

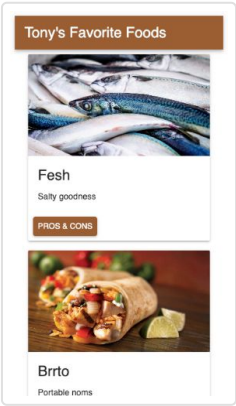
There were issues affecting this run of Lighthouse:

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



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)



▲ 0–49 50–89 90–100

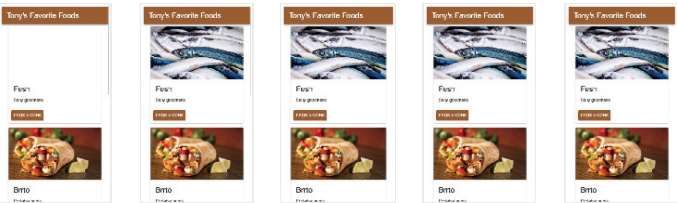
METRICS

Expand view

▲ First Contentful Paint 17.4 s	▲ Time to Interactive 23.7 s
▲ Speed Index 17.4 s	▲ Total Blocking Time 3,070 ms
▲ Largest Contentful Paint 28.6 s	Cumulative Layout Shift 0

 View Treemap

View Original Trace



Show audits relevant to: All FCP TBT LCP CLS

OPPORTUNITIES

Opportunity	Estimated Savings
▲ Enable text compression	17.4 s ^

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

URL	Transfer size	Potential savings
/bundle.js (localhost)	1,257.1 KiB	999.8 KiB
/libs/lodash.js (localhost)	543.9 KiB	449.3 KiB
/libs/jquery.js (localhost)	275.5 KiB	195.8 KiB

▲

Eliminate render-blocking resources

10.39 s

^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more](#). FCP LCP

URL	Transfer size	Potential savings
/libs/lodash.js (localhost)	544.2 KiB	2,850 ms
/libs/jquery.js (localhost)	275.8 KiB	1,500 ms

▲

Minify JavaScript

10.2 s

^

Minifying JavaScript files can reduce payload sizes and script parse time. [Learn more](#). FCP LCP




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](#).

URL	Transfer size	Potential savings
/bundle.js (localhost)	1,257.5 KiB	442.5 KiB
/libs/lodash.js (localhost)	544.2 KiB	395.5 KiB
/libs/jquery.js (localhost)	275.8 KiB	132.0 KiB
chrome-extension://dagdlcijhfbmgkjokkjjicnnfimlebc11/page_context.js	10.6 KiB	5.0 KiB

▲ Reduce unused JavaScript7.95 s ^

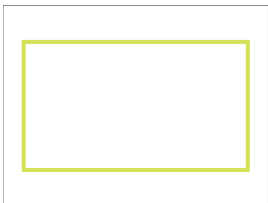
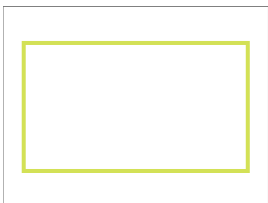
Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn more.](#) LCP

 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](#).

URL	Transfer size	Potential savings
/bundle.js (localhost)	1,257.5 KiB	387.2 KiB
/libs/jquery.js (localhost)	275.8 KiB	203.1 KiB
/libs/lodash.js (localhost)	544.2 KiB	161.6 KiB

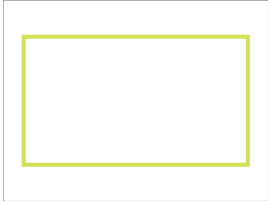
▲ Serve images in next-gen formats2.4 s ^

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more.](#)

	URL	Resource size	Potential savings
	div.MuiCard Media-root-69.MyCard-media-67 ...big/pezza.jpg (storage.googleapis.com)	812.8 KiB	393.0 KiB
	div.MuiCard Media-root-69.MyCard-media-67 ...big/brrto.jpg (storage.googleapis.com)	161.2 KiB	84.9 KiB

Efficiently encode images0.15 s ^

Optimised images load faster and consume less mobile data. [Learn more.](#)

URL		Resource size	Potential savings
	div.MuiCard Media-root-69.MyCard-media-67	...big/brtto.jpg (storage.googleapis.com) 161.2 KiB	24.0 KiB

These suggestions can help your page load faster. They don't [directly affect](#) the performance score.

DIAGNOSTICS

▲ Serve static assets with an efficient cache policy — 4 resources found ^

A long cache lifetime can speed up repeat visits to your page. [Learn more](#).

