Nigerian Journal of Engineering Science Research (NIJESR).
Copyright@ Department of Mechanical Engineering, Gen. Abdusalami Abubakar College of Engineering, Igbinedion University, Okada, Edo State, Nigeria.
ISSN: 2636-7114
Journal Homepage: https://nijesr.iuokada.edu.ng/

Analysis of Website Performance Optimisation Techniques for Mobile Users in Low-Bandwidth Environments

[1*]Abubakar Atiku Muslim and [2]Muhammad Saidu Aliero

[1]*Department of Electrical and Electronics Engineering, Abdullahi Fodio University of Science and Technology, Aliero, Kebbi State, Nigeria*
atiku.muslim@ksusta.edu.ng
[2]*Department of Information and Communication Technology, Abdullahi Fodio University of Science and Technology, Aliero, Kebbi State, Nigeria*
msaidua2000@ksusta.edu.ng

**Corresponding Author: Abubakar Atiku Muslim; atiku.muslim@ksusta.edu.ng*

Manuscript History
Received: 03/07/2026
Revised: 26/07/2026
Accepted: 05/08/2026
Published: 13/08/2026
https://doi.org/10.5281/zenodo.21917690

Abstract: *The proliferation of mobile internet access in sub-Saharan Africa has made website performance optimisation a matter of critical engineering importance. Mobile users on constrained networks face severe delays from oversized images, weak caching policies and unoptimised front-end code. This study experimentally evaluates three widely adopted front-end optimisation methods, image compression, browser caching and code minification, across ten representative Nigerian websites. Evaluation used Google Lighthouse and PageSpeed Insights under two network conditions: Long-Term Evolution (LTE) and Slow Fourth-Generation (Slow 4G) throttling. Recorded metrics were page load time, total page size, number of resource requests and the composite Lighthouse performance score. Image compression reduced average page size by 38.1%; browser caching reduced repeat-visit load time by 43.3%; and code minification improved resource efficiency by 18.3%. Combined application of all three techniques achieved a mean load-time reduction of 50.5% and a Lighthouse score improvement of 28.1 percentage points, raising average scores from 54.9 to 83.0. Paired-sample t-tests confirmed that all improvements were statistically significant ($p < 0.001$), with very large effect sizes. These findings demonstrate that an integrated, multi-technique optimisation strategy delivers measurably superior outcomes and strongly supports its systematic adoption by developers serving bandwidth-constrained mobile users across Nigeria and sub-Saharan Africa. Further studies should extend this framework to server-side interventions, including Content Delivery Network deployment, Hypertext Transfer Protocol version 3 adoption and Brotli compression, and should incorporate real-user monitoring across diverse geographic and network contexts to validate the synthetic findings reported here.*

Keywords: *Website Performance, Mobile Optimisation, Image Compression, Browser Caching, Code Minification, Web Performance Score*

INTRODUCTION

The global shift towards mobile-first internet access has fundamentally altered the engineering requirements of web development. Mobile devices now account for more than 60% of all internet traffic worldwide, a proportion that is considerably higher across Africa, where smartphones frequently represent the sole means of internet access for a large segment of the population (International Telecommunication Union, 2023; Statista, 2024). In Nigeria, the Nigerian Communications Commission (2023) reported over 157 million active mobile internet subscriptions, underscoring national dependence on mobile connectivity for commerce, education, healthcare delivery and civic participation. Despite this trajectory, websites serving African mobile users remain markedly under-optimised. Average mobile broadband download speeds in Nigeria range between 10 and 25 Mbps, well below global averages, while latency is compounded by congestion and intermittent coverage (Ookla, 2024). High server latency and extended page load times cause significant user abandonment

Abubakar and Muhammad (2026). Analysis of Website Performance Optimisation Techniques for Mobile Users in Low-Bandwidth Environments. *Nigeria Journal of Engineering Science Research (NIJESR)*, 9(2), 72-84. https://doi.org/10.5281/zenodo.21917690

and reduce conversion rates (Bai *et al.*, 2017; Basalla *et al.*, 2021). Industry benchmarks report that 53% of mobile users abandon websites requiring more than three seconds to load, and each additional second corresponds to a 7% reduction in conversion rates (Google, 2018; Portent, 2023). For Nigerian e-government portals, digital commerce platforms and distance-learning systems, these penalties translate into measurable reductions in service uptake and economic productivity. The engineering causes of poor mobile web performance are well documented. Oversized image assets can constitute 60 to 70% of total page payload on unoptimised sites (HTTP Archive, 2024). The absence of browser caching policies forces repeated retrieval of static resources on every page visit, consuming bandwidth unnecessarily (Barford *et al.*, 1999). Unminified Cascading Style Sheets (CSS) and JavaScript (JS) files contain redundant whitespace, comments and formatting characters that increase transfer size without contributing to functionality (Grigorik, 2019; Souders, 2009; Malavolta *et al.*, 2023).

Collectively, these deficiencies impose disproportionate costs on users in low-bandwidth environments. Optimisation techniques targeting these causes have been studied extensively in high-bandwidth contexts. However, their effectiveness under the infrastructure constraints typical of sub-Saharan African mobile networks remains insufficiently documented (Abara *et al.*, 2022; Chukwu and Okafor, 2023). Most existing studies also evaluate techniques in isolation rather than examining the compounding benefits of combined application, and the inclusion of Lighthouse-based Core Web Vitals assessment and network throttling simulation remains limited in African-focused studies. The evolution of performance measurement has introduced constructs that extend beyond traditional load-time metrics. Google's Core Web Vitals framework, now a search ranking signal, emphasises Largest Contentful Paint, Total Blocking Time (approximated in the field by Interaction to Next Paint since 2024 and Cumulative Layout Shift (Walton, 2024; Wagner and Viscomi, 2024). Web performance is directly linked to Search Engine Optimisation outcomes (Sellamuthu *et al.*, 2022; Eslamdoust, 2022). Nabi and Jhanjhi (2021) showed that minification and asset compression have differential effects: image compression most strongly reduces Largest Contentful Paint, while script minification predominantly improves Total Blocking Time. At the protocol level, Hypertext Transfer Protocol version 3 (HTTP/3), based on the QUIC transport protocol, reduces connection-establishment latency in high-packet-loss environments (Vogel and Springer, 2023a; World Wide Web Consortium, 2023), while Brotli server-side compression achieves 15 to 25% smaller transfer sizes than GZIP for text resources (Alakuijala *et al.*, 2015). These server-side advances complement the front-end techniques evaluated here. The aim of this study is to quantitatively evaluate the effectiveness of image compression, browser caching and code minification, applied individually and in combination, in improving mobile website performance. The specific objectives are:

i. To establish pre-optimisation performance benchmarks across ten websites using Google Lighthouse under standardised mobile simulation conditions.
ii. To quantify the impact of image compression on page size and load time.
iii. To assess the effect of browser caching on repeat-visit load time and data consumption.
iv. To determine the contribution of code minification to resource efficiency and rendering speed.
v. To evaluate the compounding performance gains of combined optimisation.
vi. To assess performance improvements under both LTE and Slow 4G network conditions.
vii. To validate observed improvements using paired-sample inferential statistics and effect-size analysis.

Beyond applying established optimisation techniques, this study makes three contributions that distinguish it from prior work. First, it is among the first to quantify the *compounding* effect of combining image compression, browser caching and code minification within a single controlled experiment on Nigerian web properties, rather than evaluating each technique in isolation as earlier studies have done. Second, it pairs modern Lighthouse Core Web Vitals scoring with explicit LTE and Slow 4G network-throttling conditions representative of Nigerian connectivity, a combination absent from existing African-focused literature. Third, it subjects all reported gains to formal inferential validation (paired-sample t-tests, Shapiro-Wilk normality screening, Wilcoxon confirmation and Cohen's d effect sizes), providing a level of statistical rigour rarely present in applied web-performance studies for this region.

MATERIALS AND METHODS

2.1 Research Design

Abubakar and Muhammad (2026). Analysis of Website Performance Optimisation Techniques for Mobile Users in Low-Bandwidth Environments. *Nigeria Journal of Engineering Science Research (NIJESR)*, 9(2), 72-84. https://doi.org/10.5281/zenodo.21917690

A before-and-after experimental design was adopted. Each intervention was applied sequentially and performance re-measured at each stage, with a final combined-optimisation condition assessing cumulative effects. This design isolates individual technique contributions while quantifying synergistic benefits. All tests were repeated ten times per condition and the median recorded to reduce measurement variance from network fluctuation.

2.2 Equipment and Software Tools

The following hardware and software were employed:

i. Mobile test device: Android smartphone, 6.5-inch display, 4 GB RAM, Android v16.
ii. Network conditions: live LTE connection (18 Mbps downlink); Lighthouse Slow 4G throttle (1.6 Mbps downlink, 150 ms round-trip time).
iii. Google Chrome browser (stable release) with Developer Tools.
iv. Google Lighthouse v13: automated composite scoring across First Contentful Paint, Largest Contentful Paint, Total Blocking Time, Cumulative Layout Shift and Speed Index.
v. Google PageSpeed Insights: field-data integration and origin summary.
vi. GTmetrix Pro: waterfall analysis, resource inventory and historical tracking.
vii. Squoosh (Google): lossy and lossless image compression with WebP conversion.
viii. PurifyCSS and UglifyJS v3: automated dead-code removal and minification.
ix. Apache HTTP Server 2.4: local hosting for controlled caching-header configuration.

2.3 Website Selection and Sampling

Ten publicly accessible websites were selected to improve generalisability, drawn from four categories: news and information portals (three sites), e-commerce platforms (three sites), government agency websites (two sites) and higher-education portals (two sites). Selection criteria were: (a) availability to Nigerian mobile users; (b) sufficient asset complexity to produce measurable optimisation effects; (c) no third-party Content Delivery Network (CDN) already in place, verified by examining response headers for CDN provider signatures using curl; and (d) no existing minification pipeline, confirmed by inspecting each page's CSS and JS for unminified whitespace and comments using Chrome Developer Tools. Sites failing any criterion were replaced from the same category. Sites are designated S1 through S10. Anonymisation was adopted because the sample includes government and educational portals whose administrators did not consent to named disclosure of performance (Pallant, 2020), and because naming sites risks confounding comparative findings with reputational interpretation. Categories and baseline metrics are given in Table-1; equivalent websites can be identified by applying the same criteria to publicly accessible Nigerian web properties.

Table-1 Test Website Categories and Baseline Complexity

Site ID	Category	Load Time (s)	Page Size (MB)	Requests	Score (%)
S1	News portal	4.8	5.2	92	58
S2	E-commerce	5.4	6.1	103	54
S3	News portal	4.5	4.8	87	61
S4	Government agency	6.2	7.3	118	48
S5	E-commerce	5.8	6.8	109	52
S6	Higher education	3.9	3.7	74	67
S7	Government agency	7.1	8.4	134	44
S8	E-commerce	5.0	5.6	97	56
S9	Higher education	4.3	4.4	83	63
S10	News portal	6.6	7.7	125	46
Mean	–	5.36	6.00	102.2	54.9

2.4 Performance Metrics

Abubakar and Muhammad (2026). Analysis of Website Performance Optimisation Techniques for Mobile Users in Low-Bandwidth Environments. *Nigeria Journal of Engineering Science Research (NIJESR)*, 9(2), 72-84. https://doi.org/10.5281/zenodo.21917690

Four metrics were recorded at each stage, consistent with World Wide Web Consortium (2023) recommendations:

i. Page Load Time (seconds): duration from initial request to DOMContentLoaded plus onLoad completion.
ii. Total Page Size (megabytes): aggregate uncompressed transfer size of all resources.
iii. Number of HTTP Requests: total individual resource requests per page load.
iv. Lighthouse Performance Score (0–100): composite index weighting First Contentful Paint (10%), Speed Index (10%), Largest Contentful Paint (25%), Total Blocking Time (30%) and Cumulative Layout Shift (25%).

