Audit Report: Full-stack digital solutions – premier custom software & design

Website: https://playandnope.com/
Date: 12.10.2026
Audit Coverage: 89% — PageSpeed Insights (desktop): PSI HTTP 502; PageSpeed Insights (mobile): PSI HTTP 500; PageSpeed Insights: mobile: PageSpeed fetched an error page — The target returned HTTP 500, so the fetched response is an error page rather than the real content. The audit cannot reach the page.; desktop: PageSpeed fetched an error page — The target returned HTTP 500, so the fetched response is an error page rather than the real content. The audit cannot reach the page.; W3C Validator: W3C validator could not fetch the page — The target returned HTTP 500, so the fetched response is an error page rather than the real content. The audit cannot reach the page.; PageSpeed Insights (desktop): PSI HTTP 500
Confidence: low

Pages Audited (5 of 5):

- https://playandnope.com/
- https://playandnope.com/privacy-policy
- https://playandnope.com/contact
- https://playandnope.com/full-stack-digital-agency
- https://playandnope.com/digital-solutions-contact

Summary of results

Overall Score: 58 / 100
Status: □ □ **Poor**

Site overall 58 is the mean of 4 pages. Scores range 18 (https://playandnope.com/digital-solutions-contact) → 88 (https://playandnope.com/privacy-policy). Weakest page: The audited URL returns HTTP 500 with the WordPress 'Error establishing a database connection' page — confirmed independently by the HTTP response (status 500), the W3C validator fetch failure, the browser runtime console error, and the HTML inventory (title 'Database Error', 2.4 KB body, zero scripts, images, links or landmarks). That single fact dominates the score: the page is unusable and unindexable, so no amount of header or asset polish matters until the database is restored. Even setting the outage aside, PSI

mobile performance is 35 with LCP 14.2 s and TBT 25.66 s — both far past the heavy-penalty thresholds — driven by ~600 KB of unused reCAPTCHA/theme JavaScript and long tasks from jQuery (898 ms) and core.js (626 ms). Accessibility scores 99 in Lighthouse but axe still flags a serious html-has-lang violation plus missing main landmark and region containment. Security basics PASSes, but the CSP is only `frame-ancestors 'self'` with no default-src or object-src 'none', and the document is served `no-store` with no cache plugin or CDN detected. Desktop PSI errored (HTTP 500) and W3C could not fetch the page, so desktop parity is unverified and confidence is medium.

Per-page scores

 Poor · https://playandnope.com/

Score	Performance	Accessibility	Best Practices	SEO	Security
58	41	100	100	92	PASS

 Needs Improvement · https://playandnope.com/privacy-policy

Score	Performance	Accessibility	Best Practices	SEO	Security
88	96	100	100	100	PASS

 Needs Improvement · https://playandnope.com/contact

Score	Performance	Accessibility	Best Practices	SEO	Security
68	45	99	96	100	PASS

— · https://playandnope.com/full-stack-digital-agency

Score	Performance	Accessibility	Best Practices	SEO	Security
—	—	—	—	—	PASS

 Critical · https://playandnope.com/digital-solutions-contact

Score	Performance	Accessibility	Best Practices	SEO	Security
18	35	99	96	100	PASS

PageSpeed Insights — Mobile vs Desktop

*Lower is worse for Performance; higher is worse for LCP and CLS. Worse value is **bolded**.*

URL	Performance (M / D)	LCP (M / D)	CLS (M / D)
https://playandnope.com/	41 / —	6.60 s / —	0.000 / —
https://playandnope.com/privacy-policy	**96** / 97	**2.34 s** / 581 ms	0.000 / **0.098**
https://playandnope.com/contact	— / 45	— / 2.59 s	— / 0.083
https://playandnope.com/digital-solutions-contact	35 / —	14.16 s / —	0.056 / —

Optimization Checklist

1 of 2 passing — 1 pass · 0 warn · 1 fail · 6 n/a

Item	Status	Detail
Page caching plugin / CDN active	**Fail**	No WordPress cache plugin marker or CDN edge cache detected on the document response.
Response compressed (gzip / brotli)	**Pass**	Document response is compressed with gzip.
Images lazy-loaded	N/A	No raster elements found (25 SVGs, 5 placeholders excluded).
Hero image eagerly loaded	N/A	No raster elements found (25 SVGs, 5 placeholders excluded).
Hero is a real (not a CSS background-image)	N/A	No CSS background-images detected on raster-image-eligible elements.
Responsive images (srcset /)	N/A	Only 0 raster images on the page (25 SVGs, 5 placeholders excluded) — responsive-image rule does not apply.