URL	Cache TTL	Transfer size
...big/soop.jpg (storage.googleapis.com)	1 h	2,295 KiB
...big/fesh.jpg (storage.googleapis.com)	1 h	2,245 KiB
...big/pezza.jpg (storage.googleapis.com)	1 h	813 KiB
...big/brtto.jpg (storage.googleapis.com)	1 h	161 KiB

▲ Registers an `unload` listener ^

The ``unload`` event does not fire reliably and listening for it can prevent browser optimisations like the back-forward cache. Use ``pagehide`` or ``visibilitychange`` events instead. [Learn more](#)


Source
page_context.js:210

▲ Avoid enormous network payloads — Total size was 7,606 KiB ^

Large network payloads cost users real money and are highly correlated with long load times. [Learn more](#). [LCP]

☒ Show 3rd-party resources (4)

URL	Transfer size
...big/soop.jpg (storage.googleapis.com)	2,294.8 KiB
...big/fesh.jpg (storage.googleapis.com)	2,244.8 KiB
/bundle.js (localhost)	1,257.5 KiB
...big/pezza.jpg (storage.googleapis.com)	813.3 KiB
/libs/lodash.js (localhost)	544.2 KiB
/libs/jquery.js (localhost)	275.8 KiB
...big/brrto.jpg (storage.googleapis.com)	161.4 KiB
chrome-extension://dagdlcijhfbmgkjokkjicnnfimlebc11/page_context.js	10.6 KiB
chrome-extension://dagdlcijhfbmgkjokkjicnnfimlebc11/style.css	2.8 KiB
http://localhost:1234	0.8 KiB

 Minimise main-thread work — 6.9 s

^

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

Category	Time Spent
Script Evaluation	6,179 ms
Script Parsing & Compilation	223 ms
Rendering	200 ms
Other	183 ms
Garbage Collection	103 ms
Style & Layout	8 ms
Parse HTML & CSS	7 ms

▲

Reduce JavaScript execution time — 6.3 s

^

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

URL	Total CPU Time	Script Evaluation	Script Parse
/bundle.js (localhost)	6,154 ms	6,004 ms	57 ms
http://localhost:1234	324 ms	35 ms	41 ms
Unattributable	138 ms	3 ms	0 ms
chrome-extension://bnjjngeaknajbdcgpfkgnonkmififhfo/build/content-script.js	127 ms	60 ms	61 ms
chrome-extension://kbfnbcaepplbcioakkpcpgfkobkghlhen/src/js/Grammarly-check.js	55 ms	34 ms	16 ms

○

Avoid chaining critical requests — 3 chains found

^

The critical request chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources or deferring the download of unnecessary resources to improve page load. [Learn more.](#) FCP LCP

Maximum critical path latency: 40 ms

Initial Navigation

- http://localhost:1234
- /libs/lodash.js (localhost) - 10 ms, 544.20 KiB

/libs/jquery.js (localhost) - 10 ms, 275.81 KiB

/bundle.js (localhost) - 20 ms, 1,257.45 KiB

○

User Timing marks and measures — 290 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	80.26 ms	1,515.54 ms
⌘ App [mount]	Measure	80.82 ms	1,514.88 ms
⌘ BrowserRouter [mount]	Measure	1,581.09 ms	14.61 ms
⌘ BrowserRouter.componentWillMount	Measure	1,581.42 ms	0.08 ms
⌘ Router [mount]	Measure	1,581.52 ms	14.19 ms
⌘ Router.componentWillMount	Measure	1,581.6 ms	0.11 ms
⌘ MuiThemeProvider [mount]	Measure	1,581.83 ms	13.87 ms
⌘ WithStyles(Header) [mount]	Measure	1,582.2 ms	4.7 ms
⌘ Header [mount]	Measure	1,583.57 ms	3.33 ms
⌘ WithStyles(AppBar) [mount]	Measure	1,583.66 ms	3.24 ms
⌘ AppBar [mount]	Measure	1,584.11 ms	2.79 ms
⌘ WithStyles(Paper) [mount]	Measure	1,584.28 ms	2.62 ms
⌘ Paper [mount]	Measure	1,584.72 ms	2.18 ms
⌘ WithStyles(Toolbar) [mount]	Measure	1,584.77 ms	2.13 ms
⌘ Toolbar [mount]	Measure	1,585.11 ms	1.79 ms
⌘ WithStyles(Typography) [mount]	Measure	1,585.15 ms	1.75 ms
⌘ Typography [mount]	Measure	1,585.83 ms	1.07 ms
⌘ WithStyles(Home) [mount]	Measure	1,587.46 ms	7.94 ms
⌘ Home [mount]	Measure	1,587.63 ms	7.77 ms
⌘ WithStyles(MyCard) [mount]	Measure	1,587.76 ms	4.94 ms
⌘ MyCard [mount]	Measure	1,587.9 ms	4.8 ms
⌘ WithStyles(Card) [mount]	Measure	1,588.04 ms	4.66 ms

