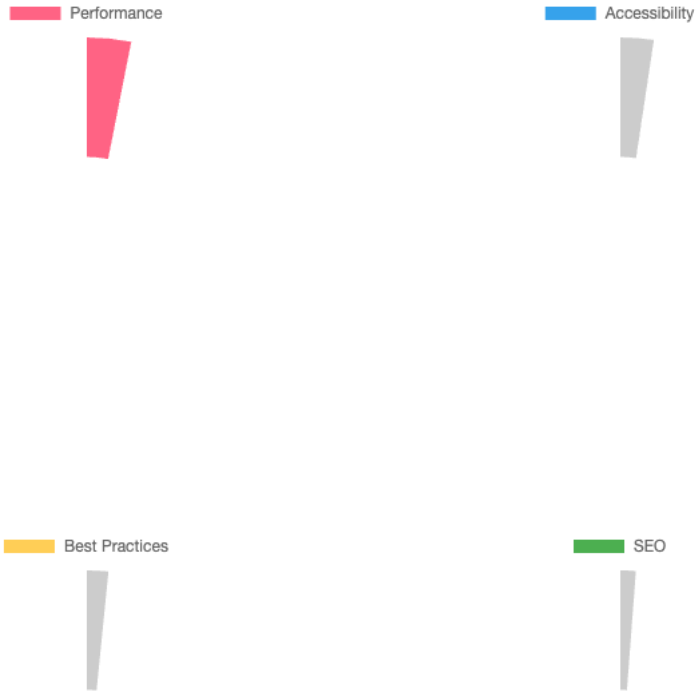
PageSpeed Report

Generated on: 2024-07-30T20:06:59.001Z



Title	Description	Score	
Cumulative Layout Shift	Cumulative Layout Shift measures the movement of visible elements within the viewport. [Learn more about the Cumulative Layout Shift metric] (https://web.dev/articles/cls).	100.00	{"type":"debugdata","item
Tasks	Lists the toplevel main thread tasks that executed during page load.	N/A	{"items":[{"duration":15.06
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] (https://developer.chrome.com/docs/lighthouse/performance/uses-webp-images/).	N/A	{"sortedBy":["wastedBytes
Resources Summary	Aggregates all network requests and groups them by type	N/A	{"type":"table","headings"
Largest Contentful Paint	Largest Contentful Paint marks the time at which the largest text or image is painted. [Learn more about the Largest Contentful Paint metric] (https://developer.chrome.com/docs/lighthouse/performance/lighthouse-largest-contentful-paint/)	95.00	No details available
Time to Interactive	Time to Interactive is the amount of time it takes for the page to become fully interactive. [Learn more about the Time to Interactive metric] (https://developer.chrome.com/docs/lighthouse/performance/interactive/).	100.00	No details available
Minimizes main-thread work	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] (https://developer.chrome.com/docs/lighthouse/performance/mainthread-work-breakdown/)	N/A	{"sortedBy":["duration"],"it
Properly size images	Serve images that are appropriately-sized to save cellular data and improve load time. [Learn how to size images] (https://developer.chrome.com/docs/lighthouse/performance/uses-responsive-images/).	N/A	{"headings":[{"key":"node"

