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

Website: https://playandnope.com/
Date: 12.10.2026
Audit Coverage: 93% — PageSpeed Insights (desktop): PSI request timed out after 120000ms; PageSpeed Insights: mobile: PSI request timed out after 120000ms; desktop: PSI request timed out after 120000ms
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: 63 / 100
Status: □ □ **Needs Improvement**

Site overall 63 is the mean of 3 pages. Scores range 50 (https://playandnope.com/digital-solutions-contact) → 88 (https://playandnope.com/privacy-policy). Weakest page: Performance is catastrophic on mobile: PSI Performance 35 with LCP 14.2 s, TBT 27,930 ms, Speed Index 9.79 s and Time to Interactive 43.1 s — the page is effectively unusable on a mid-tier phone. The root cause is JavaScript: 842 KB of script bytes, three separate copies of recaptcha__en.js (1.05 MB from gstatic, slowest request 2.78 s), 92 KB of unused first-party JS, and long tasks of 1.08 s (jQuery) and 801 ms (core.js). Accessibility (99) and SEO (100) are excellent — axe reports only 1 minor violation (presentation-role-conflict on the reCAPTCHA iframe) and the HTML inventory shows 0 images missing alt, 0 missing dimensions, and a clean heading outline. Security basics PASS, but the CSP is effectively a no-op (only `frame-ancestors 'self'`, no `default-src`, no `object-src 'none'`), HSTS lacks preload, COOP/CORP are missing, and 3 of 4 cross-origin scripts lack SRI. Desktop PSI timed out and mobile data was duplicated for both columns, so the mobile/desktop delta is unknown — confidence is medium. The score

lands in the Poor band because the performance failure is severe enough to overshadow otherwise strong accessibility and SEO.

Per-page scores

 Poor · https://playandnope.com/

Score	Performance	Accessibility	Best Practices	SEO	Security
52	37	100	100	92	PASS

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

Score	Performance	Accessibility	Best Practices	SEO	Security
88	97	100	100	100	PASS

— · https://playandnope.com/contact

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

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

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

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

Score	Performance	Accessibility	Best Practices	SEO	Security
50	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/	37 / —	5.04 s / —	0.000 / —

URL	Performance (M / D)	LCP (M / D)	CLS (M / D)
https://playandnope.com/privacy-policy	97 / 97	**2.11 s** / 632 ms	0.000 / **0.092**
https://playandnope.com/digital-solutions-contact	35 / —	13.28 s / —	0.028 / —

Optimization Checklist

3 of 3 passing — 3 pass · 0 warn · 0 fail · 5 n/a

Item	Status	Detail
Page caching plugin / CDN active	**Pass**	Caching plugin detected (WP Rocket)
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.
Reasonable number of image sizes	N/A	Too few raster images to evaluate srcset width variety.
JS scripts not blocking in	**Pass**	No render-blocking scripts in .

Fixes

Priority 1: Critical

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

1A. Eliminate 26.7 s of total blocking time from theme JavaScript `Performance`

- **Impact:** TBT, TTI, INP, Max Potential FID
- **Problem:** TBT is 26,700 ms (Lighthouse score 0.00, weight 30) and TTI is 42.8 s, driven by long tasks on core.c872dceb075db34864df.min.js (893 ms) and global.62444fa2566b44750ca1.min.js (694/252/237/234/224/224/221/219/210 ms), plus 92 KB of unused JS in global.min.js.
- **Solution:** The main thread is blocked for tens of seconds — this is the single biggest score lever.
  - Profile with the Performance panel and split the long tasks in `global.min.js` / `core.min.js` into smaller chunks (<50 ms each).
  - Remove the 92 KB of unused JS reported for `global.62444fa2566b44750ca1.min.js` (tree-shake, drop unused jQuery plugins, or load per-section).
  - Address the `forced-reflow-insight` failure: batch DOM reads/writes instead of interleaving them in loops.
  - Yield to the main thread between chunks (`scheduler.yield()` / `setTimeout`) so input can be handled.
  - Re-measure TBT and Max Potential FID (currently 890 ms) after each change.

