http://localhost:3000/dashboard

74

Performance

Metrics

First Contentful Paint	1.5 s	Time to Interactive	1.9 s
Speed Index	1.5 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
Minify JavaScript	0.24 s

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.

Show 3rd-party resources (0)

URL	Transfer Size	Potential Savings
…js/0.chunk.js (localhost)	647.2 KiB	246.1 KiB
…js/main.chunk.js (localhost)	41.6 KiB	10.8 KiB
…js/bundle.js (localhost)	14.1 KiB	6.7 KiB

Remove unused JavaScript	0.2 s

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.

Show 3rd-party resources (0)

URL	Transfer Size	Potential Savings
…js/0.chunk.js (localhost)	647.2 KiB	216.4 KiB
…front-end/node_modules/react-dom/cjs/react-dom.development.js	178.2 KiB	75.9 KiB
…front-end/node_modules/lodash/lodash.js	102.5 KiB	29.1 KiB
…front-end/node_modules/react-error-overlay/lib/index.js	73.5 KiB	11.5 KiB
…front-end/node_modules/history/esm/history.js	5.9 KiB	4 KiB
…front-end/node_modules/react-redux/es/components/connectAdvanced.js	4.2 KiB	3.8 KiB

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	42 KiB
…js/bundle.js (localhost)	None	14 KiB

Avoid chaining critical requests — 10 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) **- 40 ms, 35.28 KiB**
  - /css?family=PT+Serif|Open+Sans:300,400,600,700,800 (fonts.googleapis.com) **- 80 ms, 1.3 KiB**
  - /css2?family=Roboto:wght@300;400;500&display=swap (fonts.googleapis.com)
    - …v20/KFOlCnqEu….woff2 (fonts.gstatic.com) **- 30 ms, 11.14 KiB**
    - …v20/KFOmCnqEu….woff2 (fonts.gstatic.com) **- 90 ms, 10.78 KiB**
  - …js/bootstrap.min.js (maxcdn.bootstrapcdn.com) **- 40 ms, 9.9 KiB**
  - /0075a9b63d.js (kit.fontawesome.com) **- 430 ms, 3.74 KiB**
  - …js/bundle.js (localhost) **- 20 ms, 14.14 KiB**
  - …js/0.chunk.js (localhost) **- 340 ms, 647.2 KiB**
  - …js/main.chunk.js (localhost) **- 30 ms, 41.59 KiB**
  - …webfonts/free-fa-solid-900.woff2 (ka-f.fontawesome.com) **- 40 ms, 82.34 KiB**

User Timing marks and measures — 126 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	839.08 ms	36.34 ms
⚛ Provider [mount]	Measure	843.18 ms	31.85 ms
⚛ App [mount]	Measure	845.44 ms	29.54 ms
⚛ Routes [mount]	Measure	845.89 ms	29.06 ms
⚛ BrowserRouter [mount]	Measure	847.33 ms	27.58 ms
⚛ Router [mount]	Measure	850.45 ms	24.43 ms
⚛ Switch [mount]	Measure	851.25 ms	23.58 ms
⚛ AuthenticatedRoute [mount]	Measure	855.42 ms	19.37 ms
⚛ Dashboard [mount]	Measure	857.32 ms	17.38 ms
⚛ LogOutButton [mount]	Measure	861.82 ms	5.95 ms
⚛ withRouter(LinkContainer) [mount]	Measure	868.33 ms	5.49 ms
⚛ LinkContainer [mount]	Measure	868.98 ms	4.8 ms
⚛ Route [mount]	Measure	870.59 ms	3.13 ms
⚛ Route [mount]	Measure	870.59 ms	4.17 ms
⚛ ListGroupItem [mount]	Measure	871.34 ms	2.24 ms
⚛ (Committing Changes)	Measure	876.47 ms	6.63 ms
⚛ (Committing Snapshot Effects: 0 Total)	Measure	876.66 ms	2.12 ms
⚛ (Committing Host Effects: 6 Total)	Measure	878.88 ms	1.5 ms
⚛ (Calling Lifecycle Methods: 6 Total)	Measure	880.59 ms	2.37 ms
⚛ Route.componentDidMount	Measure	881.41 ms	0.28 ms
⚛ Router.componentDidMount	Measure	882.53 ms	0.13 ms
⚛ BrowserRouter.componentDidMount	Measure	882.7 ms	0.1 ms
⚛ (React Tree Reconciliation: Completed Root)	Measure	994.2 ms	13.12 ms
⚛ Dashboard [update]	Measure	995.47 ms	11.73 ms
⚛ LogOutButton [update]	Measure	998.41 ms	0.88 ms
⚛ withRouter(LinkContainer) [update]	Measure	999.54 ms	2.03 ms
⚛ LinkContainer [update]	Measure	999.73 ms	1.8 ms

