

Core Web Vitals: Brindle Lane Plumbing (Lovable build)

brindle-lane-plumbing-leeds.lovable.app | Google PageSpeed Insights, Lighthouse 13.5.0, captured 27 Sep 2026 20:44 UTC | lab data only, no field data (site is too new for CrUX)

Metrics

Metric	Mobile (Moto G Power, slow 4G)	Desktop	Google threshold (good)	Read
Largest Contentful Paint (LCP)	2.9 s, orange	0.7 s, green	under 2.5 s	Mobile misses by 0.4 s. The LCP element is the H1 text, and it waits on render-blocking requests
Cumulative Layout Shift (CLS)	0.045, green	0, green	under 0.1	Pass. Small shift on mobile, likely the web font swapping in
Total Blocking Time (TBT)	0 ms, green	0 ms, green	under 200 ms	Pass. Almost no JavaScript work on the main thread
First Contentful Paint (FCP)	2.9 s, orange	0.7 s, green	under 1.8 s	Same cause as LCP: nothing paints until the blocking CSS and fonts arrive
Speed Index	2.9 s, green	0.8 s, green	under 3.4 s	Pass, just
Performance score	90	99	90+	Mobile is on the line. One more blocking request and it drops to the 80s

What is holding mobile LCP back

FCP and LCP are the same number, 2.9 s, on mobile. That means the page paints nothing at all until 2.9 s and then everything at once. Lighthouse points at one thing: **render-blocking requests, estimated 960 ms of savings**. On slow 4G the browser cannot draw the hero until it has downloaded the CSS bundle and the fonts. TBT is 0 ms, so JavaScript is not the problem. This is a loading-order problem, not a code problem.

Secondary items from the report: inefficient cache lifetimes (97 KiB), unused JavaScript (104 KiB), and a layout-shift culprit. None of these move LCP much on their own.

Fixes, in order of LCP impact

#	Fix	Effect
1	Inline the critical CSS for the hero and defer the rest. Lovable ships one CSS bundle that blocks first paint	Biggest lever. Most of the 960 ms
2	Preload the two hero fonts and set font-display: swap (or optional)	Cuts the remaining paint delay and the small CLS
3	Preconnect to the font and asset hosts	Saves one round trip on slow 4G, roughly 100-300 ms
4	Longer cache lifetimes on static assets	Repeat visits only, no effect on first-load LCP
5	Tree-shake the 104 KiB of unused JS	TBT is already 0, so this is hygiene, not LCP

Target: mobile LCP under 2.5 s, which needs about 400 ms off the current number. Fix 1 alone should do it. Note that these are Lovable's build settings, not the page content; a Lovable user cannot change the CSS loading strategy from the editor.

Mobile: metrics

METRICSExpand view

■ First Contentful Paint

2.9 s

● Total Blocking Time

0 ms

● Speed Index

2.9 s

■ Largest Contentful Paint

2.9 s

● Cumulative Layout Shift

0.045

📅 Captured at Sep 27, 2026, 8:44 PM UTC

📱 Emulated Moto G Power with Lighthouse 13.5.0

🔗 Single page session

🕒 Initial page load

🌪️ Slow 4G throttling

🖥️ Using HeadlessChromium 153.0.8010.36 with lr

🗃️ View Treemap

Desktop: metrics

METRICSExpand view

● First Contentful Paint

0.7 s

● Total Blocking Time

0 ms

● Speed Index

0.8 s

● Largest Contentful Paint

0.7 s

● Cumulative Layout Shift

0

📅 Captured at Sep 27, 2026, 8:44 PM UTC

📱 Emulated Desktop with Lighthouse 13.5.0

🔗 Single page session

🕒 Initial page load

🌪️ Custom throttling

🖥️ Using HeadlessChromium 153.0.8010.36 with lr

🗃️ View Treemap

Mobile: full Performance category (score, metrics, insights, diagnostics)

its

Mobile

Desktop

90

Performance

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

▲ 0-49

■ 50-89

● 90-100

Leaks, boilers & blocked drains — sorted fast.

Emergency call out

Request a quote

Fastly on site 24/7

24/7 emergency call out

No call charges — just the fix

METRICS

Expand view

■ First Contentful Paint

2.9 s

● Total Blocking Time

0 ms

● Speed Index

2.9 s

■ Largest Contentful Paint

2.9 s

● Cumulative Layout Shift

0.045

Captured at Sep 27, 2026, 8:44 PM UTC

Emulated Moto G Power with Lighthouse 13.5.0

Single page session

Initial page load

Slow 4G throttling

Using HeadlessChromium 153.0.8010.36 with Ir

View Treemap

Show audits relevant to: All FCP LCP CLS

INSIGHTS

▲ Render-blocking requests — Est savings of 960 ms

▲ Network dependency tree

■ Use efficient cache lifetimes — Est savings of 97 KiB

○ Layout shift culprits

○ LCP breakdown

○ 3rd parties

These insights are also available in the Chrome DevTools Performance Panel - record a trace to view more detailed information.

DIAGNOSTICS

▲ Reduce unused JavaScript — Est savings of 104 KiB

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

PASSED AUDITS (20)

Show

Desktop: full Performance category

its

