

Element

Phase % of LCP Timing

TTFB	8%	190 ms
Load Delay	8%	180 ms
Load Time	17%	380 ms
Render Delay	67%	1,520 ms

Cloudinary [Content]

vetvahocode [Content]

Minify CSS – Potential savings of 2 kB

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

URL	Transfer Size	Potential Savings
!important { font-size: 0.8em; } —	24.1 kB	2.4 kB

Serve static assets with an efficient cache policy – 1 resource found

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

URL	Cache TTL	Transfer Size
...v170.../card4-seal-platinum-2024_week1.png (via cloudinary.com)	35d	7 kB

Ensure text remains visible during webfont load

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading. [Learn how to use font-display](#) [HTML](#) [CSS](#)

URL	Potential Savings
Loom - Screen Recorder & Screen Capture (Chrome Extension)	0 ms
chrome-extension://laebabbbkhhedelbltlllej14ddc/fonts/circularN000000-wfT2	0 ms

Defer offscreen images – Potential savings of 8 kB

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](#)

URL	Resource Size	Potential Savings
vetvahocode [Content]	7.8 kB	7.8 kB

about:blank

This report provides a detailed analysis of the resources loaded by the page at [about:blank](#). It includes sections for URL, Resource Size, Potential Savings, and a breakdown of third-party resources.

Resource Analysis

The page has loaded 11 resources:

- URL:** [about:blank](#)
- Resource Size:** 7.8 KB
- Potential Savings:** 7.8 KB

Third-Party Resources: 1 resource (100% of total)

URL	Transfer Size	Potential Savings
vethocode.io	7.8 KB	7.8 KB

Performance & Optimization

The page has a **JavaExecution time** of 0.6 ms. 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 payload](#)

Avoid serving legacy JavaScript to modern browsers — **Potential savings of 19 KB**

Polyfill and transform viable 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. Modern browsers support ES6+ and can handle most modern features without polyfills or legacy resources. [Learn how to use modern JavaScript](#)

Show 3rd-party resources (1)

Unattributable 10.7 KB

chrome-extension://nfapjheqjapjhahmgl1apjkaa/v/content_24 10.7 KB

content.js2 [!/polyfill/poly.js_0](#)
[!/polyfill/poly.js_0](#)
[!/polyfill/poly.js_1](#)
[!/polyfill/poly.js_2](#)
[global/plugins-transformer-classes](#)

content.js2 [Loon - Screen Recorder & Screen Capture \[Chrome Extension\]](#) 0.4 KB

chrome-extension://nfapjheqjapjhahmgl1apjkaa/v/content_24 0.4 KB

companionBubble.js4 [global/plugins-transformer-classes](#)
[!object.keys](#)

companionBubble.js4 [global/plugins-transformer-classes](#)

vethocode.io 0.0 KB

...chunks/main-12f1294293d9e.js [vethocode.io](#) 0.0 KB

main-12f1294293d9e.js1 [global/plugins-transformer-classes](#)

Reduce unused CSS — **Potential savings of 61 KB**

Reduce unused rules from styleheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn how to reduce unused CSS](#)

12x1: Consider setting up PurgeCSS in React.js configuration to remove unused rules from styleheets. [Learn more](#)

URL	Transfer Size	Potential Savings
Unattributable	45.4 KB	42.1 KB
vethocode.io	24.1 KB	20.8 KB
#root-shadow-capsize { font-size: 14px; }	21.3 KB	18.5 KB
vethocode.io	23.2 KB	18.5 KB
...css/13edead23dc.css vethocode.io	23.2 KB	18.5 KB

Reduce unused JavaScript — **Potential savings of 2,001 KB**

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

12x1: Use Webpack Bundle Analyzer to detect unused JavaScript code. [Learn more](#)

Show 3rd-party resources (2)