Item	Status	Detail
Reasonable number of image sizes	N/A	Too few raster images to evaluate srcset width variety.
JS scripts not blocking in	N/A	No external scripts on the page.

Fixes

Priority 1: Critical

Immediate action — impacts user experience, search rankings, or site safety.

1A. Eliminate long tasks driving 21.14 s TBT `Performance`

- **Impact:** TBT, TTI, INP, Speed Index
- **Problem:** TBT is 21,140 ms (threshold is 600 ms) with long tasks of 444 ms and 428 ms from global.min.js and core.min.js, plus 92 KB of unused JS in global.min.js.
- **Solution:**
  - Code-split `global.min.js` and `core.min.js`; load only what the current route needs.
  - Remove the 92 KB of unused JS reported by Lighthouse.
  - Defer or lazy-init non-critical widgets (carousels, modals, analytics) until interaction.
  - Audit the WebGL/canvas work — the 4 console warnings ('GPU stall due to ReadPixels') suggest a render loop blocking the main thread.
  - Re-measure TBT after each change; target <200 ms.

1B. Fix 6.6 s LCP `Performance`

- **Impact:** LCP, Speed Index
- **Problem:** LCP is 6.6 s while FCP is only 1.81 s — the LCP element is being delayed by JS execution, not by network.
- **Solution:**
  - Identify the LCP element in DevTools Performance panel.
  - If it's an image, add `<link rel="preload" as="image">` and ensure it's not lazy-loaded.
  - If it's text, ensure the font is preloaded and `font-display: swap` is set.

- Avoid rendering the hero via JS after hydration; server-render it.

1C. Fix color-contrast failure on the h1 `Accessibility`

- **Impact:** WCAG 1.4.3 (AA), axe serious violation
- **Problem:** axe reports 1 serious `color-contrast` violation on the `h1` ("Privacy Policy"), and could not determine contrast for 7 additional nodes.
- **Solution:**
  - Raise the h1 foreground/background contrast to at least **4.5:1** for normal text, or **3:1** if the h1 renders at ≥24 px (or ≥18.66 px bold).
  - Re-run axe after the change and manually verify the 7 undetermined nodes (alpha-transparency / gradient backgrounds are skipped by axe).
  - Also check the 1 `link-in-text-block` node — links inside body copy must be distinguishable by more than color alone (e.g. add `text-decoration: underline`).

1D. Cut main-thread blocking time (TBT 23.84 s) `Performance`

- **Impact:** TBT, TTI, Max Potential FID, Speed Index
- **Problem:** TBT is 23,840 ms and TTI 31.4 s; long tasks come from core.js (932 ms), jquery.min.js (729 ms) and global.min.js (300–366 ms ×7), while reCAPTCHA ships 1.05 MB from gstatic with three ~170 KB unused chunks.
- **Solution:**
  - Load reCAPTCHA only when the form is actually submitted (lazy-init the widget on first interaction) instead of on page load.
  - Drop jQuery if the theme allows; the 729 ms long task is pure overhead on a contact page.
  - Split `global.min.js` (92 KB wasted) and `core.js` so only the code needed for this route is parsed.
  - Audit the 22 script requests / 842 KB of script bytes — most are not needed for a contact form.

1E. Harden the Content-Security-Policy `Security`

- **Impact:** XSS mitigation, script injection
- **Problem:** CSP is `frame-ancestors 'self';` only — no `default-src`, no `object-src 'none'`, no `script-src`. The page contains a user-submitted `<textarea>` (hasUserContent: yes), so an XSS would persist into stored content.
- **Solution:** Deploy a nonce-based CSP with `'strict-dynamic'` rather than a flat allowlist:

```
Content-Security-Policy:
  default-src 'self';
  script-src 'nonce-{random}' 'strict-dynamic' https:;
  object-src 'none';
  base-uri 'none';
  frame-ancestors 'self';
  require-trusted-types-for 'script';
```

Emit the nonce server-side per request and propagate it to every `<script>` tag.