Name	Type	Start Time	Duration
✱ Card [mount]	Measure	1,588.13 ms	4.58 ms
✱ WithStyles(Paper) [mount]	Measure	1,588.16 ms	4.54 ms
✱ Paper [mount]	Measure	1,588.2 ms	4.5 ms
✱ WithStyles(CardMedia) [mount]	Measure	1,588.22 ms	0.28 ms
✱ CardMedia [mount]	Measure	1,588.34 ms	0.16 ms
✱ WithStyles(CardContent) [mount]	Measure	1,588.5 ms	0.4 ms
✱ CardContent [mount]	Measure	1,588.74 ms	0.16 ms
✱ WithStyles(Typography) [mount]	Measure	1,588.77 ms	0.03 ms
✱ WithStyles(Typography) [mount]	Measure	1,588.87 ms	0.03 ms
✱ WithStyles(CardActions) [mount]	Measure	1,588.98 ms	3.63 ms
✱ CardActions [mount]	Measure	1,589.14 ms	3.46 ms
✱ WithStyles(Button) [mount]	Measure	1,589.32 ms	3.28 ms
✱ Button [mount]	Measure	1,590.35 ms	2.25 ms
✱ WithStyles(ButtonBase) [mount]	Measure	1,590.51 ms	2.09 ms
✱ ButtonBase [mount]	Measure	1,590.87 ms	1.73 ms
✱ Link [mount]	Measure	1,591.18 ms	1.42 ms
✱ WithStyles(TouchRipple) [mount]	Measure	1,591.46 ms	0.84 ms
✱ TouchRipple [mount]	Measure	1,592.03 ms	0.27 ms
✱ TransitionGroup [mount]	Measure	1,592.17 ms	0.13 ms
✱ WithStyles(MyCard) [mount]	Measure	1,592.75 ms	0.95 ms
✱ MyCard [mount]	Measure	1,592.8 ms	0.9 ms
✱ WithStyles(Card) [mount]	Measure	1,592.85 ms	0.85 ms

Name	Type	Start Time	Duration
⌘ Card [mount]	Measure	1,592.88 ms	0.82 ms
⌘ WithStyles(Paper) [mount]	Measure	1,592.9 ms	0.81 ms
⌘ Paper [mount]	Measure	1,592.93 ms	0.78 ms
⌘ WithStyles(CardMedia) [mount]	Measure	1,592.95 ms	0.05 ms
⌘ CardMedia [mount]	Measure	1,592.97 ms	0.03 ms
⌘ WithStyles(CardContent) [mount]	Measure	1,593.06 ms	0.14 ms
⌘ CardContent [mount]	Measure	1,593.08 ms	0.12 ms
⌘ WithStyles(Typography) [mount]	Measure	1,593.1 ms	0 ms
⌘ WithStyles(Typography) [mount]	Measure	1,593.19 ms	0.01 ms
⌘ WithStyles(CardActions) [mount]	Measure	1,593.27 ms	0.43 ms
⌘ CardActions [mount]	Measure	1,593.3 ms	0.4 ms
⌘ WithStyles(Button) [mount]	Measure	1,593.33 ms	0.37 ms
⌘ Button [mount]	Measure	1,593.37 ms	0.33 ms
⌘ WithStyles(ButtonBase) [mount]	Measure	1,593.39 ms	0.31 ms
⌘ ButtonBase [mount]	Measure	1,593.43 ms	0.27 ms
⌘ Link [mount]	Measure	1,593.47 ms	0.13 ms
⌘ WithStyles(TouchRipple) [mount]	Measure	1,593.53 ms	0.07 ms
⌘ TouchRipple [mount]	Measure	1,593.56 ms	0.04 ms
⌘ TransitionGroup [mount]	Measure	1,593.59 ms	0.01 ms
⌘ WithStyles(MyCard) [mount]	Measure	1,593.76 ms	0.74 ms
⌘ MyCard [mount]	Measure	1,593.78 ms	0.72 ms
⌘ WithStyles(Card) [mount]	Measure	1,593.82 ms	0.68 ms