URL	Transfer Size	Potential Savings
Unattributable	2,916.6 KB	1,861.3 KB
chrome-extension://nfapjheqjapjhahmgl1apjkaa/v/content_24	2,916.6 KB	1,861.3 KB
Loon - Screen Recorder & Screen Capture [Chrome Extension]	560.0 KB	573.2 KB
chrome-extension://nfapjheqjapjhahmgl1apjkaa/v/content_24	560.0 KB	573.2 KB
Google Tag Manager [Tag Manager]	88.6 KB	35.9 KB
...js/PIC-G-WXV2020.js www.googletagmanager.com	88.6 KB	35.9 KB

Avoid enormous network payloads — **Total size was 3,152 KB**

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

Show 3rd-party resources (2)

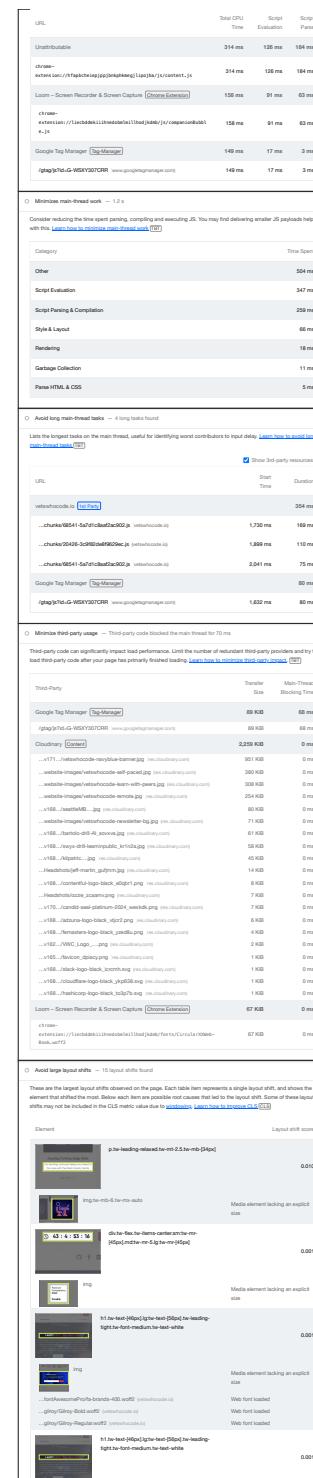
URL	Transfer Size
Cloudinary [Content]	2,043.2 KB
...PIC-G-WXV2020.js www.googletagmanager.com	661.5 KB
...website-image/vethocode-self-packed.js www.cloudinary.com	370.8 KB
...website-image/vethocode-learn-with-pears.js www.cloudinary.com	307.5 KB
...website-image/vethocode-remove.js www.cloudinary.com	254.3 KB
...168...-website.js www.cloudinary.com	70.8 KB
...website-image/vethocode-newsletter-lq.js www.cloudinary.com	71.4 KB
vethocode.io	375.2 KB
...fontAwesomeFont-regular-400-wf2 vethocode.io	100.0 KB
...fontAwesomeFont-solid-900-wf2 vethocode.io	134.7 KB
...fontAwesomeFont-brands-400-wf2 vethocode.io	74.4 KB
Google Tag Manager [Tag Manager]	88.7 KB
...js/PIC-G-WXV2020.js www.googletagmanager.com	88.7 KB

JavaExecution time — 0.6 ms

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 payload](#)

Show 3rd-party resources (2)

URL	Total CPU Time	Script Evaluation	Script Parse
vethocode.io	427 ms	85 ms	3 ms
...chunks/2f4d0-3cfbd-de409c2ec.js vethocode.io	205 ms	0 ms	0 ms
...chunks/framework-daf4e5e552fd1c0.js vethocode.io	91 ms	73 ms	2 ms
https://vethocode.io	64 ms	1 ms	0 ms
...chunks/18541-5a7d1bba2ac302.js vethocode.io	78 ms	12 ms	1 ms



The screenshot shows the Lighthouse performance audit report for the URL 'about:blank'. The report includes sections for 'Layout shift score' (with a score of 0.00), 'User Timing marks and measures' (including 'Nest.js-before-hydration' and 'Nest.js-hydration' marks), 'Animations which are not composited' (with a link to 'Learn more about this'), 'Initial server response time was short' (with a link to 'Learn more about this'), and 'Avoid an excessive DOM size' (with a table showing static elements: Total DOM Elements = 471, Maximum DOM Depth = 13, Maximum Child Elements = 4). The report also includes a note at the bottom: 'More information about the performance of your application. These numbers don't directly affect the Performance score.'