1F. Restore the database connection — the page is returning HTTP 500 `Best Practices`

- **Impact:** Availability, SEO, all metrics
- **Problem:** The audited URL returns HTTP 500 with title 'Database Error' and h1 'Error establishing a database connection'; W3C and the browser runtime both confirm the failure.
- **Solution:**
  - Check MySQL/MariaDB is running and reachable from the web host.
  - Verify `DB_HOST`, `DB_USER`, `DB_PASSWORD`, `DB_NAME` in `wp-config.php`.
  - Inspect `wp-content/debug.log` and the PHP/MySQL error logs for the root cause (corrupt table, disk full, max connections).
  - Serve a proper 503 with `Retry-After` during maintenance instead of a bare 500 so crawlers back off gracefully.

1G. Eliminate 25.7 s of blocking JavaScript and the 14.2 s LCP `Performance`

- **Impact:** TBT, LCP, Speed Index, TTI
- **Problem:** TBT is 25,660 ms and LCP 14.2 s on mobile; long tasks come from jquery.min.js (898 ms), core.min.js (626 ms) and global.min.js (331/312/297/293/284/273/268 ms), with ~520 KB of unused reCAPTCHA JS.
- **Solution:**
  - Defer or async every non-critical script:

```
<script src="/wp-content/themes/.../global.min.js" defer></script>
```

  - Load reCAPTCHA only when the contact form is focused/visible (IntersectionObserver + dynamic `<script>` injection) instead of on page load.
  - Audit jQuery usage — if the theme no longer needs it, drop it entirely.
  - Split the long tasks in `global.min.js` and `core.min.js` with `requestIdleCallback` / `scheduler.yield`.

- Ship source maps so the bundles can be profiled.

1H. Stop serving the document with no-store and add page caching / CDN `Performance`

- **Impact:** TTFB, repeat-visit LCP, bandwidth
- **Problem:** The document response carries `cache-control: no-cache, must-revalidate, max-age=0, no-store, private`, and the optimized-web checklist reports no WordPress cache plugin or CDN edge cache detected.
- **Solution:**
  - Remove `no-store` from cacheable HTML responses; use `Cache-Control: public, max-age=0, s-maxage=600, stale-while-revalidate=60` behind a CDN.
  - Install a page-cache plugin (e.g. WP Rocket, LiteSpeed Cache) or front the site with a CDN edge cache.
  - Keep `no-store` only on authenticated or cart/checkout responses.
  - Add long-lived immutable caching for hashed static assets:

```
Cache-Control: public, max-age=31536000, immutable
```

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Set a real cache policy on the document `Performance`

- **Impact:** Repeat-visit TTFB, LCP
- **Problem:** The document response sends `cache-control: max-age=0` with an `expires` in the past, forcing revalidation on every navigation despite WP Rocket being active.
- **Solution:**
  - Configure WP Rocket / server to send a positive `max-age` with `stale-while-revalidate`, e.g.:

```
Cache-Control: public, max-age=600, stale-while-revalidate=86400
```

  - Verify the cache is warmed for the canonical URL after deploy.

2B. Drop legacy JavaScript polyfills `Performance`

- **Impact:** Bundle size, TBT

- **Problem:** Lighthouse reports ~10 KiB of legacy JavaScript savings and 92 KB unused JS in global.min.js.
- **Solution:**
  - Update `browserslist` to drop IE11 / legacy targets.
  - Remove `jquery-migrate` if no legacy plugin depends on it.
  - Rebuild and re-measure.

2C. Replace vague repeated link text `SEO`

- **Impact:** SEO link-text audit, screen-reader navigation
- **Problem:** The `link-text` SEO audit fails with 4 links lacking descriptive text, and the inventory shows 'read more' repeated ×4 plus 'development', 'design & innovation', etc. repeated 3× each.
- **Solution:**
  - Replace 'read more' with the destination, e.g. `<a href="/services/design">Design & Innovation services</a>`.
  - If visual design requires short text, add `aria-label` with the full context.