Name	Type	Start Time	Duration
⌘ Card [mount]	Measure	1,593.84 ms	0.66 ms
⌘ WithStyles(Paper) [mount]	Measure	1,593.85 ms	0.65 ms
⌘ Paper [mount]	Measure	1,593.88 ms	0.63 ms
⌘ WithStyles(CardMedia) [mount]	Measure	1,593.9 ms	0.11 ms
⌘ WithStyles(CardContent) [mount]	Measure	1,593.96 ms	0.14 ms
⌘ CardContent [mount]	Measure	1,593.98 ms	0.12 ms
⌘ WithStyles(Typography) [mount]	Measure	1,594.07 ms	0.04 ms
⌘ WithStyles(Typography) [mount]	Measure	1,594.07 ms	0.04 ms
⌘ Typography [mount]	Measure	1,594.09 ms	0.01 ms
⌘ Typography [mount]	Measure	1,594.09 ms	0.01 ms
⌘ WithStyles(CardActions) [mount]	Measure	1,594.15 ms	0.35 ms
⌘ CardActions [mount]	Measure	1,594.17 ms	0.33 ms
⌘ WithStyles(Button) [mount]	Measure	1,594.2 ms	0.3 ms
⌘ Button [mount]	Measure	1,594.23 ms	0.27 ms
⌘ WithStyles(ButtonBase) [mount]	Measure	1,594.25 ms	0.25 ms
⌘ ButtonBase [mount]	Measure	1,594.29 ms	0.21 ms
⌘ Link [mount]	Measure	1,594.32 ms	0.18 ms
⌘ WithStyles(TouchRipple) [mount]	Measure	1,594.37 ms	0.13 ms
⌘ TouchRipple [mount]	Measure	1,594.4 ms	0.1 ms
⌘ TransitionGroup [mount]	Measure	1,594.43 ms	0.07 ms
⌘ TransitionGroup.getChildContext	Measure	1,594.45 ms	0.05 ms
⌘ WithStyles(MyCard) [mount]	Measure	1,594.59 ms	0.81 ms

Name	Type	Start Time	Duration
⌘ MyCard [mount]	Measure	1,594.62 ms	0.78 ms
⌘ WithStyles(Card) [mount]	Measure	1,594.66 ms	0.64 ms
⌘ Card [mount]	Measure	1,594.68 ms	0.62 ms
⌘ WithStyles(Paper) [mount]	Measure	1,594.69 ms	0.61 ms
⌘ Paper [mount]	Measure	1,594.71 ms	0.49 ms
⌘ WithStyles(CardMedia) [mount]	Measure	1,594.73 ms	0.07 ms
⌘ CardMedia [mount]	Measure	1,594.74 ms	0.06 ms
⌘ WithStyles(CardContent) [mount]	Measure	1,594.79 ms	0.11 ms
⌘ CardContent [mount]	Measure	1,594.81 ms	0.09 ms
⌘ Typography [mount]	Measure	1,594.85 ms	0.05 ms
⌘ WithStyles(Typography) [mount]	Measure	1,594.89 ms	0.01 ms
⌘ WithStyles(Typography) [mount]	Measure	1,594.89 ms	0.01 ms
⌘ WithStyles(CardActions) [mount]	Measure	1,594.96 ms	0.24 ms
⌘ CardActions [mount]	Measure	1,594.98 ms	0.22 ms
⌘ WithStyles(Button) [mount]	Measure	1,595.01 ms	0.2 ms
⌘ Button [mount]	Measure	1,595.04 ms	0.16 ms
⌘ WithStyles(ButtonBase) [mount]	Measure	1,595.06 ms	0.14 ms
⌘ ButtonBase [mount]	Measure	1,595.08 ms	0.12 ms
⌘ Link [mount]	Measure	1,595.12 ms	0.09 ms
⌘ WithStyles(TouchRipple) [mount]	Measure	1,595.16 ms	0.04 ms
⌘ TouchRipple [mount]	Measure	1,595.18 ms	0.02 ms
⌘ Route [mount]	Measure	1,595.37 ms	0.03 ms

Name	Type	Start Time	Duration
⌘ Route [mount]	Measure	1,595.37 ms	0.13 ms
⌘ Route.componentWillMount	Measure	1,595.48 ms	0.02 ms
⌘ Route.getChildContext	Measure	1,595.49 ms	0.01 ms
⌘ Route [mount]	Measure	1,595.57 ms	0.03 ms
⌘ Route [mount]	Measure	1,595.62 ms	0.08 ms
⌘ WithStyles(Typography) [mount]	Measure	1,595.66 ms	0.04 ms
⌘ Typography [mount]	Measure	1,595.69 ms	0.01 ms
⌘ (Committing Changes)	Measure	1,595.93 ms	1.07 ms
⌘ (Committing Snapshot Effects: 0 Total)	Measure	1,596 ms	0.2 ms
⌘ (Committing Host Effects: 66 Total)	Measure	1,596.21 ms	0.09 ms
⌘ (Calling Lifecycle Methods: 65 Total)	Measure	1,596.45 ms	0.55 ms
⌘ WithStyles(Typography).componentDidMount	Measure	1,596.55 ms	0.05 ms
⌘ WithStyles(Toolbar).componentDidMount	Measure	1,596.59 ms	0.01 ms
⌘ WithStyles(TouchRipple).componentDidMount	Measure	1,596.69 ms	0.01 ms
⌘ ButtonBase.componentDidMount	Measure	1,596.7 ms	0.1 ms
⌘ WithStyles(Button).componentDidMount	Measure	1,596.85 ms	0.05 ms
⌘ WithStyles(CardActions).componentDidMount	Measure	1,596.86 ms	0.04 ms
⌘ WithStyles(Card).componentDidMount	Measure	1,596.87 ms	0.03 ms
⌘ WithStyles(MyCard).componentDidMount	Measure	1,596.88 ms	0.02 ms
⌘ (React Tree Reconciliation)	Mark	80.27 ms	
⌘ App [mount] (#4)	Mark	80.82 ms	
⌘ BrowserRouter [mount] (#5)	Mark	1,581.1 ms	