Layout shift score

Media element lacking an explicit size

Web font loaded

User Timing marks and measures

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

Consider using Nest.js Analytics to measure your app's real-world performance. [Learn more](#)

Name	Type	Start Time	Duration
Nest.js-before-hydration	Measure	0.00 ms	544.38 ms
Nest.js-hydration	Measure	544.38 ms	65.69 ms
beforeRender	Mark	544.38 ms	
afterHydrate	Mark	610.07 ms	

Avoid non-composited animations

Animations which are not composited can be jerky and increase CPU usage. [Learn more about this](#)

Element Name

div.swiper > div.swiper-pagination > span.swiper-pagination-bullet > :before

width

background-color

height

div.swiper > div.swiper-pagination > span.swiper-pagination-bullet > :before

width

background-color

height

Initial server response time was short

Keep the server response time for the main document short because all other requests depend on it. [Learn more about this](#)

URL Time Spent

https://vercel-hocode.id 40 ms

Avoid an excessive DOM size

A large DOM will increase memory usage, cause longer [page load times](#), and produce costly [layout shifts](#). [Learn more](#)

Static	Element	Value
Total DOM Elements		471
Maximum DOM Depth	img[alt="full-width-full-height object-cover"]	13
Maximum Child Elements	ul	8

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

PASSED AUDITS (110)

- 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/css. [Learn how to remove render-blocking resources](#) [PDF](#) [CSV](#)

(1) [X] Use the next/last component to defer loading of non-critical third-party script. [Learn more](#)
- Minify JavaScript

Minifying JavaScript files can reduce payload size and script run time. [Learn how to minify JavaScript](#) [PDF](#) [CSV](#)
- Efficiently encode images

Optimized images load faster and consume less cellular data. [Learn how to optimize images](#) [PDF](#) [CSV](#)

(1) [X] Use the next/last image component instead of to adjust image quality. [Learn more](#)
- Serve images in web-safe formats

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn how to serve images in web-safe formats](#) [PDF](#) [CSV](#)

(1) [X] Use the next/last image component instead of to automatically optimize image format. [Learn more](#)
- Enable text compression

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn about text compression](#) [PDF](#) [CSV](#)

(1) [X] Enable compression on your Next.js server. [Learn more](#)
- Precache to required origins

Consider adding precache or on-demand fetch resource hints to establish early connections to important third-party origins. [Learn how to precache or on-demand fetch](#) [PDF](#) [CSV](#)
- Avoid multiple page reloads

Redirections introduce additional delays before the page can be loaded. [Learn how to avoid page reloads](#) [PDF](#) [CSV](#)
- Preload by requests

Consider using <link rel="preload"> to prioritize fetching resources that are currently requested later in page load. [Learn how to preload by requests](#) [PDF](#) [CSV](#)
- Use HTTP/2

HTTP/2 offers many benefits over HTTP/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 MP4/GIF/WebM videos for animations and PNG/WAV for static images instead of GIF to save network bytes. [Learn more about efficient video formats](#) [PDF](#)
- Remove duplicate modules in JavaScript bundles

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. [Learn more](#)
- 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 how to preload LCP elements](#) [PDF](#)

(1) [X] Use the next/last image component and set "priority" to true to preload LCP image. [Learn more](#)
- Avoid chaining critical requests

The Critical Request Chain below shows what resources are loaded with a high priority. Consider reducing the length of the chain, reducing the download cost of elements, or deferring the download of unnecessary resources to improve page load. [Learn how to reduce critical request chains](#) [PDF](#) [CSV](#)

Maximum critical path latency: 44.368 ms

Initial Navigation
https://whatwebcoders.com → 44.368 ms, 13.00 kB
- Lazy load third-party resources with facets

Some third-party embeds can be lazy loaded. Consider replacing them with a facets until they are required. [Learn how to use lazy-loaded facets](#) [PDF](#)
- Largest Contentful Paint image was not fully loaded