2D. Add valid structured data `SEO`

- **Impact:** Rich results eligibility
- **Problem:** The `structuredData` SEO audit fails and the single JSON-LD block has type 'Unknown'.
- **Solution:**
  - Add a valid `Organization` and `WebSite` JSON-LD block:

```
<script type="application/ld+json">
{"@context":"https://schema.org","@type":"Organization","name":"Pla
y and Nope","url":"https://playandnope.com/"}
</script>
```

  - Validate with Google's Rich Results Test.

2E. Enlarge tap targets `SEO`

- **Impact:** Mobile usability
- **Problem:** The `tapTargets` SEO audit fails on mobile.
- **Solution:**
  - Ensure every interactive element is at least 44×44 px with adequate spacing.
  - Increase padding on nav links and icon buttons rather than shrinking them.

2F. Add Subresource Integrity to cross-origin scripts `Security`

- **Impact:** Supply-chain integrity
- **Problem:** 2 of 2 cross-origin scripts (from cdn.jsdelivr.net) lack an `integrity` attribute.
- **Solution:**

```
<script
src="https://cdn.jsdelivr.net/gh/orestbida/cookieconsent@3.0.1/dist
/cookieconsent.umd.js"
        integrity="sha384-…" crossorigin="anonymous" defer>
</script>
```

  Generate the hash with `openssl dgst -sha384 -binary file.js | openssl base64 -A`.

2G. Harden the Content-Security-Policy `Security`

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is present but only sets `frame-ancestors 'self';` — no `default-src` and no `object-src 'none'`, so it provides no script or resource restrictions. The site also serves a WordPress login at the default `/wp-login.php` path, which is an auth surface even though the heuristic signals say otherwise.
- **Solution:** Add a nonce-based policy rather than a flat allowlist:

```
Content-Security-Policy:
  default-src 'self';
  object-src 'none';
  base-uri 'none';
  script-src 'nonce-{random}' 'strict-dynamic' https:;
  frame-ancestors 'self';
```

  - Roll out in `Content-Security-Policy-Report-Only` first, collect violations, then enforce.
  - Keep `frame-ancestors 'self'` (it already covers clickjacking alongside `X-Frame-Options: SAMEORIGIN`).

2H. Eliminate the /privacy-policy → /privacy-policy/ redirect `Performance`

- **Impact:** Document latency, FCP, Speed Index
- **Problem:** Browser runtime shows the document request to `/privacy-policy` took 638 ms and returned 0 B before redirecting to `/privacy-policy/` (287 ms, 21.1 KB); PSI flags `document-latency-insight` with an estimated 640 ms saving.
- **Solution:**

- Link to the canonical trailing-slash URL everywhere: `https://playandnope.com/privacy-policy/`.
- Ensure the redirect is a server-side 301 (not a JS/meta refresh) so it is cacheable.
- Audit internal navigation and sitemap entries for the non-slash variant.

2I. Fix document caching (cache-control: max-age=0) `Performance`

- **Impact:** repeat-visit TTFB, LCP
- **Problem:** The HTML document is served with `cache-control: max-age=0` and an `expires` equal to the response date, so nothing is cached despite WP Rocket being detected.
- **Solution:**
  - Let WP Rocket serve the cached HTML with a short `max-age` + `stale-while-revalidate` for anonymous visitors.
  - Set long-lived `Cache-Control: public, max-age=31536000, immutable` on hashed theme assets (`global.62444fa2…min.js`, `core.c872dceb…min.js`).
  - Bypass the cache for logged-in users and for POST responses.

2J. Add HSTS preload, COOP and CORP `Security`

- **Impact:** transport hardening, origin isolation
- **Problem:** HSTS is `max-age=31536000; includeSubDomains` without `preload`; Cross-Origin-Opener-Policy and Cross-Origin-Resource-Policy are both missing.
- **Solution:**

```
Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains; preload"
Header always set Cross-Origin-Opener-Policy "same-origin"
Header always set Cross-Origin-Resource-Policy "same-origin"
```

  Submit the domain to hstspreload.org once the preload directive is live.

