http://localhost:3000/

61 Performance

69 Accessibility

100 Best Practices

83 SEO

61

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

5.5 s

▲ Largest Contentful Paint

22.1 s

Total Blocking Time

60 ms

Cumulative Layout Shift

0

Speed Index

5.5 s

View Treemap
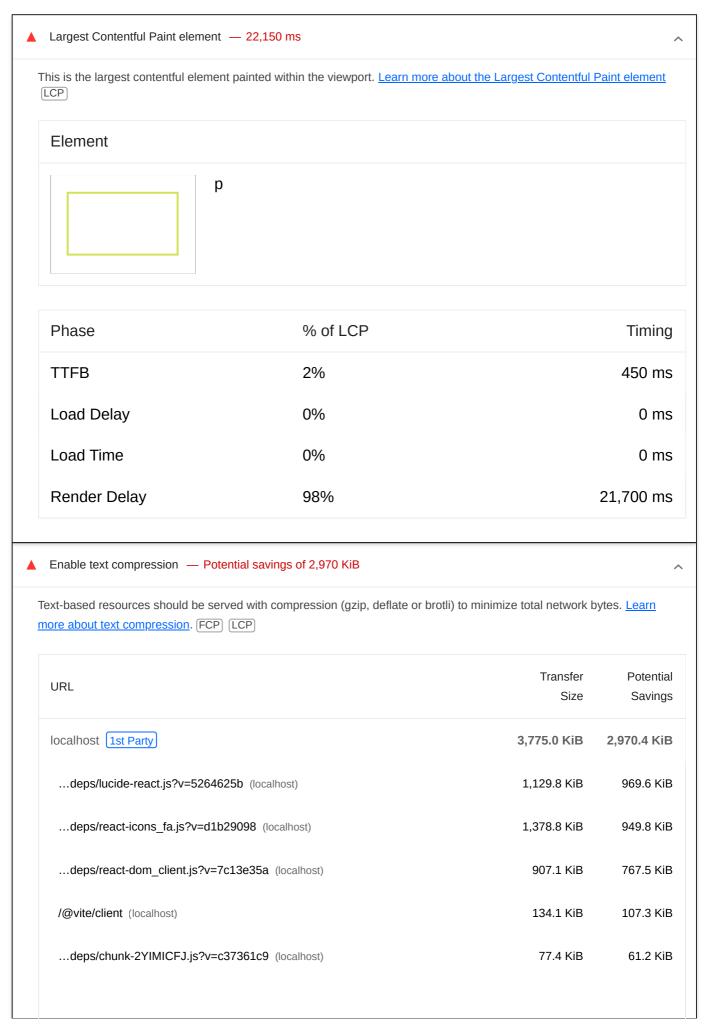
Show audits relevant to: All   FCP   LCP   TBT

DIAGNOSTICS

🔺 Largest Contentful Paint element — 22,150 ms ⌃

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

Element

▢	p

Phase	% of LCP	Timing
TTFB	2%	450 ms
Load Delay	0%	0 ms
Load Time	0%	0 ms
Render Delay	98%	21,700 ms

🔺 Enable text compression — Potential savings of 2,970 KiB ⌃

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

URL	Transfer Size	Potential Savings
localhost 1st Party	3,775.0 KiB	2,970.4 KiB
…deps/lucide-react.js?v=5264625b (localhost)	1,129.8 KiB	969.6 KiB
…deps/react-icons_fa.js?v=d1b29098 (localhost)	1,378.8 KiB	949.8 KiB
…deps/react-dom_client.js?v=7c13e35a (localhost)	907.1 KiB	767.5 KiB
/@vite/client (localhost)	134.1 KiB	107.3 KiB
…deps/chunk-2YIMICFJ.js?v=c37361c9 (localhost)	77.4 KiB	61.2 KiB

URL	Transfer Size	Potential Savings
/@react-refresh (localhost)	74.2 KiB	60.3 KiB
…deps/react_jsx-dev-runtime.js?v=81295adb (localhost)	36.1 KiB	28.3 KiB
/src/Frontend.jsx?t=173… (localhost)	10.6 KiB	8.4 KiB
/src/Links.jsx (localhost)	8.5 KiB	6.3 KiB
/src/main.jsx?t=173… (localhost)	5.5 KiB	3.5 KiB
/src/App.jsx?t=173… (localhost)	5.1 KiB	3.3 KiB
…client/env.mjs (localhost)	3.9 KiB	2.7 KiB
/src/About.jsx (localhost)	3.9 KiB	2.2 KiB

▲ Minify JavaScript — Potential savings of 810 KiB ⌃

Minifying JavaScript files can reduce payload sizes and script parse time. Learn how to minify JavaScript. FCP LCP

URL	Transfer Size	Potential Savings
localhost 1st Party	**2,392.3 KiB**	**784.9 KiB**
…deps/react-dom_client.js?v=7c13e35a (localhost)	907.1 KiB	332.0 KiB
…deps/lucide-react.js?v=5264625b (localhost)	1,129.8 KiB	206.6 KiB
/@vite/client (localhost)	134.1 KiB	116.7 KiB
/@react-refresh (localhost)	74.2 KiB	64.0 KiB
…deps/chunk-2YIMICFJ.js?v=c37361c9 (localhost)	77.4 KiB	31.5 KiB
…deps/react_jsx-dev-runtime.js?v=81295adb (localhost)	36.1 KiB	14.0 KiB
/src/Frontend.jsx?t=173… (localhost)	10.6 KiB	6.6 KiB
/src/Links.jsx (localhost)	8.5 KiB	5.1 KiB
…client/env.mjs (localhost)	3.9 KiB	3.5 KiB

URL	Transfer Size	Potential Savings
/src/App.jsx?t=173… (localhost)	5.1 KiB	2.5 KiB
/src/main.jsx?t=173… (localhost)	5.5 KiB	2.4 KiB
Unattributable	**60.1 KiB**	**25.3 KiB**
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js	60.1 KiB	25.3 KiB

⚠️ Eliminate render-blocking resources — Potential savings of 750 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 (2)