Above-the-fold images that are fully loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more about criticality for rendering](#) [PDF](#)

Element
- Use passive listeners to improve scrolling performance

Consider marking your touch and wheel event listeners as passive to improve your page's scroll performance. [Learn how to use passive scroll listeners](#) [PDF](#)
- Avoid 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\(\)](#)
- Hide a remote image with and width=0 or height=0

A remote image with alt="" has only optimizes user app for mobile screen sizes, but also prevents a DOM mutation. [Learn how to hide remote images](#) [PDF](#)
- 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 (bCache) can speed up these return navigations. [Learn how to avoid the bCache](#)

96

Accessibility

These checks highlight potential issues related to the accessibility of your page. [Learn how to fix accessibility issues](#) [PDF](#) [CSV](#). Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app. See [Known Issues](#) if any are encountered.

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](#)
- Falling Elements



ADDITIONAL ITEMS TO MANUALLY CHECK (0)	
<input type="checkbox"/> Interactive controls are keyboard focusable	Custom interactive controls are keyboard-focusable and display a focus indicator.
<input type="checkbox"/> 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.
<input type="checkbox"/> The page has a logical tab order	Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen.
<input type="checkbox"/> Visual order on the page follows DOM order	DOM order matches the visual order, improving navigation for assistive technology.
<input type="checkbox"/> 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.
<input type="checkbox"/> 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.
<input type="checkbox"/> HTML5 landmark elements are used to improve navigation	Landmark elements (region, nav, etc.) are used to improve the keyboard navigation of the page for assistive technology.
<input type="checkbox"/> Offscreen content is hidden from assistive technology	Offscreen content is hidden with display: none or aria-hidden=true.
<input type="checkbox"/> Custom controls have associated labels	Custom interactive controls have associated labels, provided by aria-label or aria-labelledby.
<input type="checkbox"/> Custom controls have ARIA roles	Custom interactive controls have appropriate ARIA roles.

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

PASSED AUDITS (12)

<input checked="" type="checkbox"/> <code>[aria-*]</code> attributes match their role
Each ARIA role supports a specific subset of <code>aria-*</code> attributes. Mismatching these invalidates the <code>aria-*</code> attributes.
Learn more about ARIA roles
<input checked="" type="checkbox"/> <code>[aria-hidden="true"]</code> is not present on the document body
Assistive technologies, like screen readers, work inconsistently when <code>aria-hidden="true"</code> is set on the document body.
Learn more about aria-hidden=true on the document body
<input checked="" type="checkbox"/> <code>[aria-*]</code> have required <code>aria-*</code> attributes
Some ARIA roles have required attributes that describe the state of the element to screen readers.
Learn more about aria-* attributes
<input checked="" type="checkbox"/> <code>[aria-*]</code> attributes have valid values
Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values.
Learn more about invalid ARIA values for ARIA attributes
<input checked="" type="checkbox"/> <code>[aria-*]</code> 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
<input checked="" type="checkbox"/> 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 more about accessible names for buttons
<input checked="" type="checkbox"/> Image elements have <code>alt</code> attributes
Informational elements should not be short, descriptive alternative text. decorative elements can be ignored with an empty alt attribute.
Learn more about alt attributes
<input checked="" type="checkbox"/> <code>[aria-mutable="true"]</code> is not used in the <code><meta name="viewport"></code> element and the <code>aria-mutable</code> attribute is not less than 1
Shadow scrolling 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
<input checked="" type="checkbox"/> Buttons, links, and <code>executives</code> 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 more about accessible names for buttons, links, and executives
<input checked="" type="checkbox"/> <code>[role="value"]</code> value is valid
ARIA roles must have valid values in order to perform their intended accessibility functions.
Learn more about valid ARIA roles
<input checked="" type="checkbox"/> Document has a <code><title></code> element
The title gives screen reader users an overview of the page, and search engines use it to help to determine if a page is relevant to their search.
Learn more about document titles
<input checked="" type="checkbox"/> <code><html></code> element has a <code>(lang)</code> 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
<input checked="" type="checkbox"/> <code><html></code> element has a valid value for its <code>lang</code> attribute
Specifying a valid BCP 47 language helps screen readers announce text properly.
Learn how to use the lang attribute
<input checked="" type="checkbox"/> Form elements have associated labels
Labels ensure that form controls are announced properly by assistive technologies, like screen readers.
Learn more about form controls and labels
<input checked="" type="checkbox"/> Links are distinguishable without relying on color
Low-contrast text is difficult or impossible for many users to read. Link text that is distinguishable improves the experience for users with low vision.
Learn how to make many colors distinguishable