2K. Resolve the reCAPTCHA console error `Best Practices`

- **Impact:** console errors, Best Practices score
- **Problem:** `requestStorageAccess: Permission denied` is logged from the reCAPTCHA anchor iframe, and `errors-in-console` is a failing Best Practices audit.
- **Solution:**
  - Update the reCAPTCHA integration to the current API and load it only on form submit.

    - If the error persists, gate the storage-access call behind a feature check and catch the rejection so it does not surface as an unhandled console error.

2L. Label the unlabeled textarea and fix the presentational-role conflict `Accessibility`

- **Impact:** WCAG 1.3.1, 4.1.2
- **Problem:** Form 1 contains a `textarea` with no label, and axe reports `presentation-role-conflict` on `iframe[title="reCAPTCHA"]`.
- **Solution:**
    - Add a visible `<label for="…">` (or `aria-label`) to the textarea.
    - Remove `role="presentation"` / `role="none"` from the reCAPTCHA iframe, or stop applying it to an element that also carries a `title`.

2M. Fix the two W3C errors and add real structured data `SEO`

- **Impact:** HTML validity, rich results eligibility
- **Problem:** W3C reports 2 errors — hidden inputs carrying `autocomplete='off'` — and PSI's `structuredData` modeled audit fails with the single JSON-LD block typed as `Unknown`.
- **Solution:**
    - Remove `autocomplete` from `<input type="hidden">` elements (Gravity Forms `gform_hidden` fields).
    - Replace the untyped JSON-LD block with a valid `Organization` / `LocalBusiness` schema including `name`, `url`, `logo` and `contactPoint`.

2N. Move login off /wp-login.php and suppress the Server header `Security`

- **Impact:** attack surface, information disclosure
- **Problem:** Security basics flags `Login not on default path` as warn (form served on `/wp-login.php`) and `Technology disclosure` as warn (`Server: Apache`).
- **Solution:**
    - Rename the login path (e.g. via a security plugin) and block direct access to `/wp-login.php`.
    - Strip the version banner: `ServerTokens Prod` and `ServerSignature Off` in Apache, or remove the `Server` header at the edge.

2O. Add a lang attribute to `Accessibility`

- **Impact:** WCAG 3.1.1, screen reader pronunciation

- **Problem:** axe-core reports a serious `html-has-lang` violation — the `<html>` element has no lang attribute, and the HTML inventory confirms 'Lang: not set'.
- **Solution:**

  ```
  <html lang="en">
  ```

  Use the correct BCP-47 tag for the page's primary language.

2P. Add main/nav landmarks, region containment and a skip link `Accessibility`

- **Impact:** WCAG 1.3.1, 2.4.1, keyboard navigation
- **Problem:** axe flags `landmark-one-main` and `region` (moderate), and the HTML inventory finds no `<nav>`, `<main>`, `<header>` or `<footer>` landmarks and no skip-to-content link.
- **Solution:**
  - Wrap primary content in `<main id="main-content">` and navigation in `<nav>`.
  - Ensure all content sits inside a landmark region.
  - Add a skip link as the first focusable element:

```
<a class="skip-link" href="#main-content">Skip to content</a>
```

2Q. Tighten the Content-Security-Policy `Security`

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is only `frame-ancestors 'self';` — it has no `default-src` and no `object-src 'none'`, so it restricts nothing about scripts or resources.
- **Solution:** Add a nonce-based policy rather than a flat allowlist:

```
Content-Security-Policy: default-src 'self'; object-src 'none';
base-uri 'none';
  frame-ancestors 'self'; script-src 'nonce-{random}' 'strict-
dynamic';
```

Note: site signals show no auth, payments or user content, so this is defense-in-depth rather than an active exposure.

Priority 3: Best Practice

Recommended for long-term maintainability.