2.5 Optimisation Techniques

Image Compression: All raster images were converted from PNG and JPEG formats to WebP using Squoosh at quality factor 80%, achieving 25–34% smaller file sizes at equivalent perceptual quality (Si and Shen, 2016; Google Developers, 2023). Images were resized to match rendered dimensions; lazy loading was applied to below-the-fold images (Netravali *et al.*, 2016), and responsive images using the srcset attribute were implemented for variable-resolution displays.

Browser Caching: Cache-Control headers were configured on the test server. Versioned static assets received Cache-Control: max-age=31536000 (one year), while dynamic HTML responses received Cache-Control: no-cache to ensure freshness. Entity tags were enabled for conditional GET validation. Returning browsers therefore serve resources from local storage rather than issuing network requests, sharply reducing repeat-visit bandwidth and load time.

Code Minification: All CSS and JS files were processed through automated minification pipelines that remove syntactically unnecessary characters without altering execution behaviour. JavaScript dead-code identification and elimination has been empirically validated (Malavolta *et al.*, 2023; Obbink *et al.*, 2018). Variable names were mangled to reduce token size, unused CSS selectors were removed using PurifyCSS, and critical above-the-fold CSS was inlined within the HTML head to eliminate render-blocking external stylesheet requests (Vogel and Springer, 2023b).

2.6 Experimental Procedure

i. Baseline measurement: each website tested ten times; median of each metric recorded. Two sites showing high variance (score ranges exceeding 12 points) were re-tested in incognito mode with extensions disabled, stabilising variance to within five points.
ii. Image compression applied; websites re-tested under identical conditions.
iii. Browser caching headers configured; repeat-visit load time recorded (cache cleared between visits).
iv. Code minification and dead-code removal applied; metrics re-recorded.
v. Combined optimisation applied; final metrics recorded under both LTE and Slow 4G conditions.
vi. Statistical analysis performed on the complete ten-site dataset.

2.7 Statistical Analysis

Percentage improvement relative to baseline was computed as [(Baseline − Post-Optimisation) ÷ Baseline] × 100. A paired-sample t-test was applied to each metric, comparing baseline and combined-optimisation measurements across all ten sites; the paired design is appropriate because the same sites were measured under both conditions (Pallant, 2020). Normality of difference scores was screened using the Shapiro-Wilk test, recommended for small samples ($n \leq 30$); all three distributions satisfied normality ($W \geq 0.91$, $p \geq 0.26$). A non-parametric Wilcoxon signed-rank test was performed as a robustness check and agreed with the t-test in all cases. Effect sizes used Cohen's d (Cohen, 1988). A significance threshold of $\alpha = 0.05$ was adopted; all p-values are two-tailed.

RESULTS AND DISCUSSION

3.1 Baseline Performance

Table 1 presents pre-optimisation metrics. Mean load time of 5.36 s far exceeds the three-second abandonment threshold. Mean page size of 6.00 MB is six times the one-megabyte mobile-optimised target recommended by the HTTP Archive (2024). Government agency sites (S4, S7) exhibited the worst performance (scores 48 and 44), while higher-education sites (S6, S9) performed relatively better, reflecting greater institutional investment in digital infrastructure. These characteristics confirm that the sampled websites present a meaningful and representative optimisation challenge.

3.2 Performance After Image Compression

Table-2 presents results following image compression. Mean page size fell from 6.00 MB to 3.72 MB, a reduction of 38.1%. Mean load time improved from 5.36 to 4.18 s (22.0% reduction), and mean Lighthouse score increased from 54.9 to 64.8. Site S7, carrying the heaviest image payload, recorded the largest absolute size saving of 3.1 MB. This gain marginally exceeds the 34% reported by Wang et al. (2022) for e-commerce platforms, attributable to the higher proportion of uncompressed PNG assets among the sampled government and news sites relative to the primarily JPEG-dominated assets in that earlier dataset. Conversion to WebP alone moved three sites (S3, S6, S9) from the “Poor” band (below 50) into the “Needs Improvement” band (50–89).

Table-2 Performance After Image Compression

Site	Load Time (s)	Page Size (MB)	Size Reduction (%)	Score (%)
S1	3.8	3.2	38.5	68
S2	4.2	3.8	37.7	66
S3	3.5	3.0	37.5	70
S4	4.9	4.5	38.4	58
S5	4.6	4.2	38.2	62
S6	3.0	2.3	37.8	76
S7	5.5	5.3	36.9	54
S8	3.9	3.5	37.5	66
S9	3.3	2.7	38.6	72
S10	5.1	4.7	39.0	56
Mean	4.18	3.72	38.1	64.8

3.3 Performance After Browser Caching

Table-3 presents repeat-visit load times following caching-header implementation. Mean repeat-visit load time fell from 5.36 to 3.04 s, a reduction of 43.3%. Sites with the largest numbers of static resources (S2, S4, S7) recorded the largest absolute gains. Unlike compression, caching eliminates network transfer entirely for locally held resources; it therefore benefits returning visitors most, whereas compression benefits all users on first load. The HTTP Archive (2024) documents that static-asset caching consistently ranks among the highest-impact single interventions for reducing repeat-visit transfer volume.