URL	Transfer Size	Potential Savings
JSDelivr CDN Cdn	**49.2 KiB**	**1,250 ms**
…css/bootstrap.min.css (cdn.jsdelivr.net)	26.9 KiB	1,100 ms
…js/bootstrap.bundle.min.js (cdn.jsdelivr.net)	22.3 KiB	150 ms
localhost 1st Party	**0.3 KiB**	**150 ms**
/src/index.css (localhost)	0.3 KiB	150 ms

⚠️ Reduce unused CSS — Potential savings of 26 KiB ⌃

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

URL	Transfer Size	Potential Savings
JSDelivr CDN Cdn	**26.8 KiB**	**26.0 KiB**
…css/bootstrap.min.css (cdn.jsdelivr.net)	26.8 KiB	26.0 KiB

⚠ **Reduce unused JavaScript** — Potential savings of 1,921 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
localhost [1st Party]	**3,529.2 KiB**	**1,848.9 KiB**
…deps/react-icons_fa.js?v=d1b29098 (localhost)	1,378.8 KiB	1,340.8 KiB
…fa/index.mjs	1,350.3 KiB	1,340.4 KiB
…deps/react-dom_client.js?v=7c13e35a (localhost)	907.1 KiB	418.5 KiB
…react-dom/cjs/react-dom.development.js	887.9 KiB	414.6 KiB
…scheduler/cjs/scheduler.development.js	16.5 KiB	3.7 KiB
…deps/chunk-2YIMICFJ.js?v=c37361c9 (localhost)	77.4 KiB	45.0 KiB
…cjs/react.development.js	75.2 KiB	45.0 KiB
…deps/lucide-react.js?v=5264625b (localhost)	1,129.8 KiB	23.9 KiB
…lucide-react/dist/esm/icons/index.js	47.9 KiB	23.9 KiB
…deps/react_jsx-dev-runtime.js?v=81295adb (localhost)	36.1 KiB	20.7 KiB
…cjs/react-jsx-dev-runtime.development.js	35.2 KiB	20.7 KiB
Unattributable	**108.7 KiB**	**72.5 KiB**
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js	60.1 KiB	39.5 KiB
../../react-devtools-shared/src/utils.js	8.1 KiB	7.3 KiB
../../react-devtools-shared/src/backend/agent.js	8.6 KiB	5.3 KiB
../../react-devtools-shared/src/hydration.js	4.1 KiB	3.9 KiB
../../../node_modules/util/util.js	4.6 KiB	3.6 KiB
../../../node_modules/lru-cache/index.js	3.5 KiB	2.4 KiB
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/installHook.js	48.6 KiB	33.1 KiB
../../react-devtools-shared/src/backend/fiber/renderer.js	16.2 KiB	8.5 KiB
../../../build/oss-experimental/react-debug-tools/cjs/react-debug-tools.production.js	4.2 KiB	3.8 KiB
../../react-devtools-shared/src/backend/legacy/renderer.js	3.6 KiB	3.6 KiB
../../react-devtools-shared/src/utils.js	2.4 KiB	2.1 KiB

URL	Transfer Size	Potential Savings
../../../build/oss-experimental/react/cjs/react.production.js	2.7 KiB	2.0 KiB

⚠ **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.	Pending browser support
http://localhost:3000	

Serve images in next-gen formats — Potential savings of 13 KiB ⌄

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](#). [FCP] [LCP]

URL	Resource Size	Potential Savings
localhost [1st Party]	**56.5 KiB**	**13.4 KiB**
/assets/carlos.jpg (localhost) img#profpic	56.5 KiB	13.4 KiB

Properly size images — Potential savings of 37 KiB ⌄

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

URL	Resource Size	Potential Savings
localhost [1st Party]	**56.5 KiB**	**37.2 KiB**

URL	Resource Size	Potential Savings
/assets/carlos.jpg (localhost) img#profpic	56.5 KiB	37.2 KiB

Remove duplicate modules in JavaScript bundles   — Potential savings of 11 KiB   ⌃

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

Source	Transfer Size	Potential Savings
`../../react-devtools-shared/src/utils.js`		0 KiB
`chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js`	10 KiB	
`chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/installHook.js`	0 KiB	
`Other`		0 KiB
`chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js`		
`chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/installHook.js`		
`node_modules/util`		0 KiB
`chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js`	0 KiB	
`chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/installHook.js`	0 KiB	
`node_modules/lru-cache`		0 KiB
`chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js`	0 KiB	
`chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/installHook.js`	0 KiB	

Source	Transfer Size	Potential Savings
node_modules/yallist		0 KiB
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js	0 KiB	
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/installHook.js	0 KiB	
../../react-devtools-shared/src/hydration.js		0 KiB
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js	0 KiB	
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/installHook.js	0 KiB	

Avoid enormous network payloads — Total size was 3,892 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 (2)

URL	Transfer Size
localhost 1st Party	**3,796.1 KiB**
…deps/react-icons_fa.js?v=d1b29098 (localhost)	1,379.1 KiB
…deps/lucide-react.js?v=5264625b (localhost)	1,130.1 KiB
…deps/react-dom_client.js?v=7c13e35a (localhost)	907.3 KiB
/@vite/client (localhost)	134.4 KiB
…deps/chunk-2YIMICFJ.js?v=c37361c9 (localhost)	77.6 KiB
/@react-refresh (localhost)	74.4 KiB
/assets/carlos.jpg (localhost)	56.8 KiB
…deps/react_jsx-dev-runtime.js?v=81295adb (localhost)	36.4 KiB
JSDelivr CDN Cdn	**49.2 KiB**

URL	Transfer Size
…css/bootstrap.min.css (cdn.jsdelivr.net)	26.9 KiB
…js/bootstrap.bundle.min.js (cdn.jsdelivr.net)	22.3 KiB

○ 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
Other	440 ms
Script Evaluation	271 ms
Script Parsing & Compilation	71 ms
Style & Layout	32 ms
Parse HTML & CSS	27 ms
Rendering	3 ms

○ Avoid long main-thread tasks — 2 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]

URL	Start Time	Duration
Unattributable		162 ms
Unattributable	3,908 ms	162 ms
localhost [1st Party]		58 ms
http://localhost:3000	604 ms	58 ms

○ User Timing marks and measures — 198 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 about User Timing marks](#).