3A. Strengthen the Content-Security-Policy `Security`

- **Impact:** XSS defense-in-depth

- **Problem:** CSP is present but only sets `frame-ancestors 'self';` — no `default-src`, no `object-src 'none'`. Site signals show no auth, payments, or user content, so XSS attack surface is low and this is P3 per the rubric.
- **Solution:**
  - Add `default-src 'self'` and `object-src 'none'` as a baseline.
  - When you later add login or user content, move to a nonce + `'strict-dynamic'` policy:

```
Content-Security-Policy: default-src 'self'; object-src 'none';
script-src 'nonce-{random}' 'strict-dynamic'; base-uri 'none';
```

3B. Add HSTS preload and COOP/CORP `Security`

- **Impact:** Transport hardening, origin isolation
- **Problem:** HSTS lacks the `preload` directive; `Cross-Origin-Opener-Policy` and `Cross-Origin-Resource-Policy` are missing.
- **Solution:**

```
Strict-Transport-Security: max-age=31536000; includeSubDomains;
preload
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
```

  Submit the domain at hstspreload.org once preload is set.

3C. Add explicit width/height to 5 images `Accessibility`

- **Impact:** CLS
- **Problem:** 5 of 30 images lack explicit `width` / `height` attributes; CLS is currently 0.000 so impact is latent.
- **Solution:**
  - Add `width` and `height` attributes to every `<img>` so the browser can reserve space before load.

3D. Suppress server technology disclosure `Best Practices`

- **Impact:** Fingerprinting surface
- **Problem:** The `Server: Apache` response header discloses the web server software.
- **Solution:**

```
ServerTokens Prod
```

```
ServerSignature Off
```

Or strip the header at the reverse proxy.

3E. Move the WordPress login off /wp-login.php `Security`

- **Impact:** Brute-force / credential-stuffing surface
- **Problem:** The login form is served on the default `/wp-login.php` path, which is the first target for automated attacks.
- **Solution:**
  - Rename the login URL (e.g. WPS Hide Login) and add rate limiting / 2FA.
  - Keep xmlrpc.php blocked (already passing).

3F. Ship source maps for first-party JS `Performance`

- **Impact:** Debuggability, error tracking
- **Problem:** The `valid-source-maps` best-practices audit fails for large first-party JavaScript.
- **Solution:**
  - Emit `.map` files in the production build and serve them from the same origin.

3G. Add HSTS preload, COOP and CORP headers `Security`

- **Impact:** Transport hardening, origin isolation
- **Problem:** HSTS is set (`max-age=31536000; includeSubDomains`) but lacks the `preload` directive; `Cross-Origin-Opener-Policy` and `Cross-Origin-Resource-Policy` are both missing.
- **Solution:**

```
Strict-Transport-Security: max-age=63072000; includeSubDomains;
preload
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
```

  - Submit the domain at https://hstspreload.org once the preload directive is live.

3H. Reduce technology disclosure and harden the WordPress login path `Security`

- **Impact:** Attack-surface reduction
- **Problem:** The `Server: Apache` header discloses the web server, and the login form is served on the default `/wp-login.php` path (flagged as a warn in the WordPress checks).
- **Solution:**

- Suppress the banner: `ServerTokens Prod` and `ServerSignature Off` in Apache config.
- Move or restrict the login: rename via a security plugin, add IP allowlisting, or front it with a WAF.
- Enable rate limiting / 2FA on `/wp-login.php`.

3I. Trim unused JavaScript and ship source maps `Performance`

- **Impact:** TBT, TTI, transfer size
- **Problem:** `global.62444fa2566b44750ca1.min.js` carries 112 KB of unused JavaScript and produces a 65 ms long task; PSI also flags `valid-source-maps` as failing for large first-party JS.
- **Solution:**
  - Code-split the theme bundle so only the modules used on this template load.
  - Emit `.map` files in the production build so DevTools can attribute the unused bytes.
  - Re-check `legacy-javascript-insight` (est. 10 KiB saving) — drop the legacy transpile target if browser support allows.

3J. Reduce font payload and preload the critical face `Performance`

- **Impact:** Page weight, FCP, Speed Index
- **Problem:** Fonts are the single largest resource type at 105.3 KB across 3 requests, out of 228.3 KB total page weight.
- **Solution:**
  - Subset the webfonts to the glyphs actually used (Latin + punctuation) with `pyftsubset` or `glyphhanger`.
  - Preload the above-the-fold face:

```
<link rel="preload" href="/fonts/body.woff2" as="font"
type="font/woff2" crossorigin>
```

  - Prefer `font-display: swap` and drop any unused weights.