about:blank

This page contains 11 items:

- **Links have a descriptive name**
 - Links contain only `<a>` elements and script supporting elements (`<script>` and `<deep-link>`)
 - Screen readers have a basic way of announcing lists. Ensuring proper list structure aids screen reader output. [Learn how to make lists accessible](#)
- **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 range list structures](#)
- **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 how to make tab index values correct](#)
- **Heading elements appear in a sequential-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 how to make heading structure correct](#)
- **Values assigned to `(aria-label)` 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 using ARIA `role` values, or `abstract` role, then the purpose of the element will not be communicated to users of assistive technologies. [Learn how to use ARIA roles](#)
- **Image elements do not have `(alt)` attributes that are redundant.**
 - Informative elements should be 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 how to use alt attributes](#)

NOT APPLICABLE (12)

- **(tabindex) values are unique**
 - Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. [Learn how to use access keys](#)
- **Elements with `(role="button")` or `(role="checkbox")` 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 dialogs announce their purpose](#)
- **(aria-hidden="true") elements do not contain focussable descendants**
 - Possible descendants within an `<aria-hidden="true">` element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn how to fix aria-hidden elements](#)
- **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 aria-label input fields](#)
- **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 make meter accessible](#)
- **ARIA progress elements have accessible names**
 - When a progress bar 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 progress accessible](#)
- **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 how to make sure ARIA roles and children have correct roles](#)
- **`(aria-label)` is contained by its 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 parents](#)
- **Elements with the `rel="noopener"` attribute do not have focussable descendants**
 - Adding `rel="noopener"` around a text node split by `mailto` enables VoiceOver to treat it as one phrase, but the element's focussable descendants will not be announced. [Learn how to fix the rel="noopener" bug](#)
- **ARIA logge fields have accessible names**
 - When a logon 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 how to make logon accessible](#)
- **ARIA `(role="button")` elements have accessible names**
 - When a button 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 buttons accessible](#)
- **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 how to use skip links](#)
- **`(alt)` contain only properly ordered `<div>` and `<hr>` groups, `<script>`, `<map>`, and `` elements.**
 - When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn how to structure definition lists](#)
- **Definition list items are wrapped in `<dt>` 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](#)
- **`(alt)` attributes on active, focussable elements are unique**
 - All focussable elements must have a unique `id` to ensure that they're visible to assistive technologies. [Learn how to use ARIA IDs](#)
- **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 use ARIA IDs](#)
- **No form fields have multiple labels**
 - Form fields with multiple labels can be incorrectly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn how to fix multi-label forms](#)
- **`<frame>` or `<iframe>` elements have a `title`**
 - Screen reader users rely on frame titles to describe the contents of frames. [Learn how to add frame titles](#)
- **`<input>` element has an `(alt="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 how to fix lang attributes](#)
- **Input buttons have descriptive text**
 - Adding descriptive and accessible text to input buttons may help screen reader users understand the purpose of the input before. [Learn how to use descriptive buttons](#)
- **`<input type="image">` elements have `(alt)` text**
 - `<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 more about Image alt text
<input type="radio"/> 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
<input type="radio"/> <object> elements have alternate text
Screen readers cannot translate non-text content. Adding alternative text to <object> elements helps screen readers convey meaning to users. Learn more about the object element
<input type="radio"/> 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
<input type="radio"/> Skip links are available.
Including a skip link can help users skip to the main content to save time. Learn more about skip links
<input type="radio"/> Tables have efficient context in the summary attribute and <code>caption</code> .
The summary attribute should describe the table structure, while <code>caption</code> is most useful near the document title. Accurate table markup helps users of screen readers. Learn more about the table element
<input type="radio"/> Cells in a <code>thead</code> element use the <code>headers</code> attribute to refer to table cells within the same table.
Screen readers have features to make navigating tables easier. Ensuring <th> cells using the <code>headers</code> attribute only refer to other cells in the same table may improve the experience for screen reader users. Learn more about the thead element
<input type="radio"/> <td> elements and elements with <code>role="cell"</code> have data cells they describe.
Screen readers have features to make navigating tables easier. Ensuring table headings always refer to some set of cells may improve the experience for screen reader users. Learn more about table headers
<input type="radio"/> <code>label</code> attributes have a valid value
Specifying a valid <code>label</code> on elements helps ensure that text is pronounced correctly by a screen reader. Learn more about the label element
<input type="radio"/> <video> elements contain a <code>caption</code> element with <code>label="caption"</code>
When a video provides a caption it is easier for deaf and hearing impaired users to access its information. Learn more about video captions
<input type="radio"/> 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
<input type="radio"/> Document has a main landmark.
One main landmark helps screen reader users navigate a web page. Learn more about landmarks
<input type="radio"/> 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
<input type="radio"/> Elements with visible test labels have matching accessible names.
Visible test labels that do not match the accessible name can result in a confusing experience for screen reader users. Learn more about visible test labels
<input type="radio"/> Tables use <code>caption</code> instead of cells with the <code>title</code> attribute to indicate a caption.
Screen readers have features to make navigating tables easier. Ensuring that table uses the actual caption element instead of cells with the <code>title</code> attribute may improve the experience for screen reader users. Learn more about captions
<input type="radio"/> <td> elements in a large <code>thead</code> have one or more table headers.
Screen readers have features to make navigating tables easier. Ensuring that <td> elements in a large table (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