1B. Cut LCP / FCP / Speed Index render delay `Performance`

- **Impact:** LCP, FCP, Speed Index
- **Problem:** LCP is 5.0 s (score 0.26), FCP 3.78 s (0.28) and Speed Index 7.85 s (0.23) while TTFB is only 42 ms — the delay is client-side rendering, not the server.
- **Solution:** Since TTFB is fast, the LCP element is being delayed by render-blocking work and font loading.
  - Preload the 3 font files (105.3 KB total) and use `font-display: swap`:

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

  - Inline critical CSS and defer the 3 external stylesheets.
  - Inspect the `lcp-breakdown-insight` and `network-dependency-tree-insight` failures to find the actual LCP element and its critical path.

    - Verify the hero renders without waiting on JS (the LCP element may be JS-injected).

1C. Fix color-contrast failure on navigation menu items `Accessibility`

- **Impact:** WCAG 1.4.3 (AA), screen-reader and low-vision usability
- **Problem:** axe-core reports 1 serious color-contrast violation on `.menu-item-14 > .navigation__link` and `.menu-item-18 > p`, with 9 additional nodes axe could not determine.
- **Solution:** Raise the foreground/background contrast of the affected nav links to at least 4.5:1 (3:1 for large text).
    - Inspect the computed colors in DevTools and darken the link color or lighten the background.
    - Re-run axe after the change to confirm the 9 "incomplete" nodes also pass.

1D. Stop loading reCAPTCHA on page load — it is the TBT driver `Performance`

- **Impact:** TBT, TTI, Speed Index, main-thread blocking
- **Problem:** TBT is 27,930 ms and TTI is 43.1 s. reCAPTCHA pulls 1.05 MB from gstatic across 5 requests, with three separate recaptcha__en.js loads (175 KB / 173 KB / 169 KB wasted) and the slowest request at 2.78 s.
- **Solution:** Load reCAPTCHA only when the user focuses or submits the contact form, not on initial render.
    - Remove the eager `<script src="https://www.google.com/recaptcha/api.js">` tag and inject it on first `focusin` / `submit` of the form.
    - Or switch to a lighter challenge (e.g. Cloudflare Turnstile) that ships far less JS.
    - If you must keep reCAPTCHA v3, use `render=explicit` with `execute()` on submit only, and drop the invisible badge variant that spawns the webworker.

1E. Cut long tasks from jQuery and theme bundles `Performance`

- **Impact:** TBT, INP, Max Potential FID (1,080 ms)
- **Problem:** Long tasks of 1.08 s (jquery.min.js), 801 ms (core.min.js) and 8 tasks of 300–460 ms (global.min.js) block the main thread; Max Potential FID is 1,080 ms.
- **Solution:**
    - Audit whether jQuery is still needed — modern vanilla JS can replace most of it.
    - Split `global.min.js` into route-level chunks so the contact page only loads what it uses.

- Move non-critical work (analytics, cookie-consent wiring, iframe manager) into `requestIdleCallback` or a `load` event handler.
- Remove the 92 KB of unused code flagged in `global.62444fa2566b44750ca1.min.js`.

1F. Fix the 14.2 s LCP `Performance`

- **Impact:** LCP, Speed Index, FCP
- **Problem:** LCP is 14.2 s on mobile (Lighthouse score 0.00, weight 25) and the `lcp-breakdown-insight` audit failed; Speed Index is 9.79 s.
- **Solution:**
  - Identify the LCP element in DevTools Performance panel (likely a hero image or a JS-rendered heading).
  - If it is an image, preload it: `<link rel="preload" as="image" href="/hero.webp" fetchpriority="high">`.
  - If it is text, ensure the font is preloaded and `font-display: swap` is set.
  - Remove the render-blocking scripts still in `<head>` (2 detected) and inline critical CSS.

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Enable browser caching for the HTML document `Performance`

- **Impact:** Repeat-visit TTFB, LCP, FCP
- **Problem:** The document response sends `cache-control: max-age=0` with `expires` equal to the response date, so the HTML is re-fetched on every navigation despite WP Rocket being detected.
- **Solution:** Set a short but non-zero browser TTL for HTML and let the CDN/page cache handle freshness:

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

  In WP Rocket, enable Browser Caching and confirm the rule is not overridden by the theme or a plugin.

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

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is present but only sets `frame-ancestors 'self';` — `default-src` is missing and `object-src` is not `'none'`, so no script or resource restrictions apply.

