https://joskapotin.github.io/joskapotin_14_27052022/

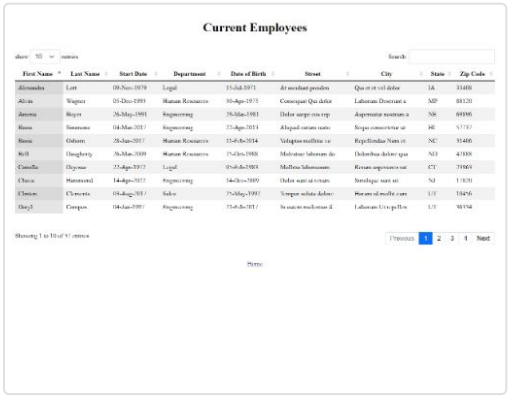
There were issues affecting this run of Lighthouse:

- **There may be stored data affecting loading performance in this location: IndexedDB. Audit this page in an incognito window to prevent those resources from affecting your scores.**

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 0.4 s	Time to Interactive 0.4 s
Speed Index 0.5 s	Total Blocking Time 0 ms
Largest Contentful Paint 0.4 s	Cumulative Layout Shift 0.017

View Original Trace View Treemap

Show audits relevant to: All FCP TBT LCP CLS

DIAGNOSTICS

Serve static assets with an efficient cache policy — 5 resources found

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

URL	Cache TTL	Transfer Size
…assets/index.bf114b5e.js (joskapotin.github.io)	10 m	61 KiB
…assets/EmployeesList.4c4fdad2.js (joskapotin.github.io)	10 m	4 KiB
…assets/api.587bd847.js (joskapotin.github.io)	10 m	3 KiB
…assets/EmployeesList.18b84f65.css (joskapotin.github.io)	10 m	2 KiB
…assets/index.0f11f824.css (joskapotin.github.io)	10 m	1 KiB

Avoid chaining critical requests — 4 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 more. FCP LCP

Maximum critical path latency: **500 ms**

Initial Navigation

/joskapotin_14_27052022/ (joskapotin.github.io)

…assets/index.0f11f824.css (joskapotin.github.io) **- 140 ms, 1.41 KiB**

…assets/index.bf114b5e.js (joskapotin.github.io)

…assets/EmployeesList.18b84f65.css (joskapotin.github.io) **- 40 ms, 2.22 KiB**

…assets/EmployeesList.4c4fdad2.js (joskapotin.github.io) **- 40 ms, 3.54 KiB**

…assets/api.587bd847.js (joskapotin.github.io) **- 150 ms, 2.65 KiB**

Keep request counts low and transfer sizes small — 6 requests • 71 KiB

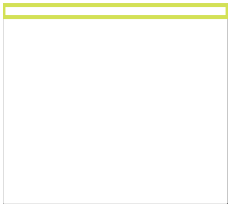
To set budgets for the quantity and size of page resources, add a budget.json file. Learn more.

Resource Type	Requests	Transfer Size
Total	6	70.9 KiB
Script	3	66.8 KiB
Stylesheet	2	3.6 KiB
Document	1	0.4 KiB
Image	0	0.0 KiB

Resource Type	Requests	Transfer Size
Media	0	0.0 KiB
Font	0	0.0 KiB
Other	0	0.0 KiB
Third-party	0	0.0 KiB

Largest Contentful Paint element — 1 element found

This is the largest contentful element painted within the viewport. Learn More LCP

Element
 h1

Avoid large layout shifts — 1 element found

These DOM elements contribute most to the CLS of the page. CLS

Element	CLS Contribution
 p.text-center	0.017

Avoid long main-thread tasks — 1 long task found

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

URL	Start Time	Duration
chrome-extension://bnjjngeaknajbdcgpfkgnonkmifihfo/build/content-script.js	356 ms	63 ms

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

PASSED AUDITS (34) Hide

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 more. FCP LCP

Properly size images

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

Defer offscreen images

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. Learn more.

Minify CSS

Minifying CSS files can reduce network payload sizes. Learn more. FCP LCP

Minify JavaScript

Minifying JavaScript files can reduce payload sizes and script parse time. Learn more. 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 more. FCP LCP

Reduce unused JavaScript — Potential savings of 20 KiB

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. Learn more. LCP

URL	Transfer Size	Potential Savings
…assets/index.bf114b5e.js (joskapotin.github.io)	60.6 KiB	20.3 KiB

Efficiently encode images

Optimized images load faster and consume less cellular data. Learn more.

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.

Enable text compression

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

Preconnect to required origins

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. Learn more. FCP LCP

Initial server response time was short — Root document took 130 ms

Keep the server response time for the main document short because all other requests depend on it. Learn more. FCP LCP

URL	Time Spent
/joskapotin_14_27052022/ (joskapotin.github.io)	130 ms

Avoid multiple page redirects

Redirects introduce additional delays before the page can be loaded. Learn more. FCP LCP

Preload key requests

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

Use HTTP/2

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

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 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 More TBT

Preload Largest Contentful Paint image

Preload the image used by the LCP element in order to improve your LCP time. Learn more. LCP

Avoids enormous network payloads — Total size was 71 KiB

Large network payloads cost users real money and are highly correlated with long load times. Learn more. LCP

URL	Transfer Size
…assets/index.bf114b5e.js (joskapotin.github.io)	60.6 KiB
…assets/EmployeesList.4c4fdad2.js (joskapotin.github.io)	3.5 KiB
…assets/api.587bd847.js (joskapotin.github.io)	2.6 KiB
…assets/EmployeesList.18b84f65.css (joskapotin.github.io)	2.2 KiB
…assets/index.0f11f824.css (joskapotin.github.io)	1.4 KiB
/joskapotin_14_27052022/ (joskapotin.github.io)	0.4 KiB

Avoids an excessive DOM size — 146 elements

A large DOM will increase memory usage, cause longer style calculations, and produce costly layout reflows. Learn more. TBT

Statistic	Element	Value
Total DOM Elements		146

Statistic	Element	Value
Maximum DOM Depth	 th.sorting.sorting_asc	9
Maximum Child Elements	 tbody.table-group-divider	10

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.

JavaScript execution time — 0.1 s

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. Learn more. TBT

URL	Total CPU Time	Script Evaluation	Script Parse
`chrome-extension://bnjjngeaknajbdcgpfkgnonkmifihfo/build/content-script.js`	76 ms	42 ms	30 ms
/joskapotin_14_27052022/ (joskapotin.github.io)	63 ms	3 ms	1 ms

Minimizes main-thread work — 0.2 s

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. Learn more TBT

Category	Time Spent
Script Evaluation	87 ms
Other	54 ms

Category	Time Spent
Style & Layout	42 ms
Script Parsing & Compilation	31 ms
Rendering	7 ms
Parse HTML & CSS	2 ms

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. FCP LCP

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 more. TBT

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 more. 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.

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.

Avoids `document.write()`

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. Learn more.

Avoid non-composited animations

Animations which are not composited can be janky and increase CLS. Learn more 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 more CLS

Has a `<meta name="viewport">` tag with `width` or `initial-scale`

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. TBT

Avoids `unload` event listeners

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. Learn more

Captured at Aug 4, 2022, 6:59 PM GMT+2	Emulated Desktop with Lighthouse 9.6.1	Single page load
Initial page load	Custom throttling	Using Chromium 103.0.0.0 with devtools

Generated by **Lighthouse** 9.6.1 | File an issue