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()](https://developer.chrome.com/docs/lighthouse/best-practices/no-document-write/).	N/A	{"headings":[],"items":[],"ty
Network Requests	Lists the network requests that were made during page load.	N/A	{"headings":[{"label":"URL
First Contentful Paint	First Contentful Paint marks the time at which the first text or image is painted. [Learn more about the First Contentful Paint metric] (https://developer.chrome.com/docs/lighthouse/performance/first-contentful-paint/).	94.00	No details available
Final Screenshot	The last screenshot captured of the pageload.	N/A	{"type":"screenshot","data
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] (https://developer.chrome.com/docs/lighthouse/performance/render-blocking-resources/).	N/A	{"headings":[{"key":"url","v
Avoid multiple page redirects	Redirects introduce additional delays before the page can be loaded. [Learn how to avoid page redirects] (https://developer.chrome.com/docs/lighthouse/performance/redirects/).	N/A	{"overallSavingsMs":0,"ty
Total Blocking Time	Sum of all time periods between FCP and Time to Interactive, when task length exceeded 50ms, expressed in milliseconds. [Learn more about the Total Blocking Time metric] (https://developer.chrome.com/docs/lighthouse/performance/lighthouse-total-blocking-time/).	100.00	No details available
Largest Contentful Paint element	This is the largest contentful element painted within the viewport. [Learn more about the Largest Contentful Paint element] (https://developer.chrome.com/docs/lighthouse/performance/lighthouse-largest-contentful-paint/)	N/A	{"items":[{"headings":[{"va
Reduce unused JavaScript	Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn how to reduce unused JavaScript](https://developer.chrome.com/docs/lighthouse/performance/unused-javascript/).	N/A	{"overallSavingsBytes":73
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] (https://developer.chrome.com/docs/lighthouse/performance/third-party-facades/).	N/A	No details available
Avoid long main-thread tasks	Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay. [Learn how to avoid long main-thread tasks] (https://web.dev/articles/long-tasks-devtools)	N/A	{"skipSumming":["startTim
Avoid chaining critical requests	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] (https://developer.chrome.com/docs/lighthouse/performance/critical-request-chains/).	N/A	{"chains":{"BCF740447E0
First Meaningful Paint	First Meaningful Paint measures when the primary content of a page is visible. [Learn more about the First Meaningful Paint metric] (https://developer.chrome.com/docs/lighthouse/performance/first-meaningful-paint/).	94.00	No details available
Efficiently encode images	Optimized images load faster and consume less cellular data. [Learn how to efficiently encode images] (https://developer.chrome.com/docs/lighthouse/performance/uses-optimized-images/).	N/A	{"overallSavingsBytes":0,
Speed Index	Speed Index shows how quickly the contents of a page are visibly populated. [Learn more about the Speed Index metric] (https://developer.chrome.com/docs/lighthouse/performance/speed-index/).	99.00	No details available
Preconnect 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](https://developer.chrome.com/docs/lighthouse/performance/uses-rel-preconnect/).	N/A	{"items":[],"type":"opportu
Metrics	Collects all available metrics.	N/A	{"items":[{"observedLarge
JavaScript execution time	Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn how to reduce Javascript execution time] (https://developer.chrome.com/docs/lighthouse/performance/bootup-time/).	N/A	{"type":"table","sortedBy":
Avoid large layout shifts	These are the largest layout shifts observed on the page. Each table item represents a single layout shift, and shows the element that shifted the most. Below each item are possible root causes that led to the layout shift. Some of these layout shifts may not be included in the CLS metric value due to [windowing](https://web.dev/articles/cls#what_is_cls). [Learn how to improve CLS](https://web.dev/articles/optimize-cls)	N/A	{"items":[{"score":0.00489 "}},{"subItems":{"iten ","selector":"di ","path":
Minify JavaScript	Minifying JavaScript files can reduce payload sizes and script parse time. [Learn how to minify JavaScript] (https://developer.chrome.com/docs/lighthouse/performance/unminified-javascript/).	N/A	{"debugData":{"metricSav