Name	Type	Start Time	Duration
__v3	Mark	0.00 ms	
clearMarks	Mark	22.21 ms	
--schedule-render-16	Mark	125.01 ms	
clearMarks	Mark	125.08 ms	
--render-start-16	Mark	127.56 ms	
clearMarks	Mark	127.62 ms	
--component-render-start-GlobalErrorHandler	Mark	129.14 ms	
clearMarks	Mark	129.38 ms	
--component-render-stop	Mark	129.61 ms	
clearMarks	Mark	129.65 ms	
--component-render-stop	Mark	130.76 ms	
clearMarks	Mark	131.37 ms	
--suspense-suspend-0-Lazy-mount-16-	Mark	131.55 ms	
clearMarks	Mark	131.65 ms	
--render-stop	Mark	133.64 ms	
clearMarks	Mark	133.76 ms	
--commit-start-16	Mark	134.14 ms	
clearMarks	Mark	134.20 ms	
--react-version-18.3.1	Mark	134.23 ms	
clearMarks	Mark	134.26 ms	
--profiler-version-1	Mark	134.26 ms	

Name	Type	Start Time	Duration
clearMarks	Mark	134.29 ms	
--react-internal-module-start- at http://localhost:3000/node_modules/.vite/deps/chunk-2YIMICFJ.js?v=c37361c9:39:70	Mark	134.32 ms	
clearMarks	Mark	134.35 ms	
--react-internal-module-stop- at http://localhost:3000/node_modules/.vite/deps/chunk-2YIMICFJ.js?v=c37361c9:1898:69	Mark	134.35 ms	
clearMarks	Mark	134.37 ms	
--react-internal-module-start- at http://localhost:3000/node_modules/.vite/deps/react-dom_client.js?v=7c13e35a:14:70	Mark	134.38 ms	
clearMarks	Mark	134.40 ms	
--react-internal-module-stop- at http://localhost:3000/node_modules/.vite/deps/react-dom_client.js?v=7c13e35a:449:69	Mark	134.40 ms	
clearMarks	Mark	134.42 ms	
--react-internal-module-start- at http://localhost:3000/node_modules/.vite/deps/react-dom_client.js?v=7c13e35a:476:70	Mark	134.43 ms	
clearMarks	Mark	134.45 ms	
--react-internal-module-stop- at http://localhost:3000/node_modules/.vite/deps/react-dom_client.js?v=7c13e35a:21567:69	Mark	134.46 ms	
clearMarks	Mark	134.48 ms	
--react-lane-labels-Sync,InputContinuousHydration,InputContinuous,DefaultHydration,Default,TransitionHydration,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Retry,Retry,Retry,Retry,Retry,SelectiveHydration,IdleHydration,Idle,Offscreen	Mark	134.50 ms	
clearMarks	Mark	134.53 ms	

Name	Type	Start Time	Duration
--layout-effects-start-16	Mark	135.31 ms	
clearMarks	Mark	135.36 ms	
--layout-effects-stop	Mark	135.54 ms	
clearMarks	Mark	135.60 ms	
--commit-stop	Mark	136.51 ms	
clearMarks	Mark	136.55 ms	
--suspense-resolved-0-Lazy	Mark	243.07 ms	
clearMarks	Mark	243.33 ms	
--render-start-4194304	Mark	246.95 ms	
clearMarks	Mark	247.00 ms	
--component-render-start-App	Mark	247.34 ms	
clearMarks	Mark	247.37 ms	
--component-render-stop	Mark	247.48 ms	
clearMarks	Mark	247.51 ms	
--component-render-start-About	Mark	248.51 ms	
clearMarks	Mark	248.55 ms	
--component-render-stop	Mark	248.60 ms	
clearMarks	Mark	248.62 ms	
--component-render-start-Frontend	Mark	248.78 ms	
clearMarks	Mark	248.79 ms	
--component-render-stop	Mark	248.90 ms	
clearMarks	Mark	248.92 ms	
--component-render-start-FaHtml5	Mark	249.04 ms	

Name	Type	Start Time	Duration
clearMarks	Mark	249.06 ms	
--component-render-stop	Mark	249.62 ms	
clearMarks	Mark	249.65 ms	
--component-render-start-IconBase	Mark	249.70 ms	
clearMarks	Mark	249.72 ms	
--component-render-stop	Mark	249.76 ms	
clearMarks	Mark	249.79 ms	
--component-render-start-Context.Consumer	Mark	249.90 ms	
clearMarks	Mark	249.92 ms	
--component-render-stop	Mark	250.06 ms	
clearMarks	Mark	250.10 ms	
--component-render-start-FaCss3Alt	Mark	251.20 ms	
clearMarks	Mark	251.32 ms	
--component-render-stop	Mark	251.45 ms	
clearMarks	Mark	251.50 ms	
--component-render-start-IconBase	Mark	251.54 ms	
clearMarks	Mark	251.57 ms	
--component-render-stop	Mark	251.58 ms	
clearMarks	Mark	251.60 ms	
--component-render-start-Context.Consumer	Mark	251.62 ms	
clearMarks	Mark	251.66 ms	
--component-render-stop	Mark	251.70 ms	
clearMarks	Mark	251.73 ms	

Name	Type	Start Time	Duration
--render-yield	Mark	252.05 ms	
clearMarks	Mark	252.10 ms	
--render-start-4194304	Mark	255.38 ms	
clearMarks	Mark	255.45 ms	
--component-render-start-FaJs	Mark	255.57 ms	
clearMarks	Mark	255.82 ms	
--component-render-stop	Mark	255.94 ms	
clearMarks	Mark	257.49 ms	
--component-render-start-IconBase	Mark	257.58 ms	
clearMarks	Mark	257.63 ms	
--component-render-stop	Mark	257.64 ms	
clearMarks	Mark	257.65 ms	
--component-render-start-Context.Consumer	Mark	257.67 ms	
clearMarks	Mark	258.56 ms	
--component-render-stop	Mark	258.63 ms	
clearMarks	Mark	258.65 ms	
--component-render-start-FaReact	Mark	259.02 ms	
clearMarks	Mark	259.05 ms	
--component-render-stop	Mark	259.15 ms	
clearMarks	Mark	259.18 ms	
--component-render-start-IconBase	Mark	259.22 ms	
clearMarks	Mark	259.25 ms	
--component-render-stop	Mark	259.25 ms	