Table-3 Performance After Browser Caching (Repeat-Visit Condition)

Site	Baseline Load (s)	Cached Load (s)	Reduction (%)	Score (%)
S1	4.8	2.8	41.7	65
S2	5.4	2.9	46.3	63
S3	4.5	2.7	40.0	69
S4	6.2	3.5	43.5	57
S5	5.8	3.3	43.1	61
S6	3.9	2.1	46.2	75
S7	7.1	4.0	43.7	53
S8	5.0	2.9	42.0	65

Abubakar and Muhammad (2026). Analysis of Website Performance Optimisation Techniques for Mobile Users in Low-Bandwidth Environments. *Nigeria Journal of Engineering Science Research (NIJESR)*, 9(2), 72-84. https://doi.org/10.5281/zenodo.21917690

S9	4.3	2.4	44.2	71
S10	6.6	3.8	42.4	55
Mean	5.36	3.04	43.3	63.4

3.4 Performance After Code Minification

Table-4 shows results following CSS and JS minification and dead-code removal. Mean resource-bundle size fell by 18.3% relative to baseline. The effect on Lighthouse score exceeded what raw size reduction alone predicts, because critical-CSS inlining additionally eliminated render-blocking requests, directly reducing Total Blocking Time — the metric given the highest weight (30%) in Lighthouse scoring (Walton, 2024). JavaScript-heavy e-commerce sites (S2, S5, S8) recorded the largest Total Blocking Time reductions, aligning with Google Developers (2023) guidance that render-blocking scripts are the primary controllable contributor to score penalties on resource-heavy pages.

Table-4 Performance After Code Minification

Site	Resource Reduction (%)	Load Time (s)	Score (%)
S1	18	4.0	72
S2	16	4.5	69
S3	21	3.8	74
S4	17	5.2	59
S5	19	4.9	63
S6	20	3.2	78
S7	16	5.9	55
S8	18	4.2	67
S9	21	3.6	75
S10	17	5.5	57
Mean	18.3	4.48	66.9

3.5 Combined Optimisation Results

Table-5 presents combined-optimisation results under LTE conditions. Combined application of all three techniques achieved a mean load time of 2.67 s, a 50.5% reduction from baseline. Mean page size fell 48.2% from 6.00 to 3.11 MB, and mean HTTP request count fell by approximately 29% through script consolidation and elimination of redundant third-party calls. Mean Lighthouse score rose from 54.9 to 83.0, an improvement of 28.1 percentage points. Six of ten sites crossed the "Good" threshold (score ≥ 80), two (S6, S9) crossed the "Excellent" classification (≥ 90), and no site remained "Poor" (≤ 49).

Table-5 Performance After Combined Optimisation (LTE)

Site	Load Time (s)	Page Size (MB)	Requests	Score (%)	Load Reduction (%)
S1	2.3	2.7	64	86	52.1
S2	2.6	3.0	73	81	51.9
S3	2.1	2.4	58	87	53.3
S4	3.2	3.7	87	78	48.4
S5	2.6	3.3	79	82	55.2
S6	1.9	2.0	51	92	51.3
S7	3.8	4.5	99	75	46.5
S8	2.7	3.1	70	84	46.0
S9	2.0	2.2	56	89	53.5
S10	3.5	4.2	91	77	47.0
Mean	2.67	3.11	72.8	83.0	50.5

3.6 Network Throttling Results (Slow 4G Simulation)

Table-6 presents performance under Lighthouse Slow 4G throttling (1.6 Mbps downlink, 150 ms round-trip time), representative of rural Nigerian conditions. Baseline mean load time rose to 15.00 s, well beyond the abandonment threshold for most users. Following combined optimisation under the same throttled conditions, mean load time fell to 6.34 s — a 57.9% reduction — showing that optimisation delivers proportionally larger absolute benefits under constrained bandwidth. The compressive effect of reduced payload is amplified as throughput decreases, reinforcing optimisation as a digital-equity intervention rather than merely a performance refinement.

Table-6 Combined Optimisation Performance Under Slow 4G Throttling

Site	Baseline Load (s)	Optimised Load (s)	Reduction (%)	Optimised Score (%)
S1	13.4	5.5	58.9	79
S2	15.1	6.5	56.9	74
S3	12.6	5.3	57.9	81
S4	17.4	7.4	57.5	71
S5	16.2	6.9	57.4	75
S6	10.9	4.3	60.6	84
S7	19.9	8.6	56.8	68
S8	14.0	6.0	57.1	77
S9	12.0	4.8	60.0	83
S10	18.5	8.1	56.2	70
Mean	15.00	6.34	57.9	76.2

3.7 Statistical Analysis

Table-7 presents paired-sample t-tests comparing baseline and combined-optimisation measurements across all ten sites. Shapiro-Wilk screening confirmed normality for all three metrics (load time: $W = 0.94$, $p = 0.57$; page size: $W = 0.91$, $p = 0.26$; performance score: $W = 0.93$, $p = 0.48$). All three metrics yielded highly significant results ($p < 0.001$), confirming that the observed improvements are not attributable to random measurement variation. Effect sizes (Cohen's d) were very large — 6.00 for load time, 3.98 for page size and 12.58 for performance score. The 95% confidence interval for mean load-time reduction (2.37 to 3.01 s) excludes zero. Wilcoxon signed-rank tests produced consistent conclusions for all three metrics.

Table-7 Paired-Sample t-Test Results — Baseline vs Combined Optimisation (n = 10)

Metric	Mean Before	Mean After	Mean Diff.	t(9)	p-value	Cohen's d	95% CI
Load Time (s)	5.36	2.67	2.69	18.97	< 0.001	6.00	[2.37, 3.01]
Page Size (MB)	6.00	3.11	2.89	12.58	< 0.001	3.98	[2.37, 3.41]
Score (%)	54.9	83.0	28.1	39.78	< 0.001	12.58	[26.5, 29.7]

Figs. 1 to 7 present the study results graphically.