- **Solution:** Add the two missing directives at minimum, then move toward a nonce-based policy:

  ```
  Content-Security-Policy: default-src 'self'; object-src 'none';
  base-uri 'self'; frame-ancestors 'self';
  ```

  Target a nonce + `'strict-dynamic'` policy rather than a flat allowlist:

  ```
  script-src 'nonce-{random}' 'strict-dynamic' https: 'unsafe-
  inline';
  ```

  Roll it out in Report-Only mode first and watch for violations from the jsdelivr CDN scripts.

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

- **Impact:** Supply-chain integrity of third-party JS
- **Problem:** 2 of 2 cross-origin scripts lack an `integrity` attribute — both are loaded from `cdn.jsdelivr.net` (6 requests, 28.6 KB).
- **Solution:** Pin versions and add SRI hashes to the jsdelivr assets:

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

  Generate hashes with `openssl dgst -sha384 -binary file.js | openssl base64 -A`, or self-host the two libraries.

2D. Verify the 19 unresolved color-contrast nodes `Accessibility`

- **Impact:** WCAG 1.4.3 contrast, 1.4.1 use of color
- **Problem:** axe reports 0 violations but flags 19 nodes as "incomplete" for color-contrast and 1 node for link-in-text-block — it could not determine pass/fail, typically due to gradients, alpha transparency or background images.
- **Solution:**
  - Manually check each of the 19 nodes with DevTools' contrast checker against the rendered background.
  - Ensure text over images/gradients has a solid or scrimmed backdrop meeting 4.5:1 (3:1 for large text).
  - For the link-in-text-block node, add a non-color affordance (underline) so links are distinguishable without relying on color.

2E. Replace vague repeated link text `SEO`

- **Impact:** SEO `link-text` audit, screen-reader link purpose (WCAG 2.4.4)
- **Problem:** The `link-text` audit fails with 4 links lacking descriptive text, and the inventory shows "read more" repeated 4 times.
- **Solution:** Make link text describe its destination:

  ```
  <!-- before -->
  <a href="/case-study-x">read more</a>
  <!-- after -->
  <a href="/case-study-x">Read the Fintech case study</a>
  ```

  If the visible text must stay short, disambiguate with `aria-label` or visually hidden text.

2F. Add explicit width/height to the 5 unsized images `Performance`

- **Impact:** CLS, layout stability
- **Problem:** 5 of 30 images lack explicit `width` / `height` attributes, so the browser cannot reserve space before they load (CLS is currently 0.000, but this is fragile).
- **Solution:** Add intrinsic dimensions to every `<img>` and let CSS control the rendered size:

  ```
  <img src="…" alt="…" width="360" height="270" loading="lazy"
  decoding="async">
  ```

  Also add `aspect-ratio` in CSS for any image whose rendered box differs from its intrinsic ratio.

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

- **Impact:** XSS defense-in-depth, script/resource loading restrictions
- **Problem:** CSP is only `frame-ancestors 'self';` — no `default-src` and no `object-src 'none'`, so no script or resource restrictions are enforced. Security basics marks "CSP is strict" as fail.
- **Solution:** Add a nonce-based policy with `'strict-dynamic'` rather than a flat allowlist:

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

  Note: the site exposes a WordPress login at `/wp-login.php`, so a real auth surface exists even though the inferred signal says otherwise — treat CSP as P2,

not P3.

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

- **Impact:** Supply-chain tampering risk on jsdelivr-hosted scripts
- **Problem:** Security basics reports 2 of 2 cross-origin scripts lack an `integrity` attribute; both are loaded from `cdn.jsdelivr.net` (iframemanager and cookieconsent).
- **Solution:** Pin a version and add SRI hashes:

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

  Better: self-host these two libraries so they fall under your own CSP and cache.

2I. Enable browser caching on the HTML document `Performance`

- **Impact:** Repeat-visit TTFB, document latency
- **Problem:** The document response is served with `cache-control: max-age=0` and an `expires` value in the past, so the HTML is revalidated on every visit even though WP Rocket is active. The document request was the slowest at 700 ms.
- **Solution:** Configure WP Rocket (or the origin) to send a positive max-age with stale-while-revalidate for HTML:

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

  Keep `max-age=0` only for logged-in or personalized responses.

2J. Complete HSTS and add COOP/CORP `Security`

- **Impact:** Transport hardening, cross-origin isolation
- **Problem:** HSTS is `max-age=31536000; includeSubDomains` but lacks `preload`; Cross-Origin-Opener-Policy and Cross-Origin-Resource-Policy are both 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 to hstspreload.org once the preload directive is live.