Name	Type	Start Time	Duration
clearMarks	Mark	259.32 ms	
--component-render-start-Context.Consumer	Mark	259.34 ms	
clearMarks	Mark	259.36 ms	
--component-render-stop	Mark	259.37 ms	
clearMarks	Mark	259.38 ms	
--component-render-start-FaBootstrap	Mark	259.48 ms	
clearMarks	Mark	259.50 ms	
--component-render-stop	Mark	259.54 ms	
clearMarks	Mark	259.56 ms	
--component-render-start-IconBase	Mark	259.57 ms	
clearMarks	Mark	259.58 ms	
--component-render-stop	Mark	259.59 ms	
clearMarks	Mark	259.60 ms	
--component-render-start-Context.Consumer	Mark	259.61 ms	
clearMarks	Mark	259.63 ms	
--component-render-stop	Mark	259.64 ms	
clearMarks	Mark	259.65 ms	
--component-render-start-FaNode	Mark	259.77 ms	
clearMarks	Mark	259.78 ms	
--component-render-stop	Mark	259.82 ms	
clearMarks	Mark	259.85 ms	
--component-render-start-IconBase	Mark	259.86 ms	
clearMarks	Mark	259.88 ms	

Name	Type	Start Time	Duration
--component-render-stop	Mark	259.88 ms	
clearMarks	Mark	259.90 ms	
--component-render-start-Context.Consumer	Mark	259.92 ms	
clearMarks	Mark	259.93 ms	
--component-render-stop	Mark	259.96 ms	
clearMarks	Mark	259.98 ms	
--component-render-start-Links	Mark	260.12 ms	
clearMarks	Mark	260.14 ms	
--component-render-stop	Mark	260.25 ms	
clearMarks	Mark	260.27 ms	
--component-render-start-Github	Mark	260.42 ms	
clearMarks	Mark	260.44 ms	
--component-render-stop	Mark	260.62 ms	
clearMarks	Mark	260.64 ms	
--render-yield	Mark	260.65 ms	
clearMarks	Mark	260.66 ms	
--render-start-4194304	Mark	261.77 ms	
clearMarks	Mark	261.80 ms	
--component-render-start-ForwardRef(Anonymous)	Mark	261.82 ms	
clearMarks	Mark	261.85 ms	
--component-render-stop	Mark	262.02 ms	
clearMarks	Mark	262.04 ms	
--component-render-start-Linkedin	Mark	262.41 ms	

Name	Type	Start Time	Duration
clearMarks	Mark	262.44 ms	
--component-render-stop	Mark	262.47 ms	
clearMarks	Mark	262.49 ms	
--component-render-start-ForwardRef(Anonymous)	Mark	262.50 ms	
clearMarks	Mark	262.51 ms	
--component-render-stop	Mark	262.65 ms	
clearMarks	Mark	262.68 ms	
--component-render-start-Mail	Mark	263.06 ms	
clearMarks	Mark	263.13 ms	
--component-render-stop	Mark	263.16 ms	
clearMarks	Mark	263.19 ms	
--component-render-start-ForwardRef(Anonymous)	Mark	263.19 ms	
clearMarks	Mark	263.22 ms	
--component-render-stop	Mark	263.26 ms	
clearMarks	Mark	263.29 ms	
--render-stop	Mark	263.42 ms	
clearMarks	Mark	263.45 ms	
--commit-start-4194304	Mark	263.57 ms	
clearMarks	Mark	263.60 ms	
--react-version-18.3.1	Mark	263.61 ms	
clearMarks	Mark	263.64 ms	
--profiler-version-1	Mark	263.64 ms	
clearMarks	Mark	263.66 ms	

Name	Type	Start Time	Duration
--react-internal-module-start- at http://localhost:3000/node_modules/.vite/deps/chunk-2YIMICFJ.js?v=c37361c9:39:70	Mark	263.68 ms	
clearMarks	Mark	263.75 ms	
--react-internal-module-stop- at http://localhost:3000/node_modules/.vite/deps/chunk-2YIMICFJ.js?v=c37361c9:1898:69	Mark	263.76 ms	
clearMarks	Mark	263.79 ms	
--react-internal-module-start- at http://localhost:3000/node_modules/.vite/deps/react-dom_client.js?v=7c13e35a:14:70	Mark	263.79 ms	
clearMarks	Mark	263.82 ms	
--react-internal-module-stop- at http://localhost:3000/node_modules/.vite/deps/react-dom_client.js?v=7c13e35a:449:69	Mark	263.82 ms	
clearMarks	Mark	263.85 ms	
--react-internal-module-start- at http://localhost:3000/node_modules/.vite/deps/react-dom_client.js?v=7c13e35a:476:70	Mark	263.85 ms	
clearMarks	Mark	263.88 ms	
--react-internal-module-stop- at http://localhost:3000/node_modules/.vite/deps/react-dom_client.js?v=7c13e35a:21567:69	Mark	263.88 ms	
clearMarks	Mark	263.89 ms	
--react-lane-labels-Sync,InputContinuousHydration,InputContinuous,DefaultHydration,Default,TransitionHydration,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Transition,Retry,Retry,Retry,Retry,Retry,SelectiveHydration,IdleHydration,Idle,Offscreen	Mark	263.92 ms	
clearMarks	Mark	263.94 ms	
--layout-effects-start-4194304	Mark	264.80 ms	

Name	Type	Start Time	Duration
clearMarks	Mark	265.02 ms	
--layout-effects-stop	Mark	265.61 ms	
clearMarks	Mark	265.66 ms	
--commit-stop	Mark	266.30 ms	
clearMarks	Mark	266.34 ms	
--passive-effects-start-4194304	Mark	266.45 ms	
clearMarks	Mark	266.47 ms	
--passive-effects-stop	Mark	266.64 ms	
clearMarks	Mark	266.65 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 about the Time to First Byte metric. FCP LCP