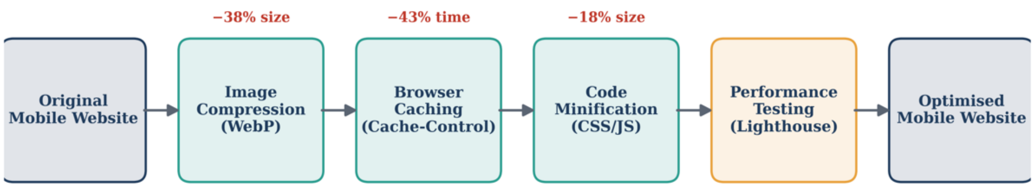

Fig. 1 Multi-technique optimization pipeline and individual performance gains per stage

Abubakar and Muhammad (2026). Analysis of Website Performance Optimisation Techniques for Mobile Users in Low-Bandwidth Environments. *Nigeria Journal of Engineering Science Research (NIJESR)*, 9(2), 72-84. https://doi.org/10.5281/zenodo.21917690

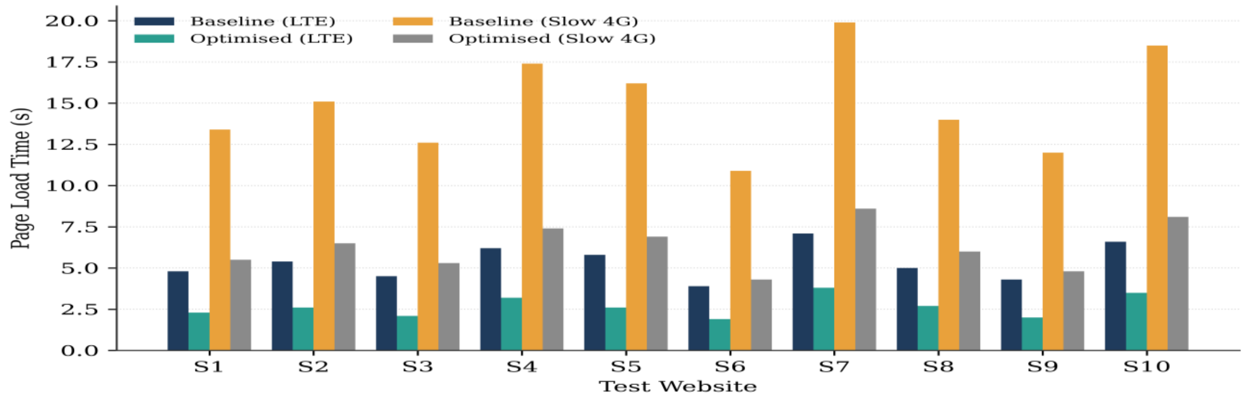

Fig. 2 Page load time across all ten sites — baseline vs combined optimisation under LTE and Slow 4G conditions.

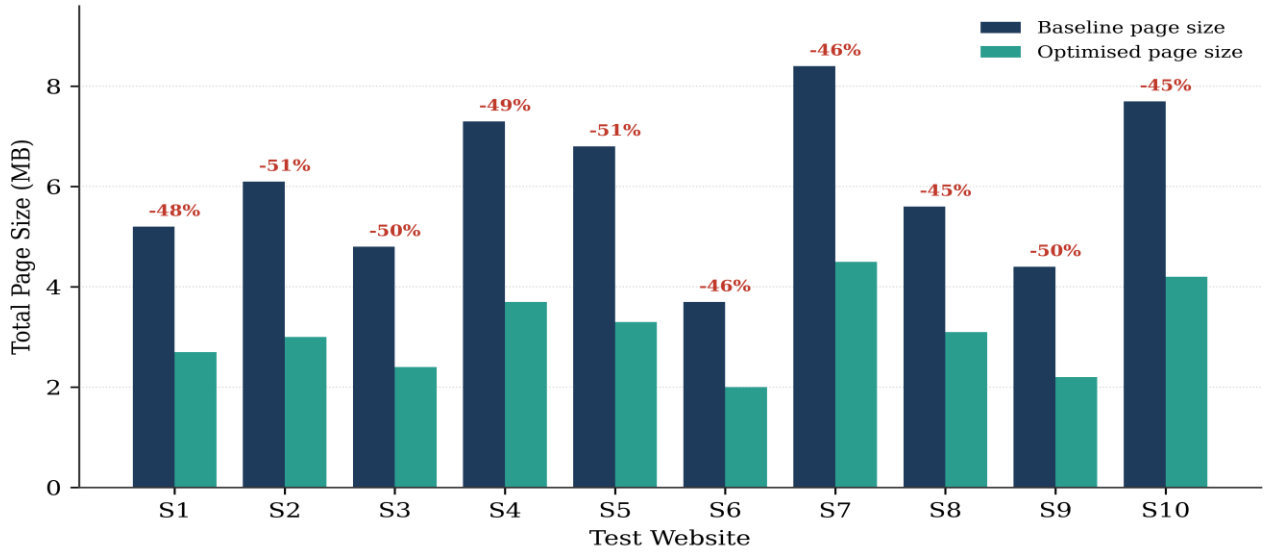

Fig. 3 Total page size before and after combined optimisation for all ten sites, with percentage reduction annotated