93

Best Practices

GENERAL						
<input checked="" type="checkbox"/> Browser errors were 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 the errors in the console diagnostic audit						
<table border="1"><thead><tr><th>Source</th><th>Description</th></tr></thead><tbody><tr><td>vetuscode.js</td><td>Error: <code>WSDL</code> trying to set the <code>allowCloud</code> value from the Manifest: https://raw.githubusercontent.com/vercel/vercel/next@main/vercel/allowCloud.js (Resource size is not correct - type in the Manifest?)</td></tr></tbody></table>	Source	Description	vetuscode.js	Error: <code>WSDL</code> trying to set the <code>allowCloud</code> value from the Manifest: https://raw.githubusercontent.com/vercel/vercel/next@main/vercel/allowCloud.js (Resource size is not correct - type in the Manifest?)		
Source	Description					
vetuscode.js	Error: <code>WSDL</code> trying to set the <code>allowCloud</code> value from the Manifest: https://raw.githubusercontent.com/vercel/vercel/next@main/vercel/allowCloud.js (Resource size is not correct - type in the Manifest?)					
<input checked="" type="checkbox"/> Issues were logged 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.						
<table border="1"><thead><tr><th>Issue type</th></tr></thead><tbody><tr><td>Content security policy</td></tr></tbody></table>	Issue type	Content security policy				
Issue type						
Content security policy						
<input checked="" type="checkbox"/> Detected JavaScript libraries						
All front-end JavaScript libraries detected on the page. Learn more about this JavaScript library detection diagnostic audit						
<table border="1"><thead><tr><th>Name</th><th>Version</th></tr></thead><tbody><tr><td>Next.js</td><td>12.2.2</td></tr></tbody></table>	Name	Version	Next.js	12.2.2		
Name	Version					
Next.js	12.2.2					
TRUST AND SAFETY						
<input checked="" type="checkbox"/> 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 CSP						
<table border="1"><thead><tr><th>Description</th><th>Directive</th><th>Severity</th></tr></thead><tbody><tr><td>No CSP found in enforcement mode</td><td></td><td>High</td></tr></tbody></table>	Description	Directive	Severity	No CSP found in enforcement mode		High
Description	Directive	Severity				
No CSP found in enforcement mode		High				
PASSED AUDITS (12)						
<input checked="" type="checkbox"/> Use HTTPS						
All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding <code>http://</code> , where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS ensures that your users' data is encrypted in transit and provides a secure platform for your app and your users, and is a prerequisite for HTTPS2 and many new web platform APIs. Learn more about HTTPS						
<input checked="" type="checkbox"/> Avoid deprecated APIs						