URL	Time Spent
localhost 1st Party	**0 ms**
http://localhost:3000	0 ms

○ Avoids an excessive DOM size — 53 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		53
Maximum DOM Depth	□ path	9
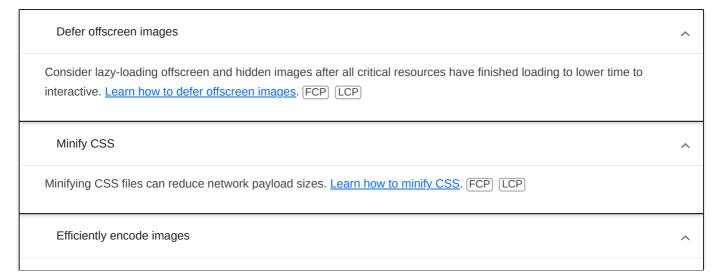
Statistic	Element	Value
Maximum Child Elements	div.App	6

○ Avoid chaining critical requests — 14 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: **179.019 ms**

Initial Navigation

    http://localhost:3000

        /@vite/client (localhost)

            …client/env.mjs (localhost) **- 8.682 ms, 4.18 KiB**

          …css/bootstrap.min.css (cdn.jsdelivr.net) **- 58.542 ms, 26.87 KiB**

          /src/index.css (localhost) **- 27.083 ms, 0.28 KiB**

          /src/main.jsx?t=173… (localhost)

            …deps/react_jsx-dev-runtime.js?v=81295adb (localhost) **- 5.238 ms, 36.37 KiB**

            …deps/react.js?v=bda1ebaa (localhost)

              …deps/chunk-2YIMICFJ.js?v=c37361c9 (localhost)

                /src/App.jsx?t=173… (localhost)

                  /src/App.css (localhost) **- 3.969 ms, 1.51 KiB**

                  /src/About.jsx (localhost)

                    /src/About.css (localhost) **- 2.893 ms, 1.04 KiB**

                  /src/Links.jsx (localhost)

                    …deps/lucide-react.js?v=5264625b (localhost) **- 22.512 ms, 1,130.08 KiB**

                    /src/Links.css (localhost) **- 16.183 ms, 1.14 KiB**

                  /src/Frontend.jsx?t=173… (localhost)

                    /src/Frontend.css?t=173… (localhost) **- 16.398 ms, 1.09 KiB**

                    …deps/react-icons_fa.js?v=d1b29098 (localhost) **- 22.65 ms, 1,379.10 KiB**

            …deps/react-dom_client.js?v=7c13e35a (localhost) **- 15.381 ms, 907.34 KiB**

          /src/index.css (localhost) **- 11.293 ms, 0.74 KiB**

        …js/bootstrap.bundle.min.js (cdn.jsdelivr.net) **- 51.918 ms, 22.32 KiB**

        /@react-refresh (localhost) **- 13.487 ms, 74.43 KiB**

○ 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 how to reduce Javascript execution time](#). TBT

☑ Show 3rd-party resources (1)

URL	Total CPU Time	Script Evaluation	Script Parse
Unattributable	**333 ms**	**42 ms**	**0 ms**
Unattributable	333 ms	42 ms	0 ms
localhost [1st Party]	**290 ms**	**143 ms**	**26 ms**
http://localhost:3000	172 ms	39 ms	21 ms
…deps/react-dom_client.js?v=7c13e35a (localhost)	118 ms	104 ms	4 ms
JSDelivr CDN [Cdn]	**60 ms**	**50 ms**	**6 ms**
…js/bootstrap.bundle.min.js (cdn.jsdelivr.net)	60 ms	50 ms	6 ms

○ 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 how to minimize third-party impact. [TBT]

Third-Party	Transfer Size	Main-Thread Blocking Time
JSDelivr CDN [Cdn]	**49 KiB**	**0 ms**
…css/bootstrap.min.css (cdn.jsdelivr.net)	27 KiB	0 ms
…js/bootstrap.bundle.min.js (cdn.jsdelivr.net)	22 KiB	0 ms

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

PASSED AUDITS (19)   Hide

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]

Efficiently encode images   ⌃

Optimized images load faster and consume less cellular data. [Learn how to efficiently encode images](#). [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](#). [LCP] [FCP]

Avoid multiple page redirects   ⌃

Redirects introduce additional delays before the page can be loaded. [Learn how to avoid page redirects](#). [LCP] [FCP]

Use HTTP/2   ⌃

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more about HTTP/2](#). [LCP] [FCP]

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](#) [FCP] [LCP]

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

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 about using the viewport meta tag.

69

Accessibility

These checks highlight opportunities to improve the accessibility of your
web app. Automatic detection can only detect a subset of issues and does
not guarantee the accessibility of your web app, so manual testing is also
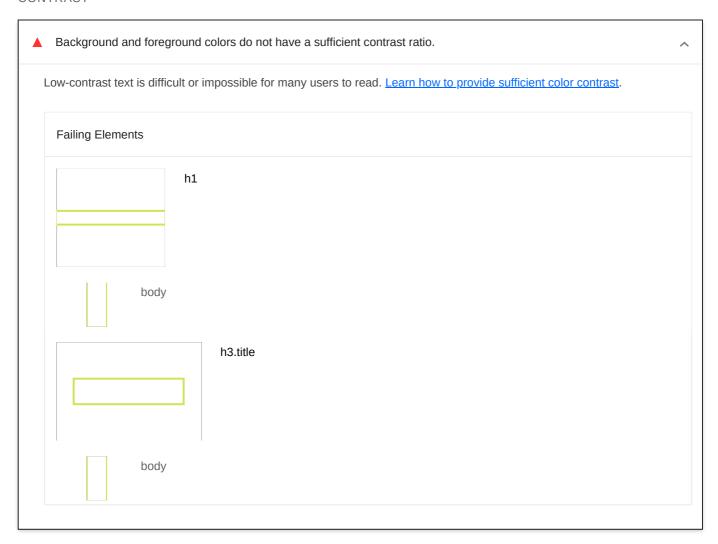encouraged.

NAMES AND LABELS

⚠ Buttons do not have an accessible name   ⌄

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn how to make buttons more accessible](#).