Name	Type	Start Time	Duration
⚛ Route [update]	Measure	1,000.3 ms	1.2 ms
⚛ ListGroupItem [update]	Measure	1,000.96 ms	0.49 ms
⚛ withRouter(LinkContainer) [mount]	Measure	1,001.65 ms	5.32 ms
⚛ LinkContainer [mount]	Measure	1,001.82 ms	5.11 ms
⚛ Route [mount]	Measure	1,002.14 ms	4.75 ms
⚛ ListGroupItem [mount]	Measure	1,002.57 ms	4.16 ms
⚛ (Committing Changes)	Measure	1,007.4 ms	2.92 ms
⚛ (Committing Snapshot Effects: 0 Total)	Measure	1,007.46 ms	0.64 ms
⚛ (Committing Host Effects: 5 Total)	Measure	1,008.15 ms	1.44 ms
⚛ (Calling Lifecycle Methods: 4 Total)	Measure	1,009.63 ms	0.65 ms
⚛ Route.componentDidUpdate	Measure	1,009.98 ms	0.13 ms
⚛ Route.componentDidMount	Measure	1,010.14 ms	0.05 ms
⚛ (React Tree Reconciliation: Completed Root)	Measure	1,010.71 ms	3.93 ms
⚛ Dashboard [update]	Measure	1,011.55 ms	2.98 ms
⚛ LogOutButton [update]	Measure	1,012.13 ms	0.22 ms
⚛ withRouter(LinkContainer) [update]	Measure	1,012.48 ms	0.84 ms
⚛ LinkContainer [update]	Measure	1,012.65 ms	0.63 ms
⚛ Route [update]	Measure	1,012.75 ms	0.5 ms
⚛ ListGroupItem [update]	Measure	1,012.95 ms	0.26 ms
⚛ withRouter(LinkContainer) [update]	Measure	1,013.39 ms	1.03 ms
⚛ LinkContainer [update]	Measure	1,013.53 ms	0.85 ms
⚛ Route [update]	Measure	1,013.63 ms	0.72 ms
⚛ ListGroupItem [update]	Measure	1,014 ms	0.31 ms
⚛ (Committing Changes)	Measure	1,014.67 ms	1.38 ms
⚛ (Committing Snapshot Effects: 0 Total)	Measure	1,014.71 ms	0.44 ms
⚛ (Committing Host Effects: 4 Total)	Measure	1,015.17 ms	0.35 ms
⚛ (Calling Lifecycle Methods: 4 Total)	Measure	1,015.56 ms	0.47 ms
⚛ Route.componentDidUpdate	Measure	1,015.88 ms	0.04 ms
⚛ (React Tree Reconciliation)	Mark	839.13 ms	
⚛ Provider [mount] (#4)	Mark	843.21 ms	
⚛ App [mount] (#8)	Mark	845.46 ms	

