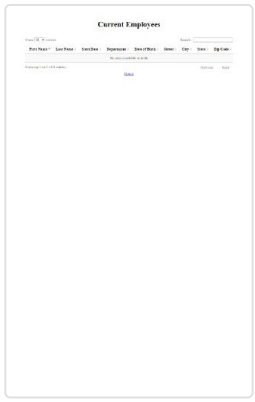
http://127.0.0.1:5500/employee-list.html

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 Expand view

■ First Contentful Paint
2.3 s

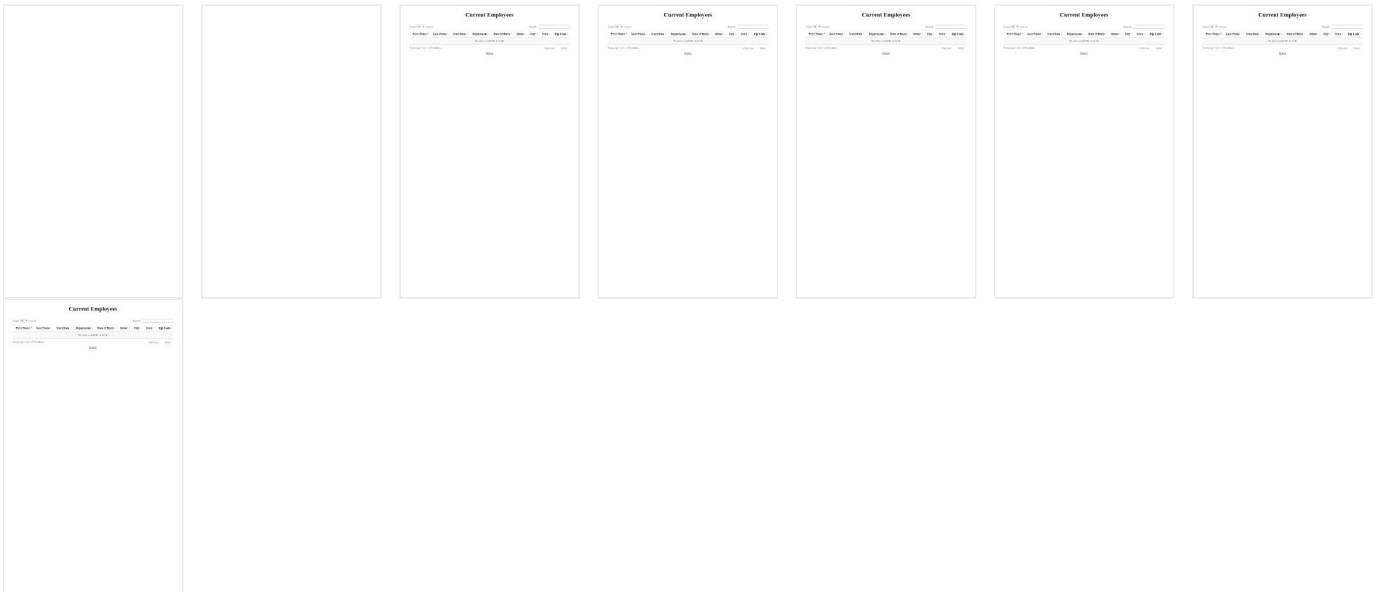
● Largest Contentful Paint
2.3 s

● Total Blocking Time
40 ms

● Cumulative Layout Shift
0

● Speed Index
2.6 s

View Treemap



Show audits relevant to: All FCP LCP TBT CLS

DIAGNOSTICS

▲ Eliminate render-blocking resources — Potential savings of 1,550 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

☑ Show 3rd-party resources (3)

URL	Transfer Size	Potential Savings
datatables.net	**31.0 KiB**	**1,260 ms**
…css/jquery.dataTables.min.css (cdn.datatables.net)	2.4 KiB	880 ms
…js/jquery.dataTables.min.js (cdn.datatables.net)	28.5 KiB	380 ms
0.1 1st Party	**1.7 KiB**	**450 ms**
/app.css (127.0.0.1)	0.7 KiB	150 ms
/employee-list.js (127.0.0.1)	1.0 KiB	300 ms
Google CDN Cdn	**30.9 KiB**	**1,530 ms**
…3.5.1/jquery.min.js (ajax.googleapis.com)	30.9 KiB	1,530 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. TBT

■ Page prevented back/forward cache restoration — 1 failure reason

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

Failure reason	Failure type
Pages with WebSocket cannot enter back/forward cache. /employee-list.html (127.0.0.1)	Pending browser support

○ Minimize third-party usage — Third-party code blocked the main thread for 100 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 Cdn	**31 KiB**	**98 ms**
…3.5.1/jquery.min.js (ajax.googleapis.com)	31 KiB	98 ms
datatables.net	**32 KiB**	**0 ms**
…js/jquery.dataTables.min.js (cdn.datatables.net)	29 KiB	0 ms

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
Google CDN Cdn	**296 ms**	**202 ms**	**7 ms**
…3.5.1/jquery.min.js (ajax.googleapis.com)	296 ms	202 ms	7 ms
0.1 1st Party	**295 ms**	**36 ms**	**98 ms**
/employee-list.html (127.0.0.1)	295 ms	36 ms	98 ms
Unattributable	**175 ms**	**33 ms**	**0 ms**
`Unattributable`	175 ms	33 ms	0 ms

