



## Performance

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

▲ 0–49 50–89 90–100

Current Employees									
First Name	Last Name	Department	Salary	Job	City	State	Country	Phone	Fax
John	Deere	Engineering	120000	Software Engineer	Chicago	IL	USA	312.555.1234	312.555.5678
Jane	Smith	Marketing	85000	Marketing Manager	Chicago	IL	USA	312.555.9012	312.555.3456
Mike	Johnson	Sales	75000	Sales Representative	Chicago	IL	USA	312.555.7890	312.555.2345
Emily	White	Human Resources	65000	HR Specialist	Chicago	IL	USA	312.555.6789	312.555.0123
David	Black	Finance	90000	Financial Analyst	Chicago	IL	USA	312.555.4567	312.555.8901
Alice	Green	Operations	70000	Operations Manager	Chicago	IL	USA	312.555.2109	312.555.5432
Bob	Brown	IT	80000	IT Support	Chicago	IL	USA	312.555.9876	312.555.1098
Charlie	Davis	Product Development	110000	Product Manager	Chicago	IL	USA	312.555.3210	312.555.6543
Diana	Evens	Customer Support	60000	Customer Support	Chicago	IL	USA	312.555.0987	312.555.4321

### METRICS

Expand view

First Contentful Paint

2.1 s

Largest Contentful Paint

2.1 s

Total Blocking Time

0 ms

Cumulative Layout Shift

0

Speed Index

2.1 s

View Treemap

Resource	Size	Time to First Byte	Time to Interactive	Time to First Contentful Paint	Time to Largest Contentful Paint	Time to Next Paint	Time to Full Page Load
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms
Current Employees	100 KB	100 ms	200 ms	200 ms	200 ms	200 ms	200 ms

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

### DIAGNOSTICS

▲ Eliminate render-blocking resources — Potential savings of 980 ms

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](#)

URL	Transfer Size	Potential Savings
Google CDN <span>Cdn</span>	30.4 KiB	1,070 ms
...3.5.1/jquery.min.js (ajax.googleapis.com)	30.4 KiB	1,070 ms
datatables.net	29.8 KiB	1,090 ms
...js/jquery.dataTables.min.js (cdn.datatables.net)	27.7 KiB	300 ms
...css/jquery.dataTables.min.css (cdn.datatables.net)	2.1 KiB	790 ms

▲ Does not have a `<meta name="viewport">` tag with `width` or `initial-scale` No `<meta name="viewport">` tag found ^

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

Reduce unused JavaScript — Potential savings of 34 KiB ^

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

URL	Transfer Size	Potential Savings
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/renderer.js	38.8 KiB	34.2 KiB

○ Initial server response time was short — Root document took 260 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 <span>1st Party</span>	260 ms
/employee-list.html (hrnet-jquery-two.vercel.app)	260 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 63 KiB

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

☒ Show 3rd-party resources (5)

URL	Transfer Size
datatables.net	30.8 KiB
...js/jquery.dataTables.min.js (cdn.datatables.net)	27.7 KiB
...css/jquery.dataTables.min.css (cdn.datatables.net)	2.1 KiB
...images/sort_both.png (cdn.datatables.net)	0.5 KiB
...images/sort_asc.png (cdn.datatables.net)	0.4 KiB
Google CDN <span>Cdn</span>	30.4 KiB
...3.5.1/jquery.min.js (ajax.googleapis.com)	30.4 KiB
vercel.app <span>1st Party</span>	1.9 KiB
/employee-list.html (hrnet-jquery-two.vercel.app)	0.8 KiB
/employee-list.js (hrnet-jquery-two.vercel.app)	0.7 KiB
/app.css (hrnet-jquery-two.vercel.app)	0.4 KiB

○ Avoids an excessive DOM size — 103 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		103
Maximum DOM Depth	option	7
Maximum Child Elements	<div><div></div><div></div><div></div></div> tr	9

Statistic	Element	Value

○ Avoid chaining critical requests — 5 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.](#)

Maximum critical path latency: **392.345 ms**

*Initial Navigation*

/employee-list.html (hrnet-jquery-two.vercel.app)  
...css/jquery.dataTables.min.css (cdn.datatables.net) - **74.623 ms, 2.09 KiB**  
/app.css (hrnet-jquery-two.vercel.app) - **28.774 ms, 0.38 KiB**  
...3.5.1/jquery.min.js (ajax.googleapis.com) - **26.546 ms, 30.37 KiB**  
...js/jquery.dataTables.min.js (cdn.datatables.net) - **45.264 ms, 27.72 KiB**  
/employee-list.js (hrnet-jquery-two.vercel.app) - **25.205 ms, 0.72 KiB**