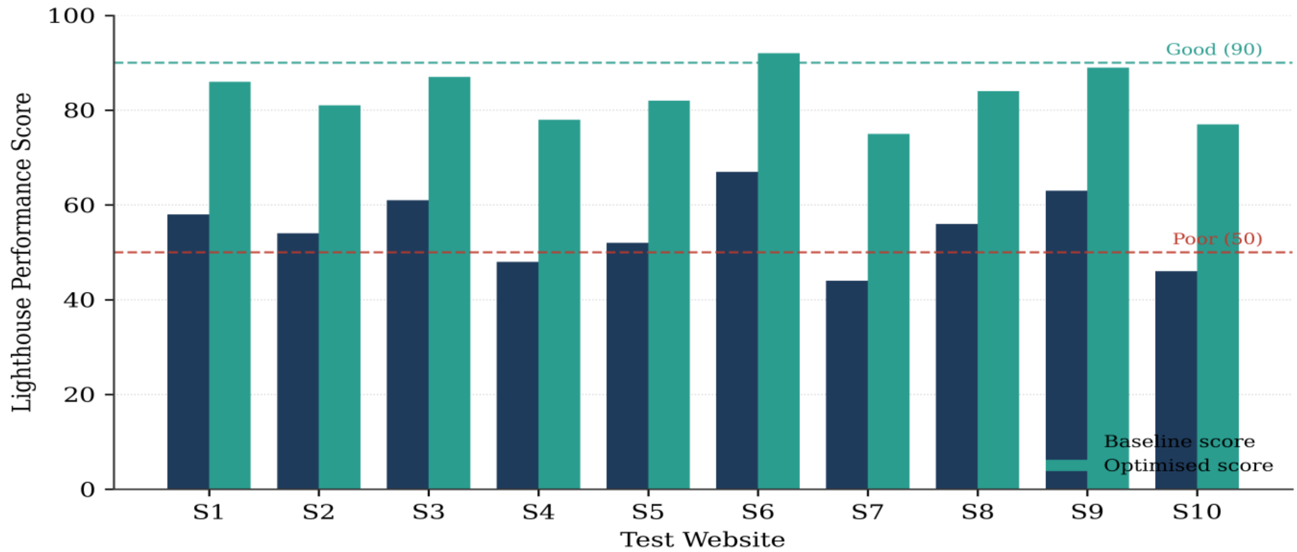

Fig. 4 Lighthouse performance scores before and after combined optimisation, with "Good" (90) and "Poor" (50) thresholds indicated.

Abubakar and Muhammad (2026). Analysis of Website Performance Optimisation Techniques for Mobile Users in Low-Bandwidth Environments. *Nigeria Journal of Engineering Science Research (NIJESR)*, 9(2), 72-84. https://doi.org/10.5281/zenodo.21917690

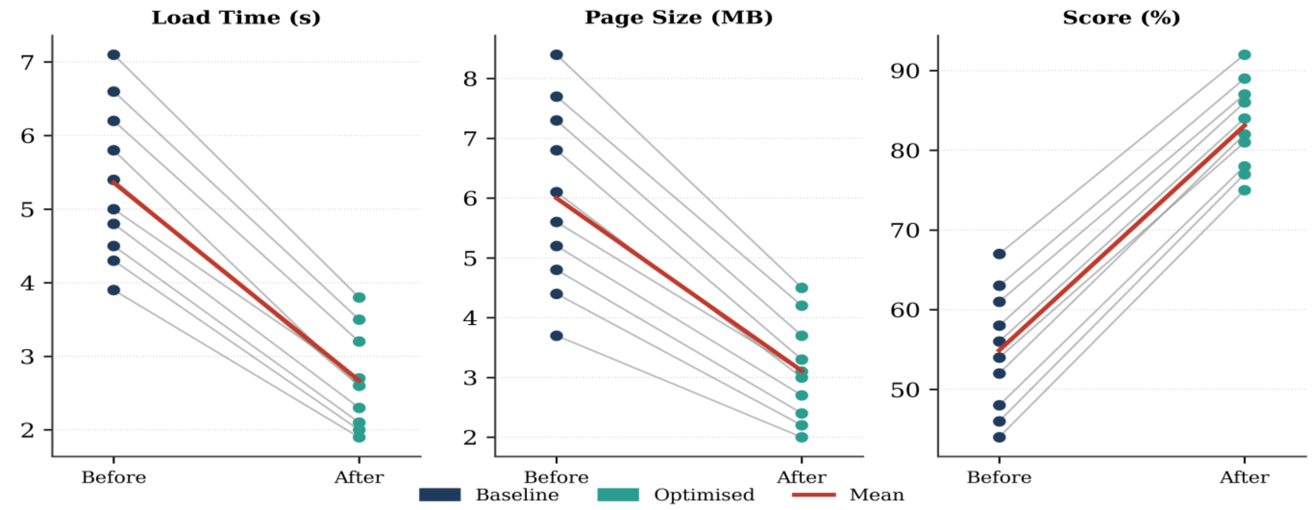

Fig. 5 Paired-sample before–after trajectories for load time, page size and performance score (n = 10).

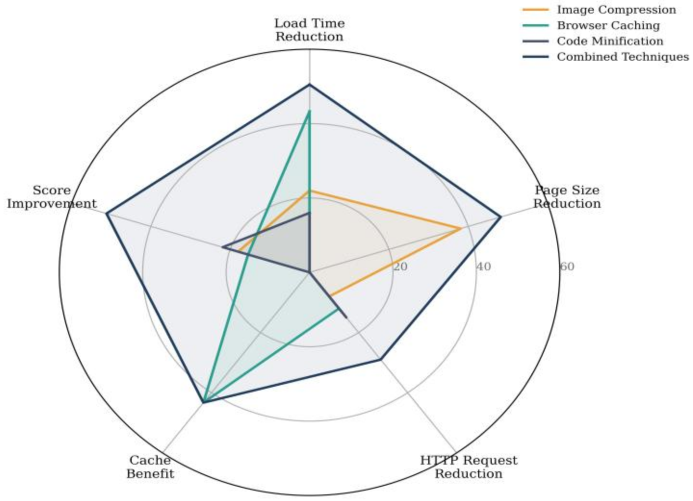

Fig. 6 Radar chart comparing the effectiveness of each individual optimisation approach across five performance dimensions

