOM depth

div.flex > a.text-neon/80 > svg.lucide > path
`<path d="M10 13a5 5 0 0 7.54.5413-3a5 5 0 0 0-7.07-7.071-1.72 1.71">`

13

Most children

REACT NEXT.JS TYPESCRIPT NODE.JS POSTGRESQL PRISMA TAILWIND VERCCEL FIGMA HTML5
`<div class="flex flex-wrap justify-center gap-4 md:gap-8 max-w-6xl mx-auto">`

27

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

DIAGNOSTICS

! 97 100 100

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

! Minify JavaScript — Error!

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

! Reduce unused CSS — Error!

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](#). [Unscored]

! Reduce unused JavaScript — Error!

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn how to reduce unused JavaScript](#). [Unscored]

! Avoid long main-thread tasks — Error!

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay.

[Learn how to avoid long main-thread tasks](#) [Unscored]

! Minimize main-thread work — 2.4 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
Script Evaluation	897 ms
Other	653 ms
Style & Layout	530 ms
Script Parsing & Compilation	171 ms
Rendering	102 ms
Garbage Collection	38 ms

Category

Time Spent

! 97 100 100

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

PASSED AUDITS (16) Hide

● Use efficient cache lifetimes ^

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

● 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. LCP FCP Unscored

Server responds quickly (observed 82 ms)

Applies text compression

Avoids redirects

● Duplicated JavaScript ^

Remove large, [duplicate JavaScript modules](#) from bundles to reduce unnecessary bytes consumed by network activity. 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

○ INP breakdown

! 97 100 100

○ LCP breakdown

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

○ LCP request discovery

[Optimize LCP](#) by making the LCP image discoverable from the HTML immediately, and avoiding lazy-loading Unscored

● 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: **0 ms**

Initial Navigation

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

● 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

● Optimize viewport for mobile

! 97 100 100

head > meta

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

● Avoids enormous network payloads — Total size was 791 KiB

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

URL	Transfer Size
mubx.dev 1st Party	616.9 KiB
...chunks/691626e331....js (mubx.dev)	231.5 KiB
/_next/image?url=%2Ficon.png&w=96&q=75 (mubx.dev)	88.1 KiB
...chunks/4b2c135c89b53e2a.js (mubx.dev)	71.2 KiB
...media/83afe278b6a6bb3c-s.p.3a6ba036.woff2 (mubx.dev)	48.8 KiB
...chunks/506b0b0b5578a47f.js (mubx.dev)	41.2 KiB
...media/70bc3e132a0a741e-s.p.15008fb.woff2 (mubx.dev)	41.0 KiB
...chunks/83918d10e01e881d.js (mubx.dev)	32.0 KiB
...media/0c89a48fa5027cee-s.p.4564287c.woff2 (mubx.dev)	23.3 KiB
...chunks/473a6939ad0d0d0c.js (mubx.dev)	20.7 KiB
https://mubx.dev	19.2 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 — 1.0 s

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller

! 97 100 100

URL	Time	Script Evaluation	Script Parse
mubx.dev 1st Party	2,155 ms	844 ms	116 ms
https://mubx.dev	789 ms	5 ms	1 ms
...chunks/4b2c135c89b53e2a.js (mubx.dev)	715 ms	464 ms	22 ms
...chunks/691626e331....js (mubx.dev)	358 ms	263 ms	76 ms
...chunks/506b0b0b5578a47f.js (mubx.dev)	209 ms	29 ms	16 ms
...chunks/turbopack-0a6693c471ba5cdc.js (mubx.dev)	85 ms	83 ms	2 ms
Unattributable	121 ms	5 ms	0 ms
Unattributable	121 ms	5 ms	0 ms

○ 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](#)



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Automatic detection can only detect a subset of issues and does

! 97 100 100

CONTRAST

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

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

Failing Elements

01

```
<div class="absolute top-6 right-6 text-4xl font-black text-muted/40 group-hover:text-...">  
 01 The Glorious Page  Link Bio A stunning, high-conversion landing point for ...  
<div class="group relative p-8 rounded-3xl bg-glass transition-all duration-300 flex f...">
```

02

```
<div class="absolute top-6 right-6 text-4xl font-black text-muted/40 group-hover:text-...">  
 02 Vynex Media  Digital Agency Professional page for client acquisition with ...  
<div class="group relative p-8 rounded-3xl bg-glass transition-all duration-300 flex f...">
```

03

```
<div class="absolute top-6 right-6 text-4xl font-black text-muted/40 group-hover:text-...">  
 03 HTU Martial Arts  Education System Centralized management reducing manual ...  
<div class="group relative p-8 rounded-3xl bg-glass transition-all duration-300 flex f...">
```

04

```
<div class="absolute top-6 right-6 text-4xl font-black text-muted/40 group-hover:text-...">  
 04 BloB.JO  E-commerce Digital order flow replacing manual communication. Bri...  
<div class="group relative p-8 rounded-3xl bg-glass transition-all duration-300 flex f...">
```

These are opportunities to improve the legibility of your content.

BEST PRACTICES

- 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]

Failing Elements

! 97 100 100

`muted/10 ..." href="https://www.theglorious.page/">`

Visit Live Website

``

Visit Live Website

``

Visit Live Website

``

These items highlight common accessibility best practices.

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

! 97 100 100

- 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 (26)

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](#).

! 97 100 100

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

[Learn more about roles and required attributes.](#)

● [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.](#)

● 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.](#)

! 97 100 100

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](#).

● 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](#).

● 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 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](#).

● 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](#).

● 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

! 97 100 100

[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.](#)

● 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.](#)

● 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.](#)

! 97 100 100

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 (32) 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

! 97 100 100

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 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.](#) Unscored

○ `[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.](#) Unscored

○ 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.](#) 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

! 97 100 100

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 `<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.](#) 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.

! 97 100 100

- 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

- 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

- 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

- 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! 97 100 100

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

- 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. ^

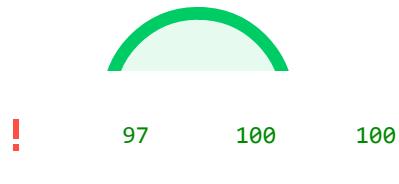
A heading with no content or inaccessible text prevent screen reader users from accessing information on the page's structure. [Learn more about headings.](#) Unscored

- 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



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
Host allowlists can frequently be bypassed. Consider using CSP nonces or hashes instead, along with 'strict-dynamic' if necessary.	script-src	High
'unsafe-inline' allows the execution of unsafe in-page scripts and event handlers. Consider using CSP nonces or hashes to allow scripts individually.	script-src	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

GENERAL

▲ Missing source maps for large first-party JavaScript ^

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

URL

Map URL

! 97 100 100

...chunks/09f020e551...js (mubx.dev)

Large JavaScript file is missing a source map

PASSED AUDITS (12)

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](#).

● Avoids requesting the notification permission on page load

Users are mistrustful of or confused by sites that request to send notifications without context.

! 97 100 100

● 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.

NOT APPLICABLE (5)

Hide

! 97 100 100

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

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

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

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

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](#)

! 97 100 100

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](#).

! 97 100 100

143.0.7499.192 with lr

Generated by **Lighthouse** 13.0.1 | [File an issue](#)