3K. Fix document caching and structured data `Best Practices`

- **Impact:** Repeat-visit TTFB, rich results eligibility
- **Problem:** The document response sets `cache-control: max-age=0`, and PSI's modeled SEO audit flags `structuredData` (the single JSON-LD block has type "Unknown").
- **Solution:**

    - Let WP Rocket serve the cached HTML with a positive `max-age` (or `s-maxage` at the CDN edge) instead of `max-age=0`.
    - Replace the generic JSON-LD block with a typed schema (`Organization`, `WebSite`, `WebPage`) so search engines can parse it.
    - Also address the failing `tapTargets` SEO audit by ensuring interactive elements are ≥44×44 px with adequate spacing.

3L. Disambiguate repeated link text `SEO`

- **Impact:** anchor text quality, screen-reader navigation
- **Problem:** 43 anchors include repeated vague text: "info@playandnope.com" ×3, "home" ×2, "work" ×2, "about" ×2, "contact" ×2, "design & innovation" ×2.
- **Solution:**
    - Add `aria-label` to icon/logo links, e.g. `<a href="/" aria-label="Play and Nope home">`.
    - Differentiate footer/nav duplicates with context ("Contact — Play and Nope" vs "Contact — NOPE Creative").

3M. Verify color contrast on the 23 undetermined nodes `Accessibility`

- **Impact:** WCAG 1.4.3
- **Problem:** axe could not determine contrast for 23 nodes (alpha/gradient backgrounds) and flagged 1 `link-in-text-block` node.
- **Solution:**
    - Sample the rendered colors in DevTools and confirm ≥4.5:1 for body text and ≥3:1 for large text.
    - Ensure in-text links are distinguishable by more than color alone (underline or weight).

3N. Improve tap target sizing `Performance`

- **Impact:** mobile usability
- **Problem:** PSI's `tapTargets` modeled SEO audit fails.
- **Solution:**
    - Ensure every interactive element is at least 44×44 px with adequate spacing (WCAG 2.5.8).
    - Increase padding on footer/nav icon links rather than shrinking the icons.

3O. Complete HSTS preload and add COOP/CORP `Security`

- **Impact:** Transport hardening, origin isolation

- **Problem:** HSTS is `max-age=31536000; includeSubDomains` but lacks `preload`; Cross-Origin-Opener-Policy and Cross-Origin-Resource-Policy headers are both missing.
- **Solution:**

  ```
  Strict-Transport-Security: max-age=63072000; includeSubDomains;
  preload
  Cross-Origin-Opener-Policy: same-origin
  Cross-Origin-Resource-Policy: same-origin
  ```

  Submit the domain at hstspreload.org once the preload directive is live.

3P. Suppress the Server header `Security`

- **Impact:** Technology disclosure
- **Problem:** The response discloses `Server: Apache`, which the security-basics checklist flags as a warn for technology disclosure.
- **Solution:**

  ```
  ServerTokens Prod
  ServerSignature Off
  ```

  Or strip/rewrite the header at the CDN/reverse-proxy layer.

3Q. Add structured data and enlarge tap targets `SEO`

- **Impact:** Rich results eligibility, mobile usability
- **Problem:** PSI's modeled audits report `structuredData` and `tapTargets` failing, and the HTML inventory finds no JSON-LD, no canonical, and no meta description.
- **Solution:**
  - Emit JSON-LD for the organisation and contact page:

    ```
    <script type="application/ld+json">
    {"@context":"https://schema.org","@type":"ContactPage"}</script>
    ```

  - Add a canonical link and a meta description.
  - Ensure interactive controls are at least 44×44 px with adequate spacing.

3R. Ship source maps for first-party bundles `Best Practices`

- **Impact:** Debuggability, error triage
- **Problem:** PSI's `valid-source-maps` audit fails — large first-party JavaScript (global.min.js, core.min.js) ships without source maps, and the console logs a 500 error.

- **Solution:**
  - Enable source maps in the build pipeline and deploy `.map` files alongside the minified bundles.
  - Keep maps out of the public path if you prefer, and upload them to your error-tracking service instead.