2K. Strengthen the Content-Security-Policy `Security`

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is only `frame-ancestors 'self';` — no `default-src`, no `object-src 'none'`, and no script/resource restrictions. The site has a contact form with a `<textarea>` (user-content signal), so a real XSS surface exists.
- **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://www.google.com https://www.gstatic.com;
    object-src 'none';
    base-uri 'none';
    frame-ancestors 'self';
    upgrade-insecure-requests;
  ```

  Roll it out in `Content-Security-Policy-Report-Only` first and fix violations before enforcing.

2L. Complete HSTS 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 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.

2M. Label the unlabeled textarea and verify contrast `Accessibility`

- **Impact:** WCAG 1.3.1, 1.4.3, 4.1.2
- **Problem:** Form 1 contains a `<textarea>` with no label, and axe could not determine contrast on 23 nodes (alpha/gradient backgrounds).
- **Solution:**
  - Add a `<label for="…">` or `aria-label` to the textarea.
  - Manually check the 23 incomplete color-contrast nodes with a contrast checker; axe skips elements with alpha transparency or gradients.
  - Also verify the 1 `link-in-text-block` node is distinguishable without color.

2N. Fix the reCAPTCHA console error `Best Practices`

- **Impact:** Console cleanliness, Best Practices score
- **Problem:** Browser runtime logged `requestStorageAccess: Permission denied.` from the reCAPTCHA anchor iframe, and PSI's `errors-in-console` audit failed.
- **Solution:**
  - This is a known reCAPTCHA third-party storage-access issue in partitioned-cookie contexts.
  - Loading reCAPTCHA lazily (see P1 fix) usually suppresses it.
  - If it persists, check whether a consent-management script is blocking storage access before reCAPTCHA initializes.

Priority 3: Best Practice

Recommended for long-term maintainability.

3A. Complete advanced header hardening `Security`

- **Impact:** Origin isolation, HSTS preload eligibility, tech disclosure
- **Problem:** HSTS lacks the `preload` directive, Cross-Origin-Opener-Policy and Cross-Origin-Resource-Policy are missing, `X-Permitted-Cross-Domain-Policies` is absent, and the `Server: Apache` header discloses the stack.
- **Solution:**

  ```
  Header always set Strict-Transport-Security "max-age=31536000;
  includeSubDomains; preload"
  Header always set Cross-Origin-Opener-Policy "same-origin"
  Header always set Cross-Origin-Resource-Policy "same-origin"
  Header always set X-Permitted-Cross-Domain-Policies "none"
  ServerTokens Prod
  ServerSignature Off
  ```

  Submit the domain to hstspreload.org only after confirming every subdomain is HTTPS-only.

3B. Harden the WordPress login path `Security`

- **Impact:** Brute-force and credential-stuffing exposure
- **Problem:** The login form is served on the default `/wp-login.php` path, which is the first target for automated credential attacks.
- **Solution:**

    - Move or rename the login URL (e.g. WPS Hide Login) and keep a non-guessable fallback.
    - Add rate limiting and CAPTCHA on the login endpoint.
    - Enforce strong passwords and 2FA for all accounts with publishing rights.

3C. Fix structured data and tap targets `SEO`

- **Impact:** Rich results eligibility, mobile usability
- **Problem:** The `structuredData` and `tapTargets` modeled audits both fail, and the single JSON-LD block is parsed with type "Unknown".
- **Solution:**
    - Validate the JSON-LD block with Google's Rich Results Test and set an explicit `@type` (e.g. `Organization`, `WebSite`).
    - Ensure interactive elements meet the 44×44 px minimum touch target size (WCAG 2.5.8) with adequate spacing.

3D. Ship source maps for first-party JavaScript `Best Practices`

- **Impact:** Debuggability, error attribution
- **Problem:** The `valid-source-maps` audit fails — large first-party bundles such as `global.62444fa2566b44750ca1.min.js` and `core.c872dceb075db34864df.min.js` ship without source maps.
- **Solution:** Emit `.map` files alongside the minified bundles and reference them with a `//# sourceMappingURL=` comment, or upload them to your error-tracking service. Exclude the maps from the public bundle if you prefer not to expose source.

3E. Trim unused JavaScript and long tasks `Performance`

- **Impact:** TBT, TTI, main-thread work
- **Problem:** PSI flags 112 KB of unused JS in `global.62444fa2566b44750ca1.min.js`, which also produces a 101 ms long task; TTI is 4.1 s.
- **Solution:**
    - Code-split the theme bundle so only the modules used on this page load.
    - Drop `jquery-migrate.min.js` if no legacy plugin depends on it.
    - Audit the 22 scripts for anything not needed on a static privacy-policy page.

3F. Reduce font payload `Performance`

- **Impact:** Page weight, FCP

- **Problem:** 3 font files account for 105.3 KB of the 228.3 KB total page weight (46%), and FCP is 1.96 s on mobile.
- **Solution:**
  - Subset fonts to the glyphs actually used and serve WOFF2 only.
  - Preload the primary font and set `font-display: swap`.
  - Consider a system-font stack for body copy.

3G. Reduce technology disclosure and add X-Permitted-Cross-Domain-Policies `Best Practices`

- **Impact:** Fingerprinting surface, legacy cross-domain policy
- **Problem:** The response sends `Server: Apache`, and `X-Permitted-Cross-Domain-Policies` is missing. Security basics flags technology disclosure as a warn.
- **Solution:**