Name	Type	Start Time	Duration
⚛ Routes [mount] (#10)	Mark	845.9 ms	
⚛ BrowserRouter [mount] (#12)	Mark	847.36 ms	
⚛ Router [mount] (#14)	Mark	850.5 ms	
⚛ Switch [mount] (#20)	Mark	851.28 ms	
⚛ AuthenticatedRoute [mount] (#24)	Mark	855.46 ms	
⚛ Route [mount] (#26)	Mark	855.81 ms	
⚛ Dashboard [mount] (#32)	Mark	857.34 ms	
⚛ LogOutButton [mount] (#36)	Mark	861.84 ms	
⚛ withRouter(LinkContainer) [mount] (#48)	Mark	868.36 ms	
⚛ LinkContainer [mount] (#53)	Mark	868.99 ms	
⚛ Route [mount] (#55)	Mark	870.63 ms	
⚛ ListGroupItem [mount] (#61)	Mark	871.37 ms	
⚛ (Committing Changes)	Mark	876.5 ms	
⚛ (Committing Snapshot Effects)	Mark	876.68 ms	
⚛ (Committing Host Effects)	Mark	878.89 ms	
⚛ (Calling Lifecycle Methods)	Mark	880.61 ms	
⚛ Route.componentDidMount (#55)	Mark	881.43 ms	
⚛ Router.componentDidMount (#14)	Mark	882.56 ms	
⚛ BrowserRouter.componentDidMount (#12)	Mark	882.71 ms	
⚛ (React Tree Reconciliation)	Mark	994.24 ms	
⚛ Provider [update] (#4)	Mark	994.65 ms	
⚛ App [update] (#8)	Mark	994.87 ms	
⚛ Routes [update] (#10)	Mark	994.94 ms	
⚛ BrowserRouter [update] (#12)	Mark	995.03 ms	
⚛ Router [update] (#14)	Mark	995.09 ms	
⚛ Switch [update] (#20)	Mark	995.21 ms	
⚛ AuthenticatedRoute [update] (#24)	Mark	995.31 ms	
⚛ Route [update] (#26)	Mark	995.37 ms	
⚛ Dashboard [update] (#32)	Mark	995.48 ms	
⚛ LogOutButton [update] (#36)	Mark	998.46 ms	
⚛ withRouter(LinkContainer) [update] (#48)	Mark	999.55 ms	

Name	Type	Start Time	Duration
⚛ LinkContainer [update] (#53)	Mark	999.75 ms	
⚛ Route [update] (#55)	Mark	1,000.32 ms	
⚛ ListGroupItem [update] (#61)	Mark	1,000.98 ms	
⚛ withRouter(LinkContainer) [mount] (#141)	Mark	1,001.67 ms	
⚛ LinkContainer [mount] (#145)	Mark	1,001.84 ms	
⚛ Route [mount] (#147)	Mark	1,002.15 ms	
⚛ ListGroupItem [mount] (#153)	Mark	1,002.59 ms	
⚛ (Committing Changes)	Mark	1,007.43 ms	
⚛ (Committing Snapshot Effects)	Mark	1,007.49 ms	
⚛ (Committing Host Effects)	Mark	1,008.16 ms	
⚛ (Calling Lifecycle Methods)	Mark	1,009.65 ms	
⚛ Route.componentDidUpdate (#55)	Mark	1,010 ms	
⚛ Route.componentDidMount (#147)	Mark	1,010.15 ms	
⚛ (React Tree Reconciliation)	Mark	1,010.72 ms	
⚛ Provider [update] (#4)	Mark	1,010.82 ms	
⚛ App [update] (#8)	Mark	1,010.89 ms	
⚛ Routes [update] (#10)	Mark	1,010.96 ms	
⚛ BrowserRouter [update] (#12)	Mark	1,011.01 ms	
⚛ Router [update] (#14)	Mark	1,011.07 ms	
⚛ Switch [update] (#20)	Mark	1,011.16 ms	
⚛ AuthenticatedRoute [update] (#24)	Mark	1,011.23 ms	
⚛ Route [update] (#26)	Mark	1,011.48 ms	
⚛ Dashboard [update] (#32)	Mark	1,011.56 ms	
⚛ LogOutButton [update] (#36)	Mark	1,012.15 ms	
⚛ withRouter(LinkContainer) [update] (#48)	Mark	1,012.5 ms	
⚛ LinkContainer [update] (#53)	Mark	1,012.66 ms	
⚛ Route [update] (#55)	Mark	1,012.77 ms	
⚛ ListGroupItem [update] (#61)	Mark	1,012.96 ms	
⚛ withRouter(LinkContainer) [update] (#141)	Mark	1,013.4 ms	
⚛ LinkContainer [update] (#145)	Mark	1,013.55 ms	
⚛ Route [update] (#147)	Mark	1,013.64 ms	