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](https://developer.chrome.com/docs/lighthouse/performance/user-timings/).	N/A	{"headings":[],"items":[],"ty
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] (https://developer.chrome.com/docs/lighthouse/performance/unused-css-rules/).	N/A	{"type":"opportunity","sort
Server Backend Latencies	Server latencies can impact web performance. If the server latency of an origin is high, it's an indication the server is overloaded or has poor backend performance. [Learn more about server response time](https://hpbn.co/primer-on-web-performance/#analyzing-the-resource-waterfall).	N/A	{"items":[{"origin":"https://
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] (https://web.dev/articles/optimize-lcp#optimize_when_the_resource_is_discovered).	N/A	{"debugData":{"type":"deb
Avoid an excessive DOM size	A large DOM will increase memory usage, cause longer [style calculations] (https://developers.google.com/web/fundamentals/performance/rendering/reduce-the-scope-and-complexity-of-style-calculations), and produce costly [layout reflows](https://developers.google.com/speed/articles/reflow). [Learn how to avoid an excessive DOM size] (https://developer.chrome.com/docs/lighthouse/performance/dom-size/).	N/A	{"headings":[{"valueType" ","selector":"main > div.co
Has a `` tag with `width` or `initial-scale`	A `` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](https://developer.chrome.com/blog/300ms-tap-delay-gone-away/). [Learn more about using the viewport meta tag] (https://developer.chrome.com/docs/lighthouse/pwa/viewport/).	N/A	{"viewportContent":"width
Avoid non-composited animations	Animations which are not composited can be janky and increase CLS. [Learn how to avoid non-composited animations] (https://developer.chrome.com/docs/lighthouse/performance/non-composited-animations/)	N/A	{"items":[],"type":"table","h
Screenshot Thumbnails	This is what the load of your site looked like.	N/A	{"type":"filmstrip","items":[{"data":"data:image/jpeg;b {"timing":1125,"data":"data {"timestamp":247192690 {"data":"data:image/jpeg;b {"timestamp":247193440 {"timing":2625,"data":"dat {"timestamp":247194190
Script Treemap Data	Used for treemap app	N/A	{"type":"treemap-data","no
Max Potential First Input Delay	The maximum potential First Input Delay that your users could experience is the duration of the longest task. [Learn more about the Maximum Potential First Input Delay metric] (https://developer.chrome.com/docs/lighthouse/performance/lighthouse-max-potential-fid/).	100.00	No details available
Diagnostics	Collection of useful page vitals.	N/A	{"type":"debugdata","item
Avoids enormous network payloads	Large network payloads cost users real money and are highly correlated with long load times. [Learn how to reduce payload sizes] (https://developer.chrome.com/docs/lighthouse/performance/total-byte-weight/).	N/A	{"items":[{"totalBytes":125
Enable text compression	Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn more about text compression] (https://developer.chrome.com/docs/lighthouse/performance/uses-text-compression/).	N/A	{"items":[],"overallSavings
Defer offscreen 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](https://developer.chrome.com/docs/lighthouse/performance/offscreen-images/).	N/A	{"items":[],"headings":[],"d
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] (https://web.dev/articles/optimize-cls#images_without_dimensions)	N/A	{"headings":[],"items":[],"ty
Remove duplicate modules in JavaScript bundles	Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity.	N/A	{"debugData":{"metricSav
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 the Time to First Byte metric] (https://developer.chrome.com/docs/lighthouse/performance/time-to-first-byte/).	N/A	{"type":"opportunity","hea
Minify CSS	Minifying CSS files can reduce network payload sizes. [Learn how to minify CSS] (https://developer.chrome.com/docs/lighthouse/performance/unminified-css/).	N/A	{"debugData":{"type":"deb
Ensure text remains visible during webfont load	Leverage the `font-display` CSS feature to ensure text is user-visible while webfonts are loading. [Learn more about `font-display`] (https://developer.chrome.com/docs/lighthouse/performance/font-display/).	N/A	{"headings":[{"valueType"

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] (https://developer.chrome.com/docs/lighthouse/performance/efficient-animated-content/)	N/A	{"sortedBy":["wastedBytes
Serve static assets with an efficient cache policy	A long cache lifetime can speed up repeat visits to your page. [Learn more about efficient cache policies] (https://developer.chrome.com/docs/lighthouse/performance/uses-long-cache-ttl/).	N/A	{"summary":{"wastedByte
Network Round Trip Times	Network round trip times (RTT) have a large impact on performance. If the RTT to an origin is high, it's an indication that servers closer to the user could improve performance. [Learn more about the Round Trip Time](https://hpbn.co/primer-on-latency-and-bandwidth/).	N/A	{"items":[{"rtt":4.63779,"or
Minimize third-party usage	Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading. [Learn how to minimize third-party impact] (https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/loading-third-party-javascript/).	N/A	{"type":"table","summary"
Does not 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 more about adopting passive event listeners](https://developer.chrome.com/docs/lighthouse/best-practices/uses-passive-event-listeners/).	N/A	{"type":"table","headings"
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](https://web.dev/articles/lcp-lazy-loading).	N/A	{"headings":[{"label":"Eler
Avoid serving legacy JavaScript to modern browsers	Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. [Learn how to use modern JavaScript](https://web.dev/articles/publish-modern-javascript)	N/A	{"type":"opportunity","hea