Minimizes main-thread work — 0.8 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	299 ms
Other	210 ms
Style & Layout	141 ms

Category	Time Spent
Script Parsing & Compilation	112 ms
Rendering	21 ms
Parse HTML & CSS	20 ms

Avoid long main-thread tasks — 5 long tasks found

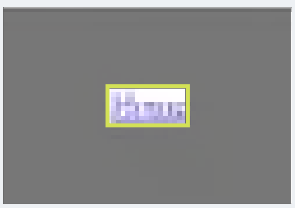
Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay. Learn how to avoid long main-thread tasks TBT

Show 3rd-party resources (3)

URL	Start Time	Duration
Google CDN Cdn		**175 ms**
…3.5.1/jquery.min.js (ajax.googleapis.com)	2,273 ms	115 ms
…3.5.1/jquery.min.js (ajax.googleapis.com)	2,213 ms	60 ms
0.1 1st Party		**138 ms**
/employee-list.html (127.0.0.1)	602 ms	84 ms
/employee-list.html (127.0.0.1)	722 ms	54 ms
datatables.net		**64 ms**
…js/jquery.dataTables.min.js (cdn.datatables.net)	1,759 ms	64 ms

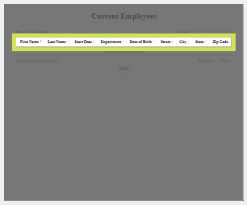
Avoid large layout shifts — 1 element found

These DOM elements contribute most to the CLS of the page. Learn how to improve CLS CLS

Element	CLS Contribution
 a	0.000

Avoids an excessive DOM size — 35 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		35
Maximum DOM Depth	option	7
Maximum Child Elements	 tr	9

Initial server response time was short — Root document took 0 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
0.1 1st Party	**0 ms**
/employee-list.html (127.0.0.1)	0 ms

Avoids enormous network payloads — Total size was 67 KiB

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

Show 3rd-party resources (5)

URL	Transfer Size
datatables.net	**31.6 KiB**
…js/jquery.dataTables.min.js (cdn.datatables.net)	28.5 KiB
…css/jquery.dataTables.min.css (cdn.datatables.net)	2.4 KiB
…images/sort_both.png (cdn.datatables.net)	0.3 KiB

URL	Transfer Size
…images/sort_asc.png (cdn.datatables.net)	0.3 KiB
Google CDN Cdn	**30.9 KiB**
…3.5.1/jquery.min.js (ajax.googleapis.com)	30.9 KiB
0.1 1st Party	**4.2 KiB**
/employee-list.html (127.0.0.1)	2.5 KiB
/employee-list.js (127.0.0.1)	1.0 KiB
/app.css (127.0.0.1)	0.7 KiB

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. FCP LCP

Maximum critical path latency: **802.024 ms**

Initial Navigation

- /employee-list.html (127.0.0.1)
  - …css/jquery.dataTables.min.css (cdn.datatables.net) **- 158.315 ms, 2.41 KiB**
  - /app.css (127.0.0.1) **- 11.569 ms, 0.67 KiB**
  - …3.5.1/jquery.min.js (ajax.googleapis.com) **- 778.774 ms, 30.91 KiB**
  - …js/jquery.dataTables.min.js (cdn.datatables.net) **- 179.517 ms, 28.54 KiB**
  - /employee-list.js (127.0.0.1) **- 7.439 ms, 1.02 KiB**

Largest Contentful Paint element — 2,330 ms

This is the largest contentful element painted within the viewport. Learn more about the Largest Contentful Paint element LCP

Element	
	h1

Phase	% of LCP	Timing
TTFB	19%	450 ms
Load Delay	0%	0 ms
Load Time	0%	0 ms
Render Delay	81%	1,880 ms

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

PASSED AUDITS (26) Hide

- Properly size images

  Serve images that are appropriately-sized to save cellular data and improve load time. Learn how to size images.

- 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.

- 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. Learn how to minify JavaScript. FCP LCP

- Reduce unused CSS

  Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. Learn how to reduce unused CSS. FCP LCP

- 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. LCP

- Efficiently encode images

  Optimized images load faster and consume less cellular data. Learn how to efficiently encode images.

- Serve images in next-gen formats

  Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. Learn more about modern image formats.

- Enable text compression

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

- 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. FCP LCP

- Avoid multiple page redirects

  Redirects introduce additional delays before the page can be loaded. Learn how to avoid page redirects. FCP LCP

- Preload key requests

  Consider using `<link rel=preload>` to prioritize fetching resources that are currently requested later in page load. Learn how to preload key requests. FCP LCP

- Use HTTP/2

  HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. Learn more about HTTP/2.

- Use video formats for animated content

  Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. Learn more about efficient video formats LCP

- Remove duplicate modules in JavaScript bundles

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

- 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 TBT

- Preload Largest Contentful Paint image

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

- Uses efficient cache policy on static assets — 0 resources found

A long cache lifetime can speed up repeat visits to your page. Learn more about efficient cache policies.

- 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.

- 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`. FCP LCP

- Lazy load third-party resources with facades

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

- Largest Contentful Paint image was not lazily loaded

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. Learn more about optimal lazy loading. LCP

- 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

Captured at Jan 24, 2024, 9:03 AM GMT+4	Emulated Moto G Power with Lighthouse 11.2.0	Single page load
Initial page load	Slow 4G throttling	Using Chromium 120.0.0.0 with devtools