Failing Elements

button#GitHub.btn.btn-primary.rounded-2.border.border-2

button#LinkedIn.btn.btn-primary.rounded-2.border.border-2

button#Mail.btn.btn-primary.rounded-2.border.border-2

⚠ Links do not have a discernible name   ⌄

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn how to make links accessible](#).
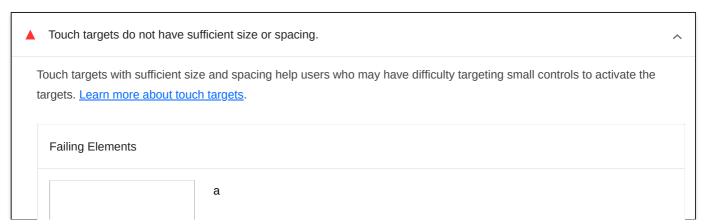
Failing Elements

a

a

a

Failing Elements

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

CONTRAST

⚠️ Background and foreground colors do not have a sufficient contrast ratio.   ⌃

Low-contrast text is difficult or impossible for many users to read. Learn how to provide sufficient color contrast.

Failing Elements

h1

body

h3.title

body

These are opportunities to improve the legibility of your content.

BEST PRACTICES

⚠️ Touch targets do not have sufficient size or spacing.   ⌃

Touch targets with sufficient size and spacing help users who may have difficulty targeting small controls to activate the targets. Learn more about touch targets.

Failing Elements

a

Failing Elements

button#GitHub.btn.btn-primary.rounded-2.border.border-2

a

button#LinkedIn.btn.btn-primary.rounded-2.border.border-2

a

button#Mail.btn.btn-primary.rounded-2.border.border-2

These items highlight common accessibility best practices.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)   Hide

○ Interactive controls are keyboard focusable ⌃

Custom interactive controls are keyboard focusable and display a focus indicator. Learn how to make custom controls focusable.

○ Interactive elements indicate their purpose and state ⌃

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. Learn how to decorate interactive elements with affordance hints.

○ The page has a logical tab order ⌃

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. Learn more about logical tab ordering.

○ Visual order on the page follows DOM order ⌃

DOM order matches the visual order, improving navigation for assistive technology. Learn more about DOM and visual ordering.

○ User focus is not accidentally trapped in a region ⌃

A user can tab into and out of any control or region without accidentally trapping their focus. Learn how to avoid focus traps.

○ The user's focus is directed to new content added to the page ⌃

If new content, such as a dialog, is added to the page, the user's focus is directed to it. Learn how to direct focus to new content.

○ HTML5 landmark elements are used to improve navigation ⌃

Landmark elements (`<main>`, `<nav>`, etc.) are used to improve the keyboard navigation of the page for assistive technology. Learn more about landmark elements.

○ Offscreen content is hidden from assistive technology ⌃

Offscreen content is hidden with display: none or aria-hidden=true. Learn how to properly hide offscreen content.

○ Custom controls have associated labels ⌃

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. Learn more about custom controls and labels.

○ Custom controls have ARIA roles ⌃

Custom interactive controls have appropriate ARIA roles. Learn how to add roles to custom controls.

These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (10)   Hide

`[aria-hidden="true"]` is not present on the document `<body>` ⌃

Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document `<body>`. Learn how `aria-hidden` affects the document body.

Image elements have `[alt]` attributes ⌃

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. Learn more about the `alt` attribute.

`[user-scalable="no"]` is not used in the `<meta name="viewport">` element and the `[maximum-scale]` attribute is not less than 5.

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. Learn more about the viewport meta tag.

Document has a `<title>` element

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. Learn more about document titles.

`<html>` element has a `[lang]` attribute

If a page doesn't specify a `lang` attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. Learn more about the `lang` attribute.

`<html>` element has a valid value for its `[lang]` attribute

Specifying a valid BCP 47 language helps screen readers announce text properly. Learn how to use the `lang` attribute.

Lists contain only `<li>` elements and script supporting elements (`<script>` and `<template>`).

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. Learn more about proper list structure.

List items (`<li>`) are contained within `<ul>`, `<ol>` or `<menu>` parent elements

Screen readers require list items (`<li>`) to be contained within a parent `<ul>`, `<ol>` or `<menu>` to be announced properly. Learn more about proper list structure.

Heading elements appear in a sequentially-descending order

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. Learn more about heading order.

Image elements do not have `[alt]` attributes that are redundant text.

Informative elements should aim for short, descriptive alternative text. Alternative text that is exactly the same as the text adjacent to the link or image is potentially confusing for screen reader users, because the text will be read twice. Learn more about the `alt` attribute.

NOT APPLICABLE (43)   Hide

○   `[accesskey]` values are unique   ⌄

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. [Learn more about access keys](#).

○   `[aria-*]` attributes match their roles   ⌄

Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. [Learn how to match ARIA attributes to their roles](#).

○   Uses ARIA roles only on compatible elements   ⌄

Many HTML elements can only be assigned certain ARIA roles. Using ARIA roles where they are not allowed can interfere with the accessibility of the web page. [Learn more about ARIA roles](#).

○   `button`, `link`, and `menuitem` elements have accessible names   ⌄

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to make command elements more accessible](#).

○   ARIA attributes are used as specified for the element's role   ⌄

Some ARIA attributes are only allowed on an element under certain conditions. [Learn more about conditional ARIA attributes](#).

○   Deprecated ARIA roles were not used   ⌄

Deprecated ARIA roles may not be processed correctly by assistive technology. [Learn more about deprecated ARIA roles](#).

○   Elements with `role="dialog"` or `role="alertdialog"` have accessible names.   ⌄

ARIA dialog elements without accessible names may prevent screen readers users from discerning the purpose of these elements. [Learn how to make ARIA dialog elements more accessible](#).

○   `[aria-hidden="true"]` elements do not contain focusable descendents   ⌄

Focusable descendents within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn how `aria-hidden` affects focusable elements](#).

○   ARIA input fields have accessible names   ⌄

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more about input field labels](#).

○   ARIA `meter` elements have accessible names   ⌄

When a meter element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to name `meter` elements](#).

○ ARIA `progressbar` elements have accessible names   ⌃

When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to label `progressbar` elements](#).

