

Website Audit Report

https://example.com

Prepared by	Kestrel Web Studio
Report date	2026-10-26 11:53
Overall score	51/100 (grade E)
Issues found	9
Scope	16 checks — availability, TLS, performance, security headers, SEO, accessibility, crawlability

1. Scorecard

#	Check	Weight	Result	Evidence
1	Site reachable and serving 2xx/3xx	15	PASS	HTTP 200 in 452 ms, 400 bytes transferred
2	HTTPS enabled	10	PASS	Served over HTTPS
3	TLS certificate valid > 14 days	8	PASS	expires 2026-12-25 (60 days left)
4	Mobile viewport declared	10	FAIL	no <meta name="viewport"> — mobile renders at 980px then shrinks
5	Page title length 8-60 chars	8	PASS	"Example Domain" — 14 chars (target 15-35)
6	Meta description 40-160 chars	7	FAIL	missing — search results show no snippet
7	Exactly one H1 heading	5	FAIL	0 H1 element(s) found
8	Every image has alt text	5	PASS	all images have alt
9	Compression enabled (gzip/br)	6	PASS	HTML Content-Encoding: gzip
10	Cache-Control	5	FAIL	no Cache-

	policy set			Control header
11	Core security headers (4)	5	FAIL	present: none; missing: x-content-type-options, x-frame-options, referrer-policy, content-security-policy
12	HSTS header present	4	FAIL	no Strict-Transport-Security — first hop can be downgraded
13	JSON-LD structured data	4	FAIL	no application/ld+json — no rich results in Google
14	robots.txt served (200)	5	FAIL	unreachable — HTTPError: HTTP Error 404: Not Found
15	sitemap.xml served (200)	5	FAIL	unreachable — HTTPError: HTTP Error 404: Not Found
16	Time to first byte (informational, not scored)	0	INFO	452 ms (target < 1500 ms)

2. Executive summary

9 of 15 scored checks failed. Fixing the items below recovers the lost score, highest weight first.

1. Mobile viewport declared (10 pts, CRITICAL)
2. Meta description 40-160 chars (7 pts, HIGH)
3. Cache-Control policy set (5 pts, MEDIUM)

3. Priority fixes

3.1 Mobile viewport declared — CRITICAL

What we found: no <meta name="viewport"> — mobile renders at 980px then shrinks

Why it matters: Without a viewport declaration mobile browsers render the desktop layout at 980px and shrink it. Mobile traffic is typically 60-70% of visits for local businesses.

How to fix

4. 1. Add inside `<head>`: `<meta name="viewport" content="width=device-width, initial-scale=1">`
5. 2. Never add `user-scalable=no` or `maximum-scale=1` — it blocks pinch-zoom and fails WCAG 1.4.4.
6. 3. After adding it, check real layout at 375px width (iPhone SE) — tables and wide images usually overflow and need `max-width:100%`.

Verify the fix: `curl -sS https://HOST/ | grep -o '<meta name="viewport"[^>]*>'`

3.2 Meta description 40-160 chars — HIGH

What we found: missing — search results show no snippet

Why it matters: The meta description is the grey snippet under the title. Google uses it about 70% of the time and it drives click-through, not ranking directly.

How to fix

7. 1. Add `<meta name="description" content="...">` with 120-155 characters that state the offer, the service area and one differentiator.
8. 2. Example: `<meta name="description" content="Same-day emergency plumbing in Austin, TX. Licensed techs, flat-rate pricing, 90-minute arrival window. Call 24/7.">`
9. 3. Write a unique description per page; a site-wide duplicate is treated as thin content and usually replaced by Google's own snippet.

Verify the fix: `curl -sS https://HOST/ | grep -o '<meta name="description"[^>]*>'`

3.3 Cache-Control policy set — MEDIUM

What we found: no Cache-Control header

Why it matters: Without Cache-Control the browser re-downloads assets on every visit and CDNs cannot serve repeat visitors from the edge. This is the cheapest speed win available.

How to fix

10. 1. nginx — long-lived assets: `location ~* \.(css|js|jpg|jpeg|png|webp|svg|woff2)$ { expires 1y; add_header Cache-Control "public, max-age=31536000, immutable"; }`
11. 2. HTML must stay revalidating: `add_header Cache-Control "public, max-age=0, must-revalidate";` so content updates are not stuck in caches.
12. 3. Cloudflare — Caching → Configuration → Browser Cache TTL: 4 hours to 1 year for static, and enable **Tiered Cache** for edge hit ratio.

13. 4. Add a content hash to asset filenames (`app.a1b2c3.js`) so `immutable` caching is safe and deploys still propagate.

Verify the fix: `curl -sSI https://HOST/ | grep -i cache-control`

3.4 Exactly one H1 heading — MEDIUM

What we found: 0 H1 element(s) found

Why it matters: The H1 tells both users and crawlers what the page is about. Zero H1 leaves the topic ambiguous; several H1s split the signal and usually signal sloppy templating.

How to fix

14. 1. Keep exactly one `

` per page, matching the page's search intent, and make it visually distinct from section headings.
15. 2. Convert logo text, sidebar titles and widget headers from `

` to ` `; a logo is never the page topic.
16. 3. Do not put the whole page in the H1 — 3-10 words is the working range.

Verify the fix: `curl -sS https://HOST/ | grep -c '<h1'`

3.5 Core security headers (4) — MEDIUM

What we found: present: none; missing: x-content-type-options, x-frame-options, referrer-policy, content-security-policy

Why it matters: Missing security headers leave clickjacking, MIME-sniffing and cross-site scripting attacks open. They are also checked by every buyer's security questionnaire.