Name	Type	Start Time	Duration
⌘ BrowserRouter.componentWillMount (#5)	Mark	1,581.42 ms	
⌘ Router [mount] (#6)	Mark	1,581.52 ms	
⌘ Router.componentWillMount (#6)	Mark	1,581.6 ms	
⌘ Router.getChildContext (#6)	Mark	1,581.8 ms	
⌘ MuiThemeProvider [mount] (#7)	Mark	1,581.83 ms	
⌘ MuiThemeProvider.getChildContext (#7)	Mark	1,581.96 ms	
⌘ WithStyles(Header) [mount] (#9)	Mark	1,582.2 ms	
⌘ Header [mount] (#16)	Mark	1,583.57 ms	
⌘ WithStyles(AppBar) [mount] (#18)	Mark	1,583.67 ms	
⌘ AppBar [mount] (#19)	Mark	1,584.11 ms	
⌘ WithStyles(Paper) [mount] (#20)	Mark	1,584.28 ms	
⌘ Paper [mount] (#21)	Mark	1,584.72 ms	
⌘ WithStyles(Toolbar) [mount] (#23)	Mark	1,584.77 ms	
⌘ Toolbar [mount] (#24)	Mark	1,585.11 ms	
⌘ WithStyles(Typography) [mount] (#26)	Mark	1,585.15 ms	
⌘ Typography [mount] (#27)	Mark	1,585.84 ms	
⌘ Route [mount] (#10)	Mark	1,587 ms	
⌘ Route.componentWillMount (#10)	Mark	1,587.36 ms	
⌘ Route.getChildContext (#10)	Mark	1,587.43 ms	
⌘ WithStyles(Home) [mount] (#29)	Mark	1,587.47 ms	
⌘ Home [mount] (#30)	Mark	1,587.63 ms	
⌘ WithStyles(MyCard) [mount] (#32)	Mark	1,587.76 ms	

Name	Type	Start Time	Duration
✱ MyCard [mount] (#36)	Mark	1,587.9 ms	
✱ WithStyles(Card) [mount] (#38)	Mark	1,588.04 ms	
✱ Card [mount] (#39)	Mark	1,588.13 ms	
✱ WithStyles(Paper) [mount] (#40)	Mark	1,588.16 ms	
✱ Paper [mount] (#41)	Mark	1,588.2 ms	
✱ WithStyles(CardMedia) [mount] (#43)	Mark	1,588.22 ms	
✱ CardMedia [mount] (#46)	Mark	1,588.34 ms	
✱ WithStyles(CardContent) [mount] (#44)	Mark	1,588.5 ms	
✱ CardContent [mount] (#48)	Mark	1,588.74 ms	
✱ WithStyles(Typography) [mount] (#50)	Mark	1,588.78 ms	
✱ Typography [mount] (#52)	Mark	1,588.82 ms	
✱ WithStyles(Typography) [mount] (#51)	Mark	1,588.87 ms	
✱ Typography [mount] (#54)	Mark	1,588.9 ms	
✱ WithStyles(CardActions) [mount] (#45)	Mark	1,588.98 ms	
✱ CardActions [mount] (#56)	Mark	1,589.14 ms	
✱ WithStyles(Button) [mount] (#58)	Mark	1,589.32 ms	
✱ Button [mount] (#59)	Mark	1,590.35 ms	
✱ WithStyles(ButtonBase) [mount] (#60)	Mark	1,590.52 ms	
✱ ButtonBase [mount] (#61)	Mark	1,590.88 ms	
✱ Link [mount] (#62)	Mark	1,591.19 ms	
✱ WithStyles(TouchRipple) [mount] (#65)	Mark	1,591.46 ms	
✱ TouchRipple [mount] (#66)	Mark	1,592.03 ms	

