http://localhost:3000/dashboard

74

Performance

Metrics

First Contentful Paint	1.6 s	Time to Interactive	1.9 s
Speed Index	1.6 s	Total Blocking Time	270 ms
Largest Contentful Paint	1.7 s	Cumulative Layout Shift	0

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

View Original Trace

Opportunities — These suggestions can help your page load faster. They don't directly affect the Performance score.

Opportunity	Estimated Savings
▲ Remove unused JavaScript	Error!

Remove unused JavaScript to reduce bytes consumed by network activity. Learn more.

If you are not server-side rendering, split your JavaScript bundles with `React.lazy()`. Otherwise, code-split using a third-party library such as loadable-components.

▲ Remove duplicate modules in JavaScript bundles	Error!

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

▲ Avoid serving legacy JavaScript to modern browsers	Error!

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

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

▲ Serve static assets with an efficient cache policy — 3 resources found

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

☐ ~~Show 3rd-party resources~~ (0)

URL	Cache TTL	Transfer Size
…js/0.chunk.js (localhost)	None	647 KiB
…js/main.chunk.js (localhost)	None	21 KiB
…js/bundle.js (localhost)	None	14 KiB

Avoid chaining critical requests — 9 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.

Maximum critical path latency: **1,130 ms**

Initial Navigation

/dashboard (localhost)

- …css/bootstrap.min.css (maxcdn.bootstrapcdn.com) **- 60 ms, 19.37 KiB**
- /css?family=PT+Serif|Open+Sans:300,400,600,700,800 (fonts.googleapis.com) **- 120 ms, 0.92 KiB**
- /css2?family=Roboto:wght@300;400;500&display=swap (fonts.googleapis.com)
  - …v20/KFOlCnqEu….woff2 (fonts.gstatic.com) **- 30 ms, 11.14 KiB**
- …js/bootstrap.min.js (maxcdn.bootstrapcdn.com) **- 60 ms, 9.86 KiB**
- /0075a9b63d.js (kit.fontawesome.com) **- 440 ms, 3.74 KiB**
- …js/bundle.js (localhost) **- 30 ms, 14.14 KiB**
- …js/0.chunk.js (localhost) **- 390 ms, 647.2 KiB**
- …js/main.chunk.js (localhost) **- 280 ms, 20.95 KiB**
- …webfonts/free-fa-solid-900.woff2 (ka-f.fontawesome.com) **- 50 ms, 142.2 KiB**

User Timing marks and measures — 108 user timings

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. Learn more.

Use the React DevTools Profiler, which makes use of the Profiler API, to measure the rendering performance of your components. Learn more.

Name	Type	Start Time	Duration
⚛ (React Tree Reconciliation: Completed Root)	Measure	856.84 ms	39.54 ms
⚛ Provider [mount]	Measure	860.83 ms	35.15 ms
⚛ App [mount]	Measure	863.61 ms	32.33 ms
⚛ Routes [mount]	Measure	864.1 ms	31.8 ms
⚛ BrowserRouter [mount]	Measure	865.57 ms	30.29 ms
⚛ Router [mount]	Measure	871.13 ms	24.69 ms
⚛ Switch [mount]	Measure	872.08 ms	23.68 ms
⚛ AuthenticatedRoute [mount]	Measure	875.62 ms	20.1 ms