Name	Type	Start Time	Duration
⚛ ListGroupItem [update] (#153)	Mark	1,014.02 ms	
⚛ (Committing Changes)	Mark	1,014.68 ms	
⚛ (Committing Snapshot Effects)	Mark	1,014.72 ms	
⚛ (Committing Host Effects)	Mark	1,015.19 ms	
⚛ (Calling Lifecycle Methods)	Mark	1,015.57 ms	
⚛ Route.componentDidUpdate (#55)	Mark	1,015.9 ms	

Keep request counts low and transfer sizes small — 19 requests • 884 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	19	884.4 KiB
Script	5	716.6 KiB
Font	3	104.3 KiB
Stylesheet	3	37.3 KiB
Other	7	25.2 KiB
Document	1	1 KiB
Image	0	0 KiB
Media	0	0 KiB
Third-party	11	173.6 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)	1,495 ms	377 ms

Passed audits (26)

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)	35.3 KiB	390 ms
/css?family=PT+Serif|Open+Sans:300,400,600,700,800 (fonts.googleapis.com)	1.3 KiB	260 ms
/css2?family=Roboto:wght@300;400;500&display=swap (fonts.googleapis.com)	0.8 KiB	260 ms
…js/bootstrap.min.js (maxcdn.bootstrapcdn.com)	9.9 KiB	270 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.

Remove unused CSS — Potential savings of 47 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)	35.3 KiB	34.7 KiB
`/*! * 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 — Potential savings of 100 ms

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

URL	Potential Savings
https://ka-f.fontawesome.com	100 ms

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

Remove duplicate modules in JavaScript bundles

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

Avoid serving legacy JavaScript to modern browsers — Potential savings of 6 KiB

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

Show 3rd-party resources (0)

URL	Potential Savings
…js/0.chunk.js (localhost)	6 KiB
0.chunk.js:67645	`@babel/plugin-transform-classes`
0.chunk.js:67645	`Object.getOwnPropertyNames`
0.chunk.js:67645	`Array.from`
0.chunk.js:5833	`Array.isArray`
0.chunk.js:5881	`Object.keys`
0.chunk.js:5955	`Object.entries`
0.chunk.js:5975	`Object.values`

Avoids enormous network payloads — Total size was 884 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)	82.3 KiB
…js/main.chunk.js (localhost)	41.6 KiB
…css/bootstrap.min.css (maxcdn.bootstrapcdn.com)	35.3 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
…v20/KFOmCnqEu….woff2 (fonts.gstatic.com)	10.8 KiB
…js/bootstrap.min.js (maxcdn.bootstrapcdn.com)	9.9 KiB
/logo192.png (localhost)	5.5 KiB

Avoids an excessive DOM size — 21 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		21
Maximum DOM Depth	`<b>`	8
Maximum Child Elements	`<body>`	9

JavaScript execution time — 0.3 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
/dashboard (localhost)	221 ms	3 ms	1 ms
Unattributable	211 ms	6 ms	0 ms
…js/main.chunk.js (localhost)	199 ms	191 ms	8 ms
…js/0.chunk.js (localhost)	129 ms	52 ms	69 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	293 ms
Other	252 ms
Parse HTML & CSS	123 ms
Script Parsing & Compilation	84 ms
Style & Layout	67 ms
Garbage Collection	9 ms
Rendering	6 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	104 KiB	0 ms
…webfonts/free-fa-solid-900.woff2 (ka-f.fontawesome.com)	82 KiB	0 ms
…css/free.min.css (ka-f.fontawesome.com)	13 KiB	0 ms
Other resources	10 KiB	0 ms
Bootstrap CDN	45 KiB	0 ms
…css/bootstrap.min.css (maxcdn.bootstrapcdn.com)	35 KiB	0 ms
…js/bootstrap.min.js (maxcdn.bootstrapcdn.com)	10 KiB	0 ms
Google Fonts	24 KiB	0 ms

Third-Party	Transfer Size	Main-Thread Blocking Time
…v20/KFOlCnqEu….woff2 (fonts.gstatic.com)	11 KiB	0 ms
…v20/KFOmCnqEu….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, 10:38 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	589

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