Name	Type	Start Time	Duration
✱ TransitionGroup [mount] (#67)	Mark	1,592.17 ms	
✱ TransitionGroup.getChildContext (#67)	Mark	1,592.31 ms	
✱ WithStyles(MyCard) [mount] (#33)	Mark	1,592.76 ms	
✱ MyCard [mount] (#69)	Mark	1,592.8 ms	
✱ WithStyles(Card) [mount] (#71)	Mark	1,592.85 ms	
✱ Card [mount] (#72)	Mark	1,592.88 ms	
✱ WithStyles(Paper) [mount] (#73)	Mark	1,592.9 ms	
✱ Paper [mount] (#74)	Mark	1,592.93 ms	
✱ WithStyles(CardMedia) [mount] (#76)	Mark	1,592.95 ms	
✱ CardMedia [mount] (#79)	Mark	1,592.98 ms	
✱ WithStyles(CardContent) [mount] (#77)	Mark	1,593.06 ms	
✱ CardContent [mount] (#81)	Mark	1,593.08 ms	
✱ WithStyles(Typography) [mount] (#83)	Mark	1,593.1 ms	
✱ Typography [mount] (#85)	Mark	1,593.13 ms	
✱ WithStyles(Typography) [mount] (#84)	Mark	1,593.19 ms	
✱ Typography [mount] (#87)	Mark	1,593.21 ms	
✱ WithStyles(CardActions) [mount] (#78)	Mark	1,593.27 ms	
✱ CardActions [mount] (#89)	Mark	1,593.3 ms	
✱ WithStyles(Button) [mount] (#91)	Mark	1,593.33 ms	
✱ Button [mount] (#92)	Mark	1,593.37 ms	
✱ WithStyles(ButtonBase) [mount] (#93)	Mark	1,593.39 ms	
✱ ButtonBase [mount] (#94)	Mark	1,593.43 ms	

Name	Type	Start Time	Duration
⌘ Link [mount] (#95)	Mark	1,593.47 ms	
⌘ WithStyles(TouchRipple) [mount] (#98)	Mark	1,593.53 ms	
⌘ TouchRipple [mount] (#99)	Mark	1,593.56 ms	
⌘ TransitionGroup [mount] (#100)	Mark	1,593.59 ms	
⌘ TransitionGroup.getChildContext (#100)	Mark	1,593.61 ms	
⌘ WithStyles(MyCard) [mount] (#34)	Mark	1,593.76 ms	
⌘ MyCard [mount] (#102)	Mark	1,593.78 ms	
⌘ WithStyles(Card) [mount] (#104)	Mark	1,593.82 ms	
⌘ Card [mount] (#105)	Mark	1,593.84 ms	
⌘ WithStyles(Paper) [mount] (#106)	Mark	1,593.85 ms	
⌘ Paper [mount] (#107)	Mark	1,593.88 ms	
⌘ WithStyles(CardMedia) [mount] (#109)	Mark	1,593.9 ms	
⌘ CardMedia [mount] (#112)	Mark	1,593.92 ms	
⌘ WithStyles(CardContent) [mount] (#110)	Mark	1,593.96 ms	
⌘ CardContent [mount] (#114)	Mark	1,593.98 ms	
⌘ WithStyles(Typography) [mount] (#116)	Mark	1,594 ms	
⌘ Typography [mount] (#118)	Mark	1,594.02 ms	
⌘ WithStyles(Typography) [mount] (#117)	Mark	1,594.07 ms	
⌘ Typography [mount] (#120)	Mark	1,594.1 ms	
⌘ WithStyles(CardActions) [mount] (#111)	Mark	1,594.15 ms	
⌘ CardActions [mount] (#122)	Mark	1,594.17 ms	
⌘ WithStyles(Button) [mount] (#124)	Mark	1,594.2 ms	

Name	Type	Start Time	Duration
✱ Button [mount] (#125)	Mark	1,594.23 ms	
✱ WithStyles(ButtonBase) [mount] (#126)	Mark	1,594.25 ms	
✱ ButtonBase [mount] (#127)	Mark	1,594.29 ms	
✱ Link [mount] (#128)	Mark	1,594.32 ms	
✱ WithStyles(TouchRipple) [mount] (#131)	Mark	1,594.38 ms	
✱ TouchRipple [mount] (#132)	Mark	1,594.4 ms	
✱ TransitionGroup [mount] (#133)	Mark	1,594.43 ms	
✱ TransitionGroup.getChildContext (#133)	Mark	1,594.45 ms	
✱ WithStyles(MyCard) [mount] (#35)	Mark	1,594.59 ms	
✱ MyCard [mount] (#135)	Mark	1,594.62 ms	
✱ WithStyles(Card) [mount] (#137)	Mark	1,594.66 ms	
✱ Card [mount] (#138)	Mark	1,594.68 ms	
✱ WithStyles(Paper) [mount] (#139)	Mark	1,594.69 ms	
✱ Paper [mount] (#140)	Mark	1,594.71 ms	
✱ WithStyles(CardMedia) [mount] (#142)	Mark	1,594.73 ms	
✱ CardMedia [mount] (#145)	Mark	1,594.75 ms	
✱ WithStyles(CardContent) [mount] (#143)	Mark	1,594.79 ms	
✱ CardContent [mount] (#147)	Mark	1,594.81 ms	
✱ WithStyles(Typography) [mount] (#149)	Mark	1,594.83 ms	
✱ Typography [mount] (#151)	Mark	1,594.85 ms	
✱ WithStyles(Typography) [mount] (#150)	Mark	1,594.89 ms	
✱ Typography [mount] (#153)	Mark	1,594.91 ms	