Name	Type	Start Time	Duration
⚛ Dashboard [mount]	Measure	877.52 ms	18.11 ms
⚛ LogOutButton [mount]	Measure	882.24 ms	7.46 ms
⚛ withRouter(LinkContainer) [mount]	Measure	890.26 ms	4.54 ms
⚛ LinkContainer [mount]	Measure	890.94 ms	3.81 ms
⚛ Route [mount]	Measure	891.9 ms	2.8 ms
⚛ Route [mount]	Measure	891.9 ms	3.79 ms
⚛ ListGroupItem [mount]	Measure	892.54 ms	2.02 ms
⚛ (Committing Changes)	Measure	897.51 ms	9.46 ms
⚛ (Committing Snapshot Effects: 0 Total)	Measure	897.7 ms	2.73 ms
⚛ (Committing Host Effects: 6 Total)	Measure	900.55 ms	2.29 ms
⚛ (Calling Lifecycle Methods: 6 Total)	Measure	903.13 ms	3.73 ms
⚛ Route.componentDidMount	Measure	905.17 ms	0.38 ms
⚛ Router.componentDidMount	Measure	906.43 ms	0.11 ms
⚛ BrowserRouter.componentDidMount	Measure	906.6 ms	0.1 ms
⚛ (React Tree Reconciliation: Completed Root)	Measure	999.13 ms	12.8 ms
⚛ Dashboard [update]	Measure	1,000.47 ms	11.32 ms
⚛ LogOutButton [update]	Measure	1,003.16 ms	5.54 ms
⚛ withRouter(LinkContainer) [update]	Measure	1,009.08 ms	2.52 ms
⚛ LinkContainer [update]	Measure	1,009.37 ms	2.19 ms
⚛ Route [update]	Measure	1,010.11 ms	1.41 ms
⚛ ListGroupItem [update]	Measure	1,010.92 ms	0.55 ms
⚛ (Committing Changes)	Measure	1,012 ms	2.06 ms
⚛ (Committing Snapshot Effects: 0 Total)	Measure	1,012.04 ms	0.59 ms
⚛ (Committing Host Effects: 3 Total)	Measure	1,012.67 ms	0.67 ms
⚛ (Calling Lifecycle Methods: 3 Total)	Measure	1,013.38 ms	0.64 ms
⚛ Route.componentDidUpdate	Measure	1,013.79 ms	0.13 ms
⚛ (React Tree Reconciliation: Completed Root)	Measure	1,014.31 ms	2.83 ms
⚛ Dashboard [update]	Measure	1,015.03 ms	2 ms
⚛ LogOutButton [update]	Measure	1,015.53 ms	0.26 ms
⚛ withRouter(LinkContainer) [update]	Measure	1,015.93 ms	0.95 ms
⚛ LinkContainer [update]	Measure	1,016.13 ms	0.72 ms

Name	Type	Start Time	Duration
⚛ Route [update]	Measure	1,016.25 ms	0.56 ms
⚛ ListGroupItem [update]	Measure	1,016.47 ms	0.29 ms
⚛ (Committing Changes)	Measure	1,017.18 ms	1.67 ms
⚛ (Committing Snapshot Effects: 0 Total)	Measure	1,017.23 ms	0.6 ms
⚛ (Committing Host Effects: 3 Total)	Measure	1,017.87 ms	0.41 ms
⚛ (Calling Lifecycle Methods: 3 Total)	Measure	1,018.32 ms	0.5 ms
⚛ Route.componentDidUpdate	Measure	1,018.67 ms	0.05 ms
⚛ (React Tree Reconciliation)	Mark	856.88 ms	
⚛ Provider [mount] (#4)	Mark	860.85 ms	
⚛ App [mount] (#8)	Mark	863.64 ms	
⚛ Routes [mount] (#10)	Mark	864.13 ms	
⚛ BrowserRouter [mount] (#12)	Mark	865.69 ms	
⚛ Router [mount] (#14)	Mark	871.17 ms	
⚛ Switch [mount] (#20)	Mark	872.11 ms	
⚛ AuthenticatedRoute [mount] (#24)	Mark	875.65 ms	
⚛ Route [mount] (#26)	Mark	875.97 ms	
⚛ Dashboard [mount] (#32)	Mark	877.53 ms	
⚛ LogOutButton [mount] (#36)	Mark	882.27 ms	
⚛ withRouter(LinkContainer) [mount] (#48)	Mark	890.28 ms	
⚛ LinkContainer [mount] (#53)	Mark	890.95 ms	
⚛ Route [mount] (#55)	Mark	891.91 ms	
⚛ ListGroupItem [mount] (#61)	Mark	892.55 ms	
⚛ (Committing Changes)	Mark	897.53 ms	
⚛ (Committing Snapshot Effects)	Mark	897.71 ms	
⚛ (Committing Host Effects)	Mark	900.59 ms	
⚛ (Calling Lifecycle Methods)	Mark	903.15 ms	
⚛ Route.componentDidMount (#55)	Mark	905.21 ms	
⚛ Router.componentDidMount (#14)	Mark	906.45 ms	
⚛ BrowserRouter.componentDidMount (#12)	Mark	906.6 ms	
⚛ (React Tree Reconciliation)	Mark	999.17 ms	
⚛ Provider [update] (#4)	Mark	999.61 ms	