How to fix

17. 1. Add at the nginx `server {}` level (one block, all four):

```
...
add_header X-Content-Type-Options "nosniff" always;
add_header X-Frame-Options "SAMEORIGIN" always;
add_header Referrer-Policy "strict-origin-when-cross-origin" always;
add_header Permissions-Policy "geolocation=(), microphone=(), camera=()" always;
...
```

18. 2. Add a Content-Security-Policy in report-only first so you can see what breaks:


```
`add_header Content-Security-Policy-Report-Only "default-src 'self'; img-src 'self' data;; script-src 'self'" always;`
```

 — then promote to `Content-Security-Policy` after cleaning the report.
19. 3. On Cloudflare without origin access: Rules → Transform Rules → Modify Response Header, add each header as a `Set static` rule.
20. 4. Re-check with ``curl -sSI https://HOST/ | grep -iE 'x-content-type|x-frame|referrer-policy|content-security'``.

Verify the fix: `curl -sSI https://HOST/ | grep -iE 'x-content-type-options|x-frame-options|referrer-policy|content-security-policy'`

3.6 robots.txt served (200) — MEDIUM

What we found: unreachable — HTTPError: HTTP Error 404: Not Found

Why it matters: robots.txt is the first file every crawler fetches. A missing or blocking file can silently deindex the whole site; a 404 also means no sitemap discovery hint.

How to fix

21. 1. Publish `https://HOST/robots.txt` with at minimum:

```
...
```

```
User-agent: *
```

```
Allow: /
```

```
Sitemap: https://HOST/sitemap.xml
```

```
...
```

22. 2. Never ship `Disallow: /` from a staging copy — it is the most common cause of a site vanishing from Google overnight.

23. 3. Block only what must not be crawled (`/admin/`, `/cart/`, search result pages) and keep CSS/JS crawlable: Google needs them to render the page.

24. 4. After publishing, submit it in Google Search Console → Sitemaps and confirm robots.txt returns 200, not 403/404.

Verify the fix: `curl -sS -o /dev/null -w '%{http_code}\n' https://HOST/robots.txt`

3.7 sitemap.xml served (200) — MEDIUM

What we found: unreachable — HTTPError: HTTP Error 404: Not Found

Why it matters: A sitemap is how you hand Google the full URL list at once instead of waiting for crawls to discover pages. New and orphaned pages depend on it.

How to fix

25. 1. Generate `sitemap.xml` listing every indexable URL with `<lastmod>`, capped at 50,000 URLs / 50 MB per file: `<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"><url><loc>https://HOST/page</loc><lastmod>2026-01-01</lastmod></url></urlset>`

26. 2. WordPress: Yoast or Rank Math generates it at `/sitemap_index.xml` — add the rewrite so `/sitemap.xml` serves it. Static site: `sitemap` npm plugin, Hugo built-in, or a 30-line script in the build step.

27. 3. Reference it from robots.txt (`Sitemap: https://HOST/sitemap.xml`) and submit in Search Console → Sitemaps.

28. 4. Regenerate on every deploy, not by hand — a stale sitemap teaches Google to ignore it.

Verify the fix: `curl -sS -o /dev/null -w '%{http_code}\n' https://HOST/sitemap.xml`

3.8 HSTS header present — MEDIUM

What we found: no Strict-Transport-Security — first hop can be downgraded

Why it matters: Without HSTS the first request of every session can still be downgraded to plain HTTP by an on-path attacker, even though HTTPS is enabled.

How to fix

29. 1. Only after HTTPS and the HTTP→HTTPS redirect both work, add: ``add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;``
30. 2. Start short so a mistake is recoverable: ``max-age=300``, verify the site over HTTPS for a day, then raise to 31536000 (1 year).
31. 3. Add ``preload`` and submit to <https://hstspreload.org> only when every subdomain — including old staging hosts — serves HTTPS. Preload is hard to undo.
32. 4. Cloudflare: SSL/TLS → Edge Certificates → Enable HSTS (max-age 12 months, no preload until you are certain).

Verify the fix: `curl -sSI https://HOST/ | grep -i strict-transport-security`

3.9 JSON-LD structured data — MEDIUM

What we found: no application/ld+json — no rich results in Google

Why it matters: JSON-LD structured data makes Google show rich results — star ratings, opening hours, FAQ dropdowns, breadcrumbs. Pages with it get measurably higher CTR at the same rank.

How to fix

33. 1. Add one `<script type="application/ld+json">` block in `<head>` describing the business:

...

```
{ "@context": "https://schema.org", "@type": "LocalBusiness", "name": "BUSINESS_NAME", "url": "https://HOST/", "telephone": "+1-555-0100", "address": { "@type": "PostalAddress", "streetAddress": "STREET", "addressLocality": "CITY", "addressRegion": "ST", "postalCode": "ZIP", "addressCountry": "US" }, "openingHours": "Mo-Fr 08:00-18:00" }
```

...

34. 2. Add ``FAQPage`` markup to any page with Q&A blocks and ``BreadcrumbList`` to inner pages — those are the two easiest rich results to earn.
35. 3. Validate at <https://search.google.com/test/rich-results> before shipping; a malformed block is silently ignored.
36. 4. Keep the markup in sync with visible page content — mismatch is a manual-action risk, not just a lost rich result.

Verify the fix: `curl -sS https://HOST/ | grep -o 'application/ld+json' | head -1`

4. Method

Each site is fetched with a Chrome user agent and Accept-Encoding: gzip, deflate, following redirects, with a 2 MB body cap. Headers come from the live response; the certificate is read from a real TLS handshake; robots.txt and sitemap.xml are requested at the site root. Each check carries a weight and the score is the weighted pass ratio.

5. Next steps

37. Work top-down through section 3 — highest weight first is the cheapest score to recover.
38. Re-run the audit after each batch of fixes.
39. Anything marked CRITICAL should ship this week.