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

Website: https://playandnope.com/
Date: 13.10.2026
Audit Coverage: 88% — W3C Validator: W3C HTTP 429
Confidence: medium

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: 76 / 100
Status: □ □ **Needs Improvement**

Site overall 76 is the mean of 5 pages. Scores range 65 (https://playandnope.com/contact) → 90 (https://playandnope.com/privacy-policy). Weakest page: Mobile PSI is 44 against desktop 77 — a 33-point gap driven almost entirely by reCAPTCHA: 1.42 MB served from gstatic.com, 10 long tasks (515/471/461 ms), and 176 KB unused JS, producing LCP 9.2 s and TBT 2.17 s on mobile. Desktop is healthy (LCP 1.2 s, TBT 227 ms) but shows CLS 0.247, above the 0.1 threshold, versus 0.000 on mobile. Everything else is strong: axe reports only 1 minor violation, PSI accessibility 99, SEO 100, no raster images (5 SVGs, all with dimensions and lazy loading), gzip compression, WP Rocket caching, TTFB 3 ms, and a PASS on the security basics checklist. Security is not clean, however — CSP is frame-ancestors-only with no default-src or object-src 'none', HSTS lacks preload, COOP/CORP are absent, and 3 of 4 cross-origin scripts lack SRI. W3C validation failed with HTTP 429, so HTML validity is unscored and confidence is medium.

Per-page scores

□ **Needs Improvement** · https://playandnope.com/

Score	Performance	Accessibility	Best Practices	SEO	Security
78	90	100	100	92	PASS

 Excellent · https://playandnope.com/privacy-policy

Score	Performance	Accessibility	Best Practices	SEO	Security
90	94	100	100	100	PASS

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

Score	Performance	Accessibility	Best Practices	SEO	Security
65	44	99	96	100	PASS

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

Score	Performance	Accessibility	Best Practices	SEO	Security
72	74	98	100	100	PASS

 Needs Improvement · https://playandnope.com/digital-solutions-contact

Score	Performance	Accessibility	Best Practices	SEO	Security
75	70	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/	90 / **80**	**3.10 s** / 486 ms	0.000 / **0.456**
https://playandnope.com/privacy-policy	**94** / 97	**2.35 s** / 618 ms	0.085 / **0.098**
https://playandnope.com/contact	**44** / 77	**9.17 s** / 1.18 s	0.000 / **0.247**

URL	Performance (M / D)	LCP (M / D)	CLS (M / D)
https://playandnope.com/full-stack-digital-agency	**74** / 97	**2.20 s** / 595 ms	**0.619** / 0.098
https://playandnope.com/digital-solutions-contact	70 / **61**	**11.68 s** / 1.73 s	0.000 / **0.246**

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 the 0.456 desktop CLS `Performance`

- **Impact:** CLS (desktop), layout stability
- **Problem:** Desktop CLS is 0.456 (heavy-penalty band) while mobile CLS is 0.000, and the HTML inventory shows 5 images without explicit width/height plus a PSI `forced-reflow-insight` failure.
- **Solution:**
  - Add explicit `width` / `height` (or `aspect-ratio`) to the 5 images missing dimensions so the browser reserves space before load.
  - Reserve space for web fonts: `font-display: optional` or `size-adjust` / `ascent-override` on the 3 font files (105.3 KB total).
  - Fix the forced reflow: batch DOM reads/writes in `core.c872dceb075db34864df.min.js` (58 ms long task) instead of interleaving them.
  - Audit desktop-only breakpoints — the shift is not reproducing on mobile, so it is likely a desktop grid/hero rule.

1B. Fix color-contrast violation on navigation description `Accessibility`

- **Impact:** WCAG 1.4.3, axe serious violation
- **Problem:** axe reports 1 serious `color-contrast` violation on `.menu-item-75 > .navigation__description`, plus 2 nodes it could not determine (likely alpha/gradient).
- **Solution:** Raise the text color to meet ≥4.5:1 for normal text (≥3:1 for large). Example:

  ```
  .navigation__description { color: #4a4a4a; /* was lighter */ }
  ```

  Manually verify the 2 "incomplete" nodes — axe skips elements with alpha-transparency or gradients.