Name	Type	Start Time	Duration
⌘ WithStyles(CardActions) [mount] (#144)	Mark	1,594.96 ms	
⌘ CardActions [mount] (#155)	Mark	1,594.98 ms	
⌘ WithStyles(Button) [mount] (#157)	Mark	1,595.01 ms	
⌘ Button [mount] (#158)	Mark	1,595.04 ms	
⌘ WithStyles(ButtonBase) [mount] (#159)	Mark	1,595.06 ms	
⌘ ButtonBase [mount] (#160)	Mark	1,595.09 ms	
⌘ Link [mount] (#161)	Mark	1,595.12 ms	
⌘ WithStyles(TouchRipple) [mount] (#164)	Mark	1,595.16 ms	
⌘ TouchRipple [mount] (#165)	Mark	1,595.18 ms	
⌘ TransitionGroup [mount] (#166)	Mark	1,595.2 ms	
⌘ TransitionGroup.getChildContext (#166)	Mark	1,595.22 ms	
⌘ Route [mount] (#11)	Mark	1,595.37 ms	
⌘ Route.componentWillMount (#11)	Mark	1,595.48 ms	
⌘ Route.getChildContext (#11)	Mark	1,595.5 ms	
⌘ Route [mount] (#12)	Mark	1,595.51 ms	
⌘ Route.componentWillMount (#12)	Mark	1,595.54 ms	
⌘ Route.getChildContext (#12)	Mark	1,595.56 ms	
⌘ Route [mount] (#13)	Mark	1,595.57 ms	
⌘ Route.componentWillMount (#13)	Mark	1,595.6 ms	
⌘ Route.getChildContext (#13)	Mark	1,595.61 ms	
⌘ Route [mount] (#14)	Mark	1,595.62 ms	
⌘ Route.componentWillMount (#14)	Mark	1,595.64 ms	

Name	Type	Start Time	Duration
✱ Route.getChildContext (#14)	Mark	1,595.65 ms	
✱ WithStyles(Typography) [mount] (#15)	Mark	1,595.66 ms	
✱ Typography [mount] (#168)	Mark	1,595.69 ms	
✱ (Committing Changes)	Mark	1,595.93 ms	
✱ (Committing Snapshot Effects)	Mark	1,596 ms	
✱ (Committing Host Effects)	Mark	1,596.22 ms	
✱ (Calling Lifecycle Methods)	Mark	1,596.45 ms	
✱ WithStyles(Typography).componentDidMount (#26)	Mark	1,596.55 ms	
✱ WithStyles(Toolbar).componentDidMount (#23)	Mark	1,596.59 ms	
✱ WithStyles(Paper).componentDidMount (#20)	Mark	1,596.6 ms	
✱ WithStyles(AppBar).componentDidMount (#18)	Mark	1,596.61 ms	
✱ WithStyles(Header).componentDidMount (#9)	Mark	1,596.62 ms	
✱ WithStyles(CardMedia).componentDidMount (#43)	Mark	1,596.63 ms	
✱ WithStyles(CardContent).componentDidMount (#44)	Mark	1,596.65 ms	
✱ TransitionGroup.componentDidMount (#67)	Mark	1,596.65 ms	
✱ WithStyles(TouchRipple).componentDidMount (#65)	Mark	1,596.69 ms	
✱ ButtonBase.componentDidMount (#61)	Mark	1,596.7 ms	
✱ WithStyles(ButtonBase).componentDidMount (#60)	Mark	1,596.85 ms	
✱ WithStyles(Button).componentDidMount (#58)	Mark	1,596.85 ms	
✱ WithStyles(CardActions).componentDidMount (#45)	Mark	1,596.86 ms	
✱ WithStyles(Card).componentDidMount (#38)	Mark	1,596.87 ms	
✱ WithStyles(MyCard).componentDidMount (#32)	Mark	1,596.88 ms	

Name	Type	Start Time	Duration
⌘ WithStyles(Home).componentDidMount (#29)	Mark	1,596.93 ms	
⌘ MuiThemeProvider.componentDidMount (#7)	Mark	1,596.94 ms	
@grammarly-extension:checkScriptInitStart	Mark	1,838.18 ms	
@grammarly-extension:checkScriptInitEnd	Mark	1,841.39 ms	