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

- **Avoid third-party cookies**
Support for third-party cookies will be removed in a future version of Chrome. [Learn more about third-party cookies](#)
- **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 paste input fields](#)
- **Avoid requesting the permission on page load**
Users are irritated if or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more about the permission requests](#)
- **Avoid requesting the notification permission on page load**
Users are irritated if or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more about the permission requests](#)
- **Display images with correct aspect ratio**
Image display dimensions should match natural aspect ratio. [Learn more about aspect ratio](#)
- **Serve images with appropriate resolution**
Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn more about image resolution](#)
- **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 `charset` tag in the first 1024 bytes of the HTML, or in the Content-Type HTTP response header. [Learn more about defining the charset encoding](#)
- **Avoid unhandled event listeners**
The unhandled event does not have a handler and listening for it can prevent browser optimizations like the Back/Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more about event listeners](#)
- **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](#)

Show 3rd-party resources (?)

Map URL

Loom - Screen Recorder & Screen Capture ([Chrome Extension](#))

```
chrome-extension://laedabkkihcheddell1bbd1z4z/extension.js  
$0:4 [x] Failed fetching source map (null)
```

Unreachable

```
chrome-extension://hfhfdchjeapjgjlephqgjljamja/extension.js  
$0:4 [x] Failed fetching source map (null)
```

Show 3rd-party resources (?)

NOT APPLICABLE (0) Hide

□ Fonts with `font-display: optional` are preloadedPreload opt (font) fonts as first-time visitors may use them. [Learn more about optional fonts](#)

91

SEO

These checks ensure that your site is following best search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Page Speed](#), [Accessibility](#), and [Security](#).

CONTENT BEST PRACTICES

⚠ Links do not have descriptive text — 1 link found

Descriptive link text helps search engines understand your content. [Learn how to write better link text](#)

Link destination	Link Text
<code>verifichashcode()</code>	<code>verifichashcode()</code>
<code>about-us</code>	Learn More

Format your HTML in a way that enables crawlers to better understand your app's content.

ADDITIONAL ITEMS TO MANUALLY CHECK (0) Hide

○ Structured data is valid

Run the [Structured Data Test 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 (10) Hide

- Has a `<meta name="viewport">` tag with `width` or `initial-scale`
- Document has a `<title>` element
- Document has a `<meta name="description">`
- Page has successful HTTP status code
- Links are crawlable
- Page isn't blocked from indexing
- robots.txt is valid

A meta `name="viewport"` not only optimizes your app for mobile screen sizes, but also prevents a 300px minimum width being applied. [Learn more about the viewport tag](#)

Search engines may use `link` attributes on links to crawl websites. Ensure that the `link` attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn how to create link crawlability](#)

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](#)

The screenshot displays the Lighthouse audit results for the URL "about:blank". The interface is divided into several sections:

- Core Web Vitals**:
 - Score: 100
 - Priority: Low
 - Details:
 - Image elements have alt attributes.
 - Document has a valid hreflang.
 - Document avoids plugins.
- Accessibility**:
 - Score: 100
 - Priority: Low
 - Details:
 - Document has a valid rel=canonical.
 - Document uses legible font sizes.
 - Tap targets are sized appropriately.
- Performance**:
 - Score: 100
 - Priority: Low
 - Details:
 - Captured at Apr 1, 2024, 7:08 Emulated Desktop with Lighthouse 11.0.0 Using Chromium 123.0.0.0
 - Initial page load
 - Single page session
 - With desktop

NOT APPLICABLE [?]

Hide

Generated by Lighthouse 11.0.0 | [https://lighthouse.dev](#)