Name	Type	Start Time	Duration
⚛ App [update] (#8)	Mark	999.83 ms	
⚛ Routes [update] (#10)	Mark	999.91 ms	
⚛ BrowserRouter [update] (#12)	Mark	999.99 ms	
⚛ Router [update] (#14)	Mark	1,000.06 ms	
⚛ Switch [update] (#20)	Mark	1,000.18 ms	
⚛ AuthenticatedRoute [update] (#24)	Mark	1,000.29 ms	
⚛ Route [update] (#26)	Mark	1,000.36 ms	
⚛ Dashboard [update] (#32)	Mark	1,000.48 ms	
⚛ LogOutButton [update] (#36)	Mark	1,003.2 ms	
⚛ withRouter(LinkContainer) [update] (#48)	Mark	1,009.12 ms	
⚛ LinkContainer [update] (#53)	Mark	1,009.4 ms	
⚛ Route [update] (#55)	Mark	1,010.13 ms	
⚛ ListGroupItem [update] (#61)	Mark	1,010.93 ms	
⚛ (Committing Changes)	Mark	1,012.01 ms	
⚛ (Committing Snapshot Effects)	Mark	1,012.05 ms	
⚛ (Committing Host Effects)	Mark	1,012.68 ms	
⚛ (Calling Lifecycle Methods)	Mark	1,013.39 ms	
⚛ Route.componentDidUpdate (#55)	Mark	1,013.8 ms	
⚛ (React Tree Reconciliation)	Mark	1,014.31 ms	
⚛ Provider [update] (#4)	Mark	1,014.41 ms	
⚛ App [update] (#8)	Mark	1,014.49 ms	
⚛ Routes [update] (#10)	Mark	1,014.58 ms	
⚛ BrowserRouter [update] (#12)	Mark	1,014.64 ms	
⚛ Router [update] (#14)	Mark	1,014.71 ms	
⚛ Switch [update] (#20)	Mark	1,014.8 ms	
⚛ AuthenticatedRoute [update] (#24)	Mark	1,014.88 ms	
⚛ Route [update] (#26)	Mark	1,014.94 ms	
⚛ Dashboard [update] (#32)	Mark	1,015.03 ms	
⚛ LogOutButton [update] (#36)	Mark	1,015.55 ms	
⚛ withRouter(LinkContainer) [update] (#48)	Mark	1,015.95 ms	
⚛ LinkContainer [update] (#53)	Mark	1,016.14 ms	

Name	Type	Start Time	Duration
⚛ Route [update] (#55)	Mark	1,016.26 ms	
⚛ ListGroupItem [update] (#61)	Mark	1,016.48 ms	
⚛ (Committing Changes)	Mark	1,017.21 ms	
⚛ (Committing Snapshot Effects)	Mark	1,017.24 ms	
⚛ (Committing Host Effects)	Mark	1,017.88 ms	
⚛ (Calling Lifecycle Methods)	Mark	1,018.33 ms	
⚛ Route.componentDidUpdate (#55)	Mark	1,018.68 ms	

Keep request counts low and transfer sizes small — 18 requests • 897 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	18	896.7 KiB
Script	5	695.9 KiB
Font	2	153.3 KiB
Other	7	25.1 KiB
Stylesheet	3	21.4 KiB
Document	1	1 KiB
Image	0	0 KiB
Media	0	0 KiB
Third-party	10	206.3 KiB

Largest Contentful Paint element — 1 element found

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

Element
h1

Avoid large layout shifts — 1 element found

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

Element	CLS Contribution
i.fas.fa-sign-out-alt	0

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

Show 3rd-party resources (1)

URL	Start Time	Duration
/0075a9b63d.js (kit.fontawesome.com)	1,486 ms	412 ms

Passed audits (25)

Eliminate render-blocking resources — Potential savings of 80 ms

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. Learn more.

Show 3rd-party resources (4)

URL	Transfer Size	Potential Savings
…css/bootstrap.min.css (maxcdn.bootstrapcdn.com)	19.4 KiB	340 ms
/css?family=PT+Serif|Open+Sans:300,400,600,700,800 (fonts.googleapis.com)	0.9 KiB	260 ms
/css2?family=Roboto:wght@300;400;500&display=swap (fonts.googleapis.com)	1.1 KiB	260 ms
…js/bootstrap.min.js (maxcdn.bootstrapcdn.com)	9.9 KiB	260 ms

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.

If your build system minifies CSS files automatically, ensure that you are deploying the production build of your application. You can check this with the React Developer Tools extension. Learn more.

Minify JavaScript

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

If your build system minifies JS files automatically, ensure that you are deploying the production build of your application. You can check this with the React Developer Tools extension. Learn more.

Remove unused CSS — Potential savings of 31 KiB

Remove dead rules from stylesheets and defer the loading of CSS not used for above-the-fold content to reduce unnecessary bytes consumed by network activity. Learn more.