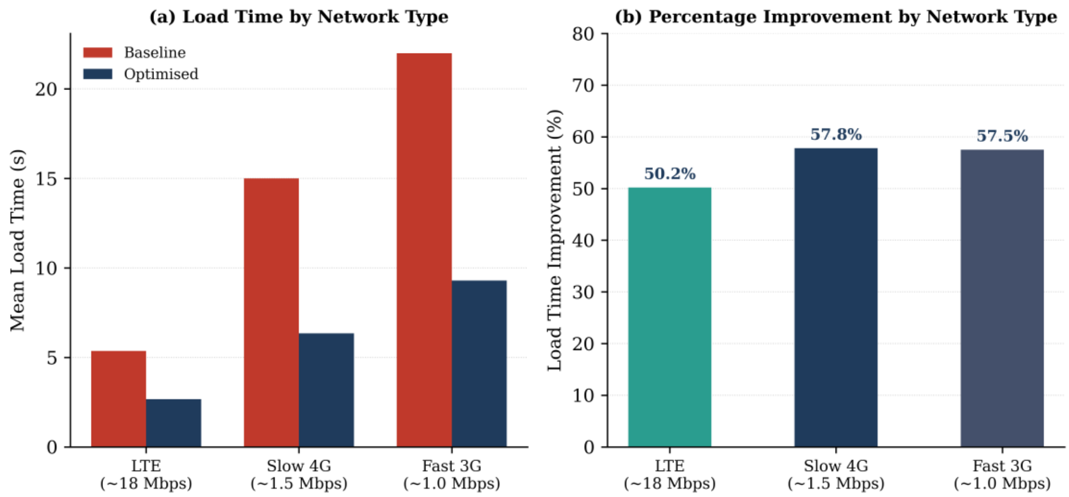

Fig. 7 Network throttling analysis — mean load times and percentage improvement under LTE, Slow 4G and Fast 3G conditions.

The results demonstrate several practically actionable findings. Image assets were the single largest contributor to page weight across all ten sites, consistent with the HTTP Archive (2024) finding that images represent roughly 68% of total mobile page payload. Image compression is therefore the highest-return, lowest-complexity intervention available to Nigerian developers seeking rapid gains with minimal infrastructure change. The throttling results carry particular significance for the Nigerian context. Under Slow 4G conditions, combined optimisation reduced mean load time from 15.00 to 6.34 s, an absolute saving of 8.66 s per page visit. At a conservative five page visits per day, an optimised website saves a rural user roughly 43 s of load time daily and meaningfully reduces data expenditure. Given that the National Digital Economy Policy and Strategy (2020) targets digital inclusion for underserved populations, optimisation emerges as a technically straightforward mechanism for advancing policy objectives. The compounding effect is practically significant. Applied independently to the first-visit LTE condition, image compression reduces load time by 22.0%, minification by approximately 16%, and isolated caching has minimal first-visit effect; applied together, all three achieve 50.5%, because compressed assets transfer and cache more efficiently while minified scripts parse faster once payload is reduced. This is consistent with the integrated framework of Nabi and Jhanjhi (2021) and Vepsalainen et al. (2024), who argue that optimisation should be treated as a holistic engineering discipline. The very large effect sizes (Cohen's d > 3.9 for all metrics) underline the depth of under-optimisation present before intervention. The ten test websites, while representative of common Nigerian service categories, cannot fully characterise the national web ecosystem. The Lighthouse synthetic environment, though standardised and reproducible, differs from real-user monitoring data that captures population-level network variation. This study evaluated three front-end methods; server-side interventions such as CDN deployment (Nygren *et al.,* 2010), Brotli compression (Alakuijala *et al.,* 2015), HTTP/3 adoption and service-worker offline caching (Tandel and Jamadar, 2018) were not evaluated and represent important directions for future research.

3.8 Threats to Validity

Internal validity concerns whether improvements are attributable to the interventions rather than extraneous factors. The primary internal threat is Lighthouse measurement variability, which can produce fluctuations of two to five points across repeated runs (Nabi and Jhanjhi, 2021); this was mitigated by ten repeated measurements per condition and median recording. A secondary threat arises from sequential application: because compression preceded caching, caching measurements reflect already-compressed assets, potentially inflating the apparent caching contribution. Future studies should use factorial designs that randomise technique order. External validity concerns generalisability beyond the ten sites. All were Nigerian properties in four categories; results may not generalise to single-page applications, media-streaming platforms or progressive web applications with service-worker caching already present. Replication across a broader sample, including mobile-first React and Vue applications, would strengthen external validity. Construct validity concerns whether the metrics measure the intended constructs. The Lighthouse score, though widely used and grounded in Core Web Vitals, is a synthetic proxy for user experience; synthetic scores and Chrome User Experience Report field data can diverge for sites with high geographic diversity (Google Developers, 2023). The Interaction to Next Paint metric, introduced as a Core Web Vitals signal in 2024 (Wagner and Viscomi, 2024), was not evaluated and represents an important construct for future measurement.

CONCLUSION

This study experimentally evaluated image compression, browser caching and code minification as website performance optimisation strategies for mobile users in low-bandwidth environments, across ten representative Nigerian websites under both live LTE and Slow 4G conditions. The following conclusions are supported by the evidence:

i. Image compression through WebP conversion is the single highest-return intervention: it targets the dominant source of page weight, requires no server infrastructure change, and delivers gains for every user on every visit.

ii. Browser caching is the most cost-efficient configuration change per unit of performance gain: a single set of server headers halves the load burden for returning visitors with no ongoing maintenance overhead.
iii. Code minification, while more modest in isolation, is indispensable for JS-heavy pages where render-blocking scripts are the primary Lighthouse bottleneck, and its benefits compound with compression and caching.
iv. Combined application produced gains greater than any individual approach: average scores rose from the "Poor" band into the "Good" band (above 80) for six of ten sites, with statistical significance at $p < 0.001$ and very large effect sizes across all metrics.
v. The performance dividend is largest where most needed: under Slow 4G conditions, combined optimisation saved nearly nine seconds per page visit, determining whether a service is practically usable in low-bandwidth areas.