```
ServerTokens Prod
ServerSignature Off
Header always set X-Permitted-Cross-Domain-Policies "none"
```

3H. Disambiguate repeated link text and fix structured data `SEO`

- **Impact:** Link purpose (WCAG 2.4.4), rich-result eligibility
- **Problem:** 10 link texts repeat verbatim (e.g. "home" ×2, "contact" ×2, "info@playandnope.com" ×2), and PSI flags `structuredData` and `tapTargets` as failing modeled SEO audits; the single JSON-LD block has type "Unknown".
- **Solution:**
  - Add `aria-label` or visually-hidden context to duplicated nav/footer links.
  - Replace the JSON-LD block with a valid Schema.org type (e.g. `WebPage`, `Organization`).
  - Ensure interactive elements meet the 44×44 px touch-target minimum.

3I. Fix the two W3C validation errors on hidden inputs `Best Practices`

- **Impact:** HTML validity
- **Problem:** W3C reports 2 errors: `<input type='hidden' autocomplete='off' class='gform_hidden' …>` at line 896 — hidden inputs must not carry `autocomplete="on"` or `"off"`.
- **Solution:** Remove the `autocomplete` attribute from Gravity Forms hidden inputs, or filter it out:

```
add_filter( 'gform_field_content', function ( $content, $field ) {
    if ( 'hidden' === $field->type ) {
        $content = preg_replace( '/\sautocomplete=["\'](on|off)
["\']/i', '', $content );
    }
    return $content;
}, 10, 2 );
```

3J. Address failing tapTargets and structuredData audits `SEO`

- **Impact:** Mobile usability, rich results
- **Problem:** PSI's modeled `tapTargets` and `structuredData` audits both failed; the single JSON-LD block is typed `Unknown`.
- **Solution:**
  - Ensure interactive controls are at least 44×44 px with adequate spacing.
  - Replace the untyped JSON-LD with a proper `@type` (e.g. `Organization`, `LocalBusiness`, `ContactPage`) and validate with Google's Rich Results Test.

3K. Reduce technology disclosure and move the login path `Security`

- **Impact:** Attack-surface reduction
- **Problem:** The `Server: Apache` header is exposed, and the login form is served on the default `/wp-login.php` path.
- **Solution:**
  - Suppress the banner: `ServerTokens Prod` and `ServerSignature Off` in Apache.
  - Move the login to a custom path (e.g. WPS Hide Login) and rate-limit it.
  - Note: the Security basics verdict is PASS, but per the disclaimer that only means the basics are in place — it does **not** mean the site is secure: application logic, authentication, plugins, server configuration and dependencies are not tested here.

3L. Disambiguate repeated link text `SEO`

- **Impact:** Screen-reader and crawler link context
- **Problem:** 43 anchors include repeated vague text: "info@playandnope.com" ×3, "home" ×2, "work" ×2, "about" ×2, "contact" ×2, plus "play and nope alliance logo" ×2.
- **Solution:** Add `aria-label` or visually hidden context to repeated links, e.g. `<a href="/work" aria-label="View our work portfolio">work</a>`, and give the logo link a descriptive label rather than the file name.