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
…css/bootstrap.min.css (maxcdn.bootstrapcdn.com)	19.4 KiB	19.1 KiB

URL	Transfer Size	Potential Savings
`/*! * Font Awesome Free 5.15.1 by @fontawesome - https://fontawesome.com * License - https://fonta...`	12 KiB	12 KiB

Efficiently encode images

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

Serve images in next-gen formats

Image formats like JPEG 2000, JPEG XR, and WebP 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.

Preconnect to required origins

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

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.

☐ ~~Show 3rd-party resources~~ (0)

URL	Time Spent
/dashboard (localhost)	0 ms

Avoid multiple page redirects

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

If you are using React Router, minimize usage of the `<Redirect>` component for route navigations.

Preload key requests

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

Use HTTP/2

HTTP/2 offers many benefits over HTTP/1.1, including binary headers, multiplexing, and server push. 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

Avoids enormous network payloads — Total size was 897 KiB

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

☑ Show 3rd-party resources (6)

URL	Transfer Size
…js/0.chunk.js (localhost)	647.2 KiB
…webfonts/free-fa-solid-900.woff2 (ka-f.fontawesome.com)	142.2 KiB
…js/main.chunk.js (localhost)	21 KiB
…css/bootstrap.min.css (maxcdn.bootstrapcdn.com)	19.4 KiB
…js/bundle.js (localhost)	14.1 KiB
…css/free.min.css (ka-f.fontawesome.com)	12.5 KiB
…v20/KFOlCnqEu….woff2 (fonts.gstatic.com)	11.1 KiB
…js/bootstrap.min.js (maxcdn.bootstrapcdn.com)	9.9 KiB
/logo192.png (localhost)	5.5 KiB
…css/free-v4-shims.min.css (ka-f.fontawesome.com)	4.2 KiB

Avoids an excessive DOM size — 18 elements

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

Consider using a “windowing” library like `react-window` to minimize the number of DOM nodes created if you are rendering many repeated elements on the page. Learn more. Also, minimize unnecessary re-renders using `shouldComponentUpdate`, `PureComponent`, or `React.memo` and skip effects only until certain dependencies have changed if you are using the `Effect` hook to improve runtime performance.

Statistic	Element	Value
Total DOM Elements		18
Maximum DOM Depth	`<b>`	8
Maximum Child Elements	`<body>`	9

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 more.

☐ ~~Show 3rd-party resources~~ (0)

URL	Total CPU Time	Script Evaluation	Script Parse
`Unattributable`	218 ms	14 ms	0 ms
…js/main.chunk.js (localhost)	214 ms	206 ms	8 ms
/dashboard (localhost)	210 ms	5 ms	2 ms
…js/0.chunk.js (localhost)	137 ms	46 ms	84 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 more

Category	Time Spent
Script Evaluation	304 ms
Other	245 ms
Parse HTML & CSS	122 ms
Script Parsing & Compilation	99 ms
Style & Layout	60 ms
Garbage Collection	11 ms
Rendering	5 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.

Minimize third-party usage — Third-party code blocked the main thread for 0 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 more.

Show 3rd-party resources (0)

Third-Party	Transfer Size	Main-Thread Blocking Time
FontAwesome CDN	164 KiB	0 ms
…webfonts/free-fa-solid-900.woff2 (ka-f.fontawesome.com)	142 KiB	0 ms
…css/free.min.css (ka-f.fontawesome.com)	13 KiB	0 ms
Other resources	9 KiB	0 ms
Bootstrap CDN	29 KiB	0 ms
…css/bootstrap.min.css (maxcdn.bootstrapcdn.com)	19 KiB	0 ms
…js/bootstrap.min.js (maxcdn.bootstrapcdn.com)	10 KiB	0 ms
Google Fonts	13 KiB	0 ms
…v20/KFOlCnqEu….woff2 (fonts.gstatic.com)	11 KiB	0 ms

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

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

Runtime Settings

URL	http://localhost:3000/dashboard
Fetch Time	Jan 3, 2021, 10:26 PM GMT-3
Device	Emulated Desktop
Network throttling	40 ms TCP RTT, 10,240 Kbps throughput (Simulated)
CPU throttling	1x slowdown (Simulated)
Channel	devtools
User agent (host)	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
User agent (network)	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4143.7 Safari/537.36 Chrome-Lighthouse
CPU/Memory Power	587

Generated by **Lighthouse** 6.4.0 | File an issue