The study recommends that Nigerian and African development teams integrate performance budgeting and automated optimisation pipelines — image conversion, caching configuration and minification — into standard continuous-integration workflows, and adopt performance-score thresholds as deployment criteria. Future research should extend this framework to CDN integration, HTTP/3 adoption and service-worker progressive-web-application strategies, and conduct real-user monitoring across diverse geographic and network contexts within Nigeria.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this research work.

ACKNOWLEDGEMENT

The authors acknowledge the Department of Electrical and Electronics Engineering, Abdullahi Fodio University of Science and Technology, Aliero, for institutional support, and colleagues who provided technical assistance during the study.

REFERENCES

Abara, C. N., Eze, O. S., & Nwosu, K. I. (2022). WebP adoption and mobile performance optimisation for Nigerian banking interfaces. *Nigerian Journal of Information Technology Research,* 8(2), 44–57.

Alakuijala, J., Kliuchnikov, E., Szabadka, Z., & Vandevenne, L. (2015). Comparison of Brotli, Deflate, Zopfli, LZMA, LZHAM and bzip2 compression algorithms. Google Inc.

Bai, X., Arapakis, I., Cambazoglu, B. B., & Freire, A. (2017). Understanding and leveraging the impact of response latency on user behaviour in web search. *ACM Transactions on Information Systems,* 36(2), 1–42.

Barford, P., Bestavros, A., Bradley, A., & Crovella, M. (1999). Changes in web client access patterns: Characteristics and caching implications. *World Wide Web,* 2(1), 15–28.

Basalla, M., Schneider, J., Luksik, M., Jaakonmäki, R., & vom Brocke, J. (2021). On latency of e-commerce platforms. *Journal of Organizational Computing and Electronic Commerce,* 31(1), 1–17.

Chukwu, P. C., & Okafor, E. D. (2023). Evaluating mobile web performance of Nigerian university portals and the impact of front-end optimisation. *Journal of Computer Science and Its Applications*, 30(1), 12–26.

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Lawrence Erlbaum Associates.

Eslamdoust, A. (2022). An overview of search engine optimization techniques. *Future Generation of Communication and Internet of Things*, 1(4), 52–57.

Grigorik, I. (2019). High performance browser networking. O'Reilly Media.

International Telecommunication Union. (2023). Measuring digital development: Facts and figures 2023. ITU.

Malavolta, I., Nirghin, K., Scoccia, G. L., Romano, S., Lombardi, S., Scanniello, G., & Lago, P. (2023). JavaScript dead code identification, elimination, and empirical assessment. *IEEE Transactions on Software Engineering,* 49, 1–24.

Nabi, G., & Jhanjhi, N. Z. (2021). Web application performance optimization: A systematic literature review. *International Journal of Advanced Computer Science and Applications*, 12(4), 311–323.

National Digital Economy Policy and Strategy. (2020). National digital economy policy and strategy for a digital Nigeria 2020–2030. Federal Ministry of Communications and Digital Economy.

Netravali, R., Goyal, A., Mickens, J., & Balakrishnan, H. (2016). Polaris: Faster page loads using fine-grained dependency tracking. In 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16).

Nigerian Communications Commission (2023). Subscriber data report quarter three 2023. NCC. https://ncc.gov.ng

Nygren, E., Sitaraman, R. K., & Sun, J. (2010). The Akamai network: A platform for high-performance internet applications. *ACM SIGOPS Operating Systems Review,* 44(3), 2–19.

Obbink, N. G., Malavolta, I., Scoccia, G. L., & Lago, P. (2018). An extensible approach for taming the challenges of JavaScript dead code elimination. In 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 291–401)

Ookla. (2024). Speedtest global index: Nigeria mobile network performance quarter one 2024. Ookla LLC. https://speedtest.net/global-index

Pallant, J. (2020). SPSS survival manual: A step-by-step guide to data analysis using IBM SPSS (7th ed.). Open University Press.

Portent. (2023). Site speed is (still) impacting your conversion rate. Portent Inc.
Sellamuthu, K., Ranjithkumar, S., Kavitha, K., & Gowtham, S. (2022). On-page SEO techniques for better ranking in search engines. In 2022 8th International Conference on Smart Structures and Systems (ICSSS) (pp. 1–6).

Si, Z., & Shen, K. (2016). Research on the WebP image format. In Advanced graphic communications, packaging technology and materials (pp. 271–277)

Souders, S. (2009). Even faster web sites: Performance best practices for web developers. O'Reilly Media.

Statista. (2024). Mobile internet traffic as percentage of total web traffic worldwide 2015–2024. Statista GmbH. https://statista.com

Tandel, S., & Jamadar, A. (2018). Impact of progressive web apps on web app development. *International Journal of Innovative Research in Science, Engineering and Technology,* 7(9), 9439–9444.

Vepsalainen, J., Hellas, A., & Vuorimaa, P. (2024). Overview of web application performance optimization techniques. Lecture Notes in Business Information Processing. Springer. arXiv:2412.07892.

Vogel, L., & Springer, T. (2023a). How streaming can improve the world (wide web). In Companion Proceedings of the ACM Web Conference 2023 (pp. 140–143).

Vogel, L., & Springer, T. (2023b). Waiter and AUTRATAC: Don't throw it away, just delay! In Web engineering, Lecture Notes in Computer Science (pp. 278–292). Springer Nature Switzerland.

Wagner, J., & Viscomi, R. (2024). Interaction to next paint becomes a Core Web Vital on March 12. Web.dev, Google LLC. https://web.dev/blog/inp-cwv-march-12

Walton, P. (2024). Web vitals. Web.dev, Google LLC. https://web.dev/articles/vitals

Wang, Y., Liu, H., & Zhang, J. (2022). Impact of image compression on mobile web performance: A comparative analysis. *International Journal of Computer Applications,* 184(12), 45–53.

World Wide Web Consortium. (2023). Web performance working group: Performance timeline specification level 3. W3C. https://www.w3.org/TR/performance-timeline