○ JavaScript execution time — 0.4 s



Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn how to reduce Javascript execution time.](#) TBT

☒ Show 3rd-party resources (1)

URL	Total CPU Time	Script Evaluation	Script Parse
vercel.app <span>1st Party</span>	491 ms	200 ms	60 ms
/employee-list.html (hrnet-jquery-two.vercel.app)	491 ms	200 ms	60 ms
Google CDN <span>Cdn</span>	232 ms	144 ms	4 ms
...3.5.1/jquery.min.js (ajax.googleapis.com)	232 ms	144 ms	4 ms
Unattributable	72 ms	12 ms	0 ms
Unattributable	72 ms	12 ms	0 ms

○ Minimizes main-thread work — 0.9 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

Category	Time Spent
Script Evaluation	384 ms
Other	247 ms
Style & Layout	77 ms
Script Parsing & Compilation	70 ms
Parse HTML & CSS	69 ms
Rendering	6 ms


Minimize third-party usage
— Third-party code blocked the main thread for 60 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 minimize third-party impact.](#) TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
Google CDN <span>Cdn</span>	<b>30 KiB</b>	<b>60 ms</b>
...3.5.1/jquery.min.js (ajax.googleapis.com)	30 KiB	60 ms
datatables.net	<b>31 KiB</b>	<b>0 ms</b>
...js/jquery.dataTables.min.js (cdn.datatables.net)	28 KiB	0 ms
...css/jquery.dataTables.min.css (cdn.datatables.net)	2 KiB	0 ms
...images/sort_both.png (cdn.datatables.net)	1 KiB	0 ms
...images/sort_asc.png (cdn.datatables.net)	0 KiB	0 ms

Largest Contentful Paint element
— 2,080 ms

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

Element
<div> <div>  </div> <div> h1 </div> </div>

Phase	% of LCP	Timing
TTFB	30%	630 ms
Load Delay	0%	0 ms
Load Time	0%	0 ms
Render Delay	70%	1,450 ms

○ Avoid long main-thread tasks — 3 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 <span>1st Party</span>		319 ms
/employee-list.html (hrnet-jquery-two.vercel.app)	920 ms	175 ms
/employee-list.html (hrnet-jquery-two.vercel.app)	776 ms	144 ms
Google CDN <span>Cdn</span>		90 ms
...3.5.1/jquery.min.js (ajax.googleapis.com)	1,988 ms	90 ms

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

PASSED AUDITS (25)

Hide

### Properly size images



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

### 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.](#) FCP LCP

### Minify CSS



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

### Minify JavaScript



Minifying JavaScript files can reduce payload sizes and script parse time. <a href="#">Learn how to minify JavaScript.</a> <span>FCP</span> <span>LCP</span>	
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. <a href="#">Learn how to reduce unused CSS.</a> <span>FCP</span> <span>LCP</span>	
Efficiently encode images	^
Optimized images load faster and consume less cellular data. <a href="#">Learn how to efficiently encode images.</a> <span>FCP</span> <span>LCP</span>	
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. <a href="#">Learn more about modern image formats.</a> <span>FCP</span> <span>LCP</span>	
Enable text compression	^
Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. <a href="#">Learn more about text compression.</a> <span>FCP</span> <span>LCP</span>	
Preconnect to required origins	^
Consider adding preconnect or dns-prefetch resource hints to establish early connections to important third-party origins. <a href="#">Learn how to preconnect to required origins.</a> <span>LCP</span> <span>FCP</span>	
Avoid multiple page redirects	^
Redirects introduce additional delays before the page can be loaded. <a href="#">Learn how to avoid page redirects.</a> <span>LCP</span> <span>FCP</span>	
Use HTTP/2	^
HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. <a href="#">Learn more about HTTP/2.</a> <span>LCP</span> <span>FCP</span>	
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. <a href="#">Learn more about efficient video formats</a> <span>FCP</span> <span>LCP</span>	
Remove duplicate modules in JavaScript bundles	^
Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. <span>FCP</span> <span>LCP</span>	
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](#) FCP LCP

☐ 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 text is user-visible while webfonts are loading. [Learn more about font-display.](#)

☐ 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 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](#). [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 which are not composited can be janky 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

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](#)

 Captured at Aug 5, 2024, 3:09 PM GMT+2

 Initial page load

 Emulated Moto G Power with Lighthouse 12.0.0

 Slow 4G throttling

 Single page session

 Using Chromium 127.0.0.0 with devtools