

PERFORMANCE (LightHouse) CodeBloggs vs FaceBook – HOME PAGE

CodeBloggs

Performance rating: 66/100

Metrics:

- First Contentful Paint: 0.5 s
- Largest Contentful Paint: 2.6 s
- Total Blocking Time: 440 ms
- Cumulative Layout Shift: 0.004
- Speed Index: 0.8 s

Diagnostic:

- [!] Largest Contentful Paint element: 2,620 ms
- [!] Reduce unused JavaScript: Potential savings of 946 KiB
- [!] Minify JavaScript: Potential savings of 431 KiB
- [!] Preconnect to required origins: Potential savings of 100 ms
- [!] Preload Largest Contentful Paint image: Potential savings of 50 ms
- [!] Properly size images: Potential savings of 10 KiB
- Minify CSS: Potential savings of 403 KiB
- Serve static assets with an efficient cache policy: 2 resources found
- Reduce unused CSS: Potential savings of 483 KiB
- Avoid serving legacy JavaScript to modern browsers: Potential savings of 0 KiB
- Page prevented back/forward cache restoration: 1 failure reason

Passed audits: 19

Facebook

Performance rating: 44/100

Metrics:

- First Contentful Paint: 1.3 s
- Largest Contentful Paint: 2.7 s
- Total Blocking Time: 1,550 ms
- Cumulative Layout Shift: 0.003
- Speed Index: 3.4 s

Diagnostic:

- [!] Reduce JavaScript execution time: 6.4 s
- [!] Minimize main-thread work: 8.7 s
- [!] Largest Contentful Paint element: 2,710 ms
- [!] Reduce unused JavaScript: Potential savings of 1,350 KiB
- [!] Eliminate render-blocking resources: Potential savings of 130 ms
- [!] Reduce unused CSS: Potential savings of 129 KiB
- [!] Serve images in next-gen formats: Potential savings of 112 KiB
- [!] Properly size images: Potential savings of 118 KiB
- [!] Avoid an excessive DOM size: 3,818 elements
- Minify JavaScript: Potential savings of 69 KiB
- Enable text compression: Potential savings of 363 KiB
- Serve static assets with an efficient cache policy: 35 resources found
- Image elements do not have explicit width and height
- Avoid serving legacy JavaScript to modern browsers: Potential savings of 7 KiB
- Page prevented back/forward cache restoration: 9 failure reasons
- Avoid enormous network payloads: Total size was 5,013 KiB

Passed audits: 17

Comparisons:

- CodeBloggs achieves a rapid First Contentful Paint at 0.5 s, significantly outperforming Facebook's 1.3 s, showcasing quicker initial rendering.
- CodeBloggs maintains a slightly lower Largest Contentful Paint element time at 2.6 s (Element: 2,620 ms) compared to Facebook's 2.7 s (Element: 2,710 ms), contributing to a better user experience.
- CodeBloggs impressively demonstrates significantly lower Total Blocking Time at 440 ms, ensuring smoother interactivity during page load, compared to Facebook's 1,550 ms.
- While both sites exhibit minimal layout shifts, Facebook has a slightly lower Cumulative Layout Shift (0.003) compared to CodeBloggs (0.004).
- CodeBloggs achieves a much faster Speed Index at 0.8 s, providing a more responsive and user-friendly experience, in contrast to Facebook's 3.4 s.
- CodeBloggs has 6 flagged issues (e.g., Largest Contentful Paint element, Reduce unused JavaScript), while Facebook has 9 flagged issues (e.g., Reduce JavaScript execution time, Avoid enormous network payloads), with both sites needing performance improvements

Conclusion:

CodeBloggs stands out as a platform with superior performance, particularly evident in key metrics like First Contentful Paint and Total Blocking Time when compared to Facebook. This distinction results in a homepage that is not only more efficient and faster-loading but also indicative of CodeBloggs' dedication to providing users with an optimized and responsive web experience. Despite both platforms passing an equal number of audits, the detailed analysis highlights CodeBloggs' commitment to speed and user-centric design, solidifying its position as a leader in delivering a seamless online interaction.