○ Elements use only permitted ARIA attributes   ⌃

Using ARIA attributes in roles where they are prohibited can mean that important information is not communicated to users of assistive technologies. [Learn more about prohibited ARIA roles](#).

○ `[role]`s have all required `[aria-*]` attributes   ⌃

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more about roles and required attributes](#).

○ Elements with an ARIA `[role]` that require children to contain a specific `[role]` have all required children.   ⌃

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more about roles and required children elements](#).

○ `[role]`s are contained by their required parent element   ⌃

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more about ARIA roles and required parent element](#).

○ `[role]` values are valid   ⌃

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more about valid ARIA roles](#).

○ Elements with the `role=text` attribute do not have focusable descendents.   ⌃

Adding `role=text` around a text node split by markup enables VoiceOver to treat it as one phrase, but the element's focusable descendents will not be announced. [Learn more about the `role=text` attribute](#).

○ ARIA toggle fields have accessible names   ⌃

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more about toggle fields](#).

○ ARIA `tooltip` elements have accessible names   ⌃

When a tooltip element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to name `tooltip` elements](#).

○ ARIA `treeitem` elements have accessible names ⌃

When a `treeitem` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more about labeling `treeitem` elements](#).

○ `[aria-*]` attributes have valid values ⌃

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more about valid values for ARIA attributes](#).

○ `[aria-*]` attributes are valid and not misspelled ⌃

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more about valid ARIA attributes](#).

○ The page contains a heading, skip link, or landmark region ⌃

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more about bypass blocks](#).

○ `<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements. ⌃

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn how to structure definition lists correctly](#).

○ Definition list items are wrapped in `<dl>` elements ⌃

Definition list items (<dt> and <dd>) must be wrapped in a parent <dl> element to ensure that screen readers can properly announce them. [Learn how to structure definition lists correctly](#).

○ ARIA IDs are unique ⌃

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn how to fix duplicate ARIA IDs](#).

○ No form fields have multiple labels ⌃

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn how to use form labels](#).

○ `<frame>` or `<iframe>` elements have a title ⌃

Screen reader users rely on frame titles to describe the contents of frames. [Learn more about frame titles](#).

○ `<html>` element has an `[xml:lang]` attribute with the same base language as the `[lang]` attribute. ⌃

If the webpage does not specify a consistent language, then the screen reader might not announce the page's text correctly. Learn more about the `lang` attribute.

○ Input buttons have discernible text. ⌃

Adding discernable and accessible text to input buttons may help screen reader users understand the purpose of the input button. Learn more about input buttons.

○ `<input type="image">` elements have `[alt]` text ⌃

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. Learn about input image alt text.

○ Form elements have associated labels ⌃

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. Learn more about form element labels.

○ Links are distinguishable without relying on color. ⌃

Low-contrast text is difficult or impossible for many users to read. Link text that is discernible improves the experience for users with low vision. Learn how to make links distinguishable.

○ The document does not use `<meta http-equiv="refresh">` ⌃

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. Learn more about the refresh meta tag.

○ `<object>` elements have alternate text ⌃

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. Learn more about alt text for `object` elements.

○ Select elements have associated label elements. ⌃

Form elements without effective labels can create frustrating experiences for screen reader users. Learn more about the `select` element.

○ Skip links are focusable. ⌃

Including a skip link can help users skip to the main content to save time. Learn more about skip links.

○   No element has a `[tabindex]` value greater than 0   ∧

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. Learn more about the `tabindex` attribute.

○   Tables have different content in the summary attribute and `<caption>`.   ∧

The summary attribute should describe the table structure, while `<caption>` should have the onscreen title. Accurate table mark-up helps users of screen readers. Learn more about summary and caption.

○   Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table.   ∧

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. Learn more about the `headers` attribute.

○   `<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe.   ∧

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. Learn more about table headers.

○   `[lang]` attributes have a valid value   ∧

Specifying a valid BCP 47 language on elements helps ensure that text is pronounced correctly by a screen reader. Learn how to use the `lang` attribute.

○   `<video>` elements contain a `<track>` element with `[kind="captions"]`   ∧

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. Learn more about video captions.

100

Best Practices

TRUST AND SAFETY

○   Ensure CSP is effective against XSS attacks   ∧

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. Learn how to use a CSP to prevent XSS

Description	Directive	Severity
No CSP found in enforcement mode		High

PASSED AUDITS (15)   Hide

Uses HTTPS ⌃

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more about HTTPS](#).

Avoids deprecated APIs ⌃

Deprecated APIs will eventually be removed from the browser. [Learn more about deprecated APIs](#).

Avoids third-party cookies ⌃

Chrome is moving towards a new experience that allows users to choose to browse without third-party cookies. [Learn more about third-party cookies](#).

Allows users to paste into input fields ⌃

Preventing input pasting is a bad practice for the UX, and weakens security by blocking password managers. [Learn more about user-friendly input fields](#).

Avoids requesting the geolocation permission on page load ⌃

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more about the geolocation permission](#).

Avoids requesting the notification permission on page load ⌃

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more about responsibly getting permission for notifications](#).

Displays images with correct aspect ratio ⌃

Image display dimensions should match natural aspect ratio. [Learn more about image aspect ratio](#).

Serves images with appropriate resolution ⌃

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn how to provide responsive images](#).

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 about using the viewport meta tag.

Document uses legible font sizes   — 100% legible text   ⌃

Font sizes less than 12px are too small to be legible and require mobile visitors to "pinch to zoom" in order to read. Strive to have >60% of page text ≥12px. Learn more about legible font sizes.

Source	Selector	% of Page Text	Font Size
Legible text		100.00%	≥ 12px

Page has the HTML doctype   ⌃

Specifying a doctype prevents the browser from switching to quirks-mode. Learn more about the doctype declaration.

Properly defines charset   ⌃

A character encoding declaration is required. It can be done with a `<meta>` tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. Learn more about declaring the character encoding.

No browser errors logged to the console   ⌃

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. Learn more about this errors in console diagnostic audit

No issues in the `Issues` panel in Chrome Devtools   ⌃

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

Page has valid source maps   ⌃

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. Learn more about source maps.