Keep request counts low and transfer sizes small — 10 requests • 7,606 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	10	7,605.9 KiB
Image	4	5,514.3 KiB
Script	4	2,088.0 KiB
Stylesheet	1	2.8 KiB
Document	1	0.8 KiB
Media	0	0.0 KiB
Font	0	0.0 KiB
Other	0	0.0 KiB
Third-party	6	5,527.7 KiB

Largest contentful paint element — 1 element found

This is the largest contentful element painted within the viewport. [Learn more](#) LCP

Element

div.MuiCardMedia-root-69.MyCard-media-67

Element

Avoid long main-thread tasks — 4 long tasks found

Lists the longest tasks on the main thread –useful for identifying worst contributors to input delay. [Learn more](#) TBT

URL	Start Time	Duration
/libs/jquery.js (localhost)	23,265 ms	6,185 ms
http://localhost:1234	828 ms	156 ms
chrome-extension://bnjjngeaknajbdcgpfkgnonkmififhfo/build/content-script.js	680 ms	108 ms
http://localhost:1234	604 ms	71 ms

More information about the performance of your application. These numbers don't [directly affect](#) the performance score.

PASSED AUDITS (24)

Hide

Properly size images

Serve images that are appropriately-sized to save mobile data and improve load time. [Learn more](#).

Defer off-screen images

Consider lazy loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more](#).

Minify CSS

Minifying CSS files can reduce network payload sizes. [Learn more](#). FCP LCP



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](#).

Reduce unused CSS

Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn more](#). FCP LCP

Pre-connect to required origins

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

Initial server response time was short — Root document took 0 ms

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



If you are server-side rendering any React components, consider using `renderToNodeStream()` or `renderToStaticNodeStream()` to allow the client to receive and hydrate different parts of the markup instead of all at once. [Learn more.](#)

URL	Time Spent
http://localhost:1234	0 ms

Avoid multiple page redirects

Redirects introduce additional delays before the page can be loaded. [Learn more.](#) FCP LCP



If you are using React Router, minimise usage of the `<Redirect>` component for [route navigations](#).

○ Pre-load key requests

Consider using `<link rel=preload>` to prioritise fetching resources that are currently requested later in page load. [Learn more.](#) FCP LCP

Use HTTP/2

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

Use video formats for animated content

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more](#) LCP

Remove duplicate modules in JavaScript bundles

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

Avoid serving legacy JavaScript to modern browsers — Potential savings of 1 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 delivered to modern browsers, while retaining support for legacy browsers.

[Learn more](#) TBT

URL	Potential savings
/bundle.js (localhost)	0.6 KiB
bundle.js:138	@babel/plugin-transform-classes

Preload largest contentful paint image

Preload the image used by the LCP element in order to improve your LCP time. [Learn more](#). LCP

	URL	Potential savings
	div.MuiCardMedia-root-69.MyCard-media-67 ...big/fesh.jpg (storage.googleapis.com)	0 ms

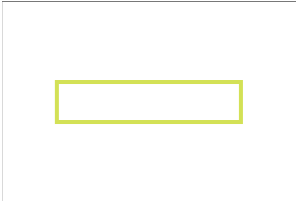
Avoids an excessive DOM size — 49 elements

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

TBT



Consider using a 'windowing' library, like `react-window`, to minimise the number of DOM nodes created if you are rendering many repeated elements on the page. [Learn more](#). Also, minimise 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		49
Maximum DOM Depth	 span.MuiButton-label-75	9

Statistic	Element	Value
	<div>main.Home-root-64</div>	
Maximum Child Elements		4

All text remains visible during webfont loads

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading. [Learn more.](#) FCP LCP

Minimise 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.](#) TBT

Third-party	Transfer size	Main-thread blocking time
Other Google APIs/SDKs	5,514 KiB	0 ms
...big/soop.jpg (storage.googleapis.com)	2,295 KiB	0 ms
...big/fesh.jpg (storage.googleapis.com)	2,245 KiB	0 ms
...big/pezza.jpg (storage.googleapis.com)	813 KiB	0 ms

☐ Lazy load third-party resources with facades

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn more.](#) TBT

Largest contentful paint image was not lazily loaded

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more.](#)

Element
<div>div.MuiCardMedia-root-69.MyCard-media-67</div>

☐ Avoid large layout shifts



These DOM elements contribute most to the CLS of the page. 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](#).

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 poor, slow and increase CLS. [Learn more](#) CLS

☐ Image elements have explicit `width` and `height`



Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn more](#) CLS

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



A ``<meta name="viewport">`` not only optimises your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT



Captured at 26 Mar 2023,
08:49 CEST



Emulated Moto G4 with
Lighthouse 9.6.8



Single page load



Initial page load



Slow 4G throttling



Using Chromium 110.0.0.0
with devtools

Generated by **Lighthouse 9.6.8** | [File an issue](#)