☑ Show 3rd-party resources (1)

URL	Map URL
Unattributable	

URL	Map URL
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js	chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/react_devtools_backend_compact.js.map
Warning: missing 35 items in `.sourcesContent`	
chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/installHook.js	chrome-extension://fmkadmapgofadopljbjfkapdkoienihi/build/installHook.js.map
Warning: missing 50 items in `.sourcesContent`	

JSDelivr CDN [Cdn]

…js/bootstrap.bundle.min.js (cdn.jsdelivr.net)   …js/bootstrap.bundle.min.js.map (cdn.jsdelivr.net)

localhost [1st Party]

/src/main.jsx?t=173… (localhost)

/src/Links.jsx (localhost)

/src/Frontend.jsx?t=173… (localhost)

/src/App.jsx?t=173… (localhost)

/src/About.jsx (localhost)

…client/env.mjs (localhost)

…deps/react.js?v=bda1ebaa (localhost)   …deps/react.js.map (localhost)

…deps/react-icons_fa.js?v=d1b29098 (localhost)   …deps/react-icons_fa.js.map (localhost)

…deps/react-dom_client.js?v=7c13e35a (localhost)   …deps/react-dom_client.js.map (localhost)

…deps/react_jsx-dev-runtime.js?
v=81295adb (localhost)   …deps/react_jsx-dev-runtime.js.map (localhost)

…deps/lucide-react.js?v=5264625b (localhost)   …deps/lucide-react.js.map (localhost)

…deps/chunk-2YIMICFJ.js?v=c37361c9 (localhost)   …deps/chunk-2YIMICFJ.js.map (localhost)

/@vite/client (localhost)

/@react-refresh (localhost)

NOT APPLICABLE (2)   Hide

○ Redirects HTTP traffic to HTTPS   ⌃

Make sure that you redirect all HTTP traffic to HTTPS in order to enable secure web features for all your users. Learn more.

○ Detected JavaScript libraries   ⌃

All front-end JavaScript libraries detected on the page. Learn more about this JavaScript library detection diagnostic audit.

83

SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on Core Web Vitals. Learn more about Google Search Essentials.
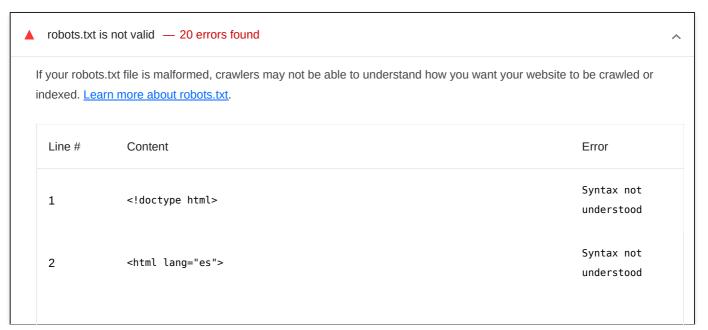
CONTENT BEST PRACTICES

⚠ Document does not have a meta description   ⌃

Meta descriptions may be included in search results to concisely summarize page content. Learn more about the meta description.

Format your HTML in a way that enables crawlers to better understand your app's content.

CRAWLING AND INDEXING

⚠ robots.txt is not valid — 20 errors found   ⌃

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. Learn more about robots.txt.

Line #	Content	Error
1	`<!doctype html>`	Syntax not understood
2	`<html lang="es">`	Syntax not understood

Line #	Content	Error
3	`<head>`	Syntax not understood
4	`<script type="module">import { injectIntoGlobalHook } from "/@react-refresh";`	Syntax not understood
5	`injectIntoGlobalHook(window);`	Syntax not understood
6	`window.$RefreshReg$ = () => {};`	Syntax not understood
7	`window.$RefreshSig$ = () => (type) => type;</script>`	Syntax not understood
9	`<script type="module" src="/@vite/client"></script>`	Syntax not understood
11	`<link rel= "stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">`	Unknown directive
12	`<meta charset="UTF-8" />`	Syntax not understood
13	`<meta name="viewport" content="width=device-width, initial-scale=1.0" />`	Syntax not understood
14	`<title>Carlos Arroyo</title>`	Syntax not understood
15	`<link rel="stylesheet" href="/src/index.css" />`	Syntax not understood
16	`</head>`	Syntax not understood
17	`<body>`	Syntax not understood
18	`<div id="root"></div>`	Syntax not understood
19	`<script type="module" src="/src/main.jsx?t=1732461871663" defer></script>`	Syntax not understood

Line #	Content	Error
20	`<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>`	Unknown directive
21	`</body>`	Syntax not understood
22	`</html>`	Syntax not understood

To appear in search results, crawlers need access to your app.

ADDITIONAL ITEMS TO MANUALLY CHECK (1)   Hide

○   Structured data is valid   ⌃

Run the Structured Data Testing Tool and the Structured Data Linter to validate structured data. Learn more about Structured Data.

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (7)   Hide

Page isn't blocked from indexing   ⌃

Search engines are unable to include your pages in search results if they don't have permission to crawl them. Learn more about crawler directives.

Document has a `<title>` element   ⌃

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. Learn more about document titles.

Page has successful HTTP status code   ⌃

Pages with unsuccessful HTTP status codes may not be indexed properly. Learn more about HTTP status codes.

Links have descriptive text   ⌃

Descriptive link text helps search engines understand your content. Learn how to make links more accessible.

Links are crawlable   ⌃

Search engines may use `href` attributes on links to crawl websites. Ensure that the `href` attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn how to make links crawlable](#)

Image elements have `[alt]` attributes ⌄

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more about the `alt` attribute](#).

Document has a valid `hreflang` ⌄

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more about `hreflang`](#).

NOT APPLICABLE (1)   Hide

⊙ Document has a valid `rel=canonical`   ⌄

Canonical links suggest which URL to show in search results. [Learn more about canonical links](#).

▪ Captured at Nov 24, 2024, 3:28 PM UTC	▪ [Emulated Moto G Power with Lighthouse 12.2.1](#)	▪ [Single page session](#)
▪ Initial page load	▪ [Slow 4G throttling](#)	▪ [Using Chromium 131.0.0.0 with devtools](#)

Generated by **Lighthouse** 12.2.1 | [File an issue](#)