1C. Stop reCAPTCHA from blocking the main thread on load `Performance`

- **Impact:** TBT (2.17 s), LCP (9.2 s), TTI (12.4 s), page weight
- **Problem:** reCAPTCHA accounts for 1.42 MB of the 1.90 MB page (10 requests from www.gstatic.com), 176 KB of unused JS, and 10 long tasks up to 515 ms —

the single largest contributor to mobile TBT 2.17 s and LCP 9.2 s.

- **Solution:** Load the challenge only when the user engages with the form, not on page load:

```
const form = document.querySelector('form');
form.addEventListener('focusin', () => {
  const s = document.createElement('script');
  s.src = 'https://www.google.com/recaptcha/api.js?
render=explicit&hl=en';
  s.defer = true;
  s.onload = () => grecaptcha.render('recaptcha-slot', { sitekey:
'…' });
  document.head.appendChild(s);
}, { once: true });
```

  - Or switch to a lighter challenge (Cloudflare Turnstile / hCaptcha) which ships a fraction of the JS.
  - Keep `render=explicit` so nothing renders until you call `grecaptcha.render()`.

1D. Fix the 7.2 s gap between FCP and LCP on mobile `Performance`

- **Impact:** LCP, FCP, Speed Index
- **Problem:** Mobile FCP is 2.0 s but LCP is 9.2 s — the LCP element paints 7.2 s after first paint, which points to a JS-injected hero rather than a server-rendered one.
- **Solution:**
  - Confirm the LCP element in DevTools (Performance → Timings → LCP) — `lcp-breakdown-insight` failed, so this needs a manual trace.
  - Ensure the hero heading/image is present in the server-rendered HTML, not created by the theme's `core.js` / `global.js` after hydration.
  - Preload the LCP font (Roboto is loaded from fonts.gstatic.com, 39.2 KB):

```
<link rel="preload" as="font" type="font/woff2" crossorigin
      href="/wp-content/.../roboto.woff2">
```

  - Remove the 2 render-blocking scripts reported in the HTML inventory.

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

- **Impact:** XSS defense-in-depth on a form that accepts user input
- **Problem:** CSP is `frame-ancestors 'self';` only — no `default-src`, no `object-src 'none'`, and no script/resource restrictions. The page contains a

`<textarea>` (user-generated content signal = yes), so a successful XSS could persist into submitted content.

- **Solution:** Deploy a nonce + `'strict-dynamic'` policy 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;
  style-src 'self' 'unsafe-inline' https://fonts.googleapis.com;
  font-src 'self' https://fonts.gstatic.com;
  img-src 'self' data: https:;
  object-src 'none';
  base-uri 'none';
  frame-ancestors 'self';
  form-action 'self';
```

  - Emit a fresh nonce per response and add it to every `<script>` tag.
  - Roll out in `Content-Security-Policy-Report-Only` first to catch breakage from reCAPTCHA, jsDelivr and Google Fonts.

1F. Eliminate the 0.619 mobile CLS caused by the cookie-consent banner `Performance`

- **Impact:** CLS (mobile 0.619 vs desktop 0.098), mobile Performance score (74)
- **Problem:** PSI's cumulative-layout-shift audit scores 0.10 and attributes the shift to body.wp-singular > div#cc-main > div.cm-wrapper > div.cm — the cookie-consent banner injected after first paint.
- **Solution:**
  - Reserve space for the banner before it renders: give `#cc-main` a fixed `min-height` or render it in a `position: fixed` overlay that does not reflow the document.
  - Load the cookieconsent CSS in the initial `<head>` (currently it comes from cdn.jsdelivr.net as a separate stylesheet) so the banner's box is sized on first paint.
  - Avoid injecting the banner DOM above existing content; append it as the last child of `<body>`.
  - Re-run PSI mobile after the change and confirm CLS < 0.1.

1G. Fix color contrast on navigation descriptions `Accessibility`

- **Impact:** WCAG 1.4.3 (AA), axe serious violation, Accessibility score
- **Problem:** axe reports a serious color-contrast violation on `.menu-item-74 > .navigation__description` and `.menu-item-75 > .navigation__description`; 2 further nodes were flagged as incomplete.

- **Solution:**
  - Raise the text/background contrast ratio to at least 4.5:1 for normal text (3:1 for ≥18.66 px bold or ≥24 px).
  - Check the computed colors in DevTools; the descriptions likely sit on a light background with a mid-grey text.
  - Verify the 2 incomplete nodes manually — axe skips elements with alpha transparency or gradients.

1H. Fix mobile LCP of 11.7 s `Performance`

- **Impact:** LCP, TTI, Speed Index
- **Problem:** Mobile LCP is 11.7 s (desktop 1.7 s) and TTI 11.9 s — the LCP element appears ~10 s after FCP (1.99 s), indicating a late-rendered hero or lazy-loaded above-fold asset.
- **Solution:** Identify the LCP element in DevTools Performance panel, then:
  - Preload the hero image/font with `<link rel="preload" as="image" fetchpriority="high">`.
  - Remove `loading="lazy"` from any above-the-fold image.
  - Inline critical CSS and defer non-critical stylesheets.
  - Investigate the `network-dependency-tree-insight` and `forced-reflow-insight` failures flagged by PSI.

1I. Defer reCAPTCHA to cut 1.4 MB of third-party JS `Performance`

- **Impact:** TBT, LCP, page weight
- **Problem:** gstatic.com serves 1.42 MB across 10 requests; recaptcha__en.js shows 176+170+170 KB unused JS and 7 long tasks (121/105/91/81/67/58 ms).
- **Solution:** Load reCAPTCHA only when the user interacts with the form:

  ```
  form.addEventListener('focusin', () => loadRecaptcha(), { once:
  true });
  ```

  Or switch to reCAPTCHA v3 with `render=explicit` and lazy init. Consider a lighter alternative (Cloudflare Turnstile, hCaptcha) if the form is low-risk.

1J. Fix console errors from reCAPTCHA integration `Best Practices`

- **Impact:** Best Practices (errorsInConsole), user trust
- **Problem:** 5 console events including a CORS-blocked `styles__ltr.css` (net::ERR_FAILED), two `requestStorageAccess: Permission denied`, and a failed XHR.

- **Solution:** The errors originate from Google's reCAPTCHA iframe. Verify the site key/domain configuration in the reCAPTCHA admin console, ensure the correct `render=explicit` callback is wired, and check that no CSP or cookie policy is blocking the iframe's storage access.

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Cut mobile LCP (3.1 s) and Speed Index (4.59 s) `Performance`

- **Impact:** LCP, FCP, Speed Index
- **Problem:** Mobile LCP is 3.1 s and Speed Index 4.59 s versus 0.5 s / 636 ms on desktop; PSI flags `legacy-javascript-insight` with ~12 KiB of avoidable polyfills and 3 font files totalling 105.3 KB.
- **Solution:**
  - Preload the LCP element and its font: `<link rel="preload" as="font" type="font/woff2" href="…" crossorigin>`.
  - Subset fonts to the glyphs actually used and self-host them instead of loading 3 full families.
  - Drop legacy polyfills by raising the browserslist target:

```
"browserslist": ["> 0.5%", "last 2 versions", "not dead"]
```

  - Add `font-display: swap` so text paints before the font arrives.

2B. Replace non-descriptive link text `SEO`

- **Impact:** SEO (link-text audit), screen-reader link lists
- **Problem:** The PSI `link-text` audit scores 0.00 with 4 links lacking descriptive text, and the inventory shows "read more" repeated 4 times.
- **Solution:**
  - Rewrite each "read more" anchor to name its destination, e.g. `<a href="/services/design/">Read more about Design & Innovation</a>`.
  - If the visible text must stay short, disambiguate with `aria-label`:

```
<a href="/case-study/x" aria-label="Read the case study for X">read more</a>
```

2C. Fix tap targets and structured data `SEO`

- **Impact:** SEO score (92), mobile usability, rich results

- **Problem:** PSI reports failing SEO audits `tapTargets` and `structuredData`, and the single JSON-LD block is typed as "Unknown".
- **Solution:**
  - Ensure every interactive element is at least 44×44 px with adequate spacing on mobile.
  - Replace the untyped JSON-LD with a valid `Organization` / `WebSite` graph:

```
{"@context":"https://schema.org","@type":"Organization","name":"Pla
y and Nope","url":"https://playandnope.com/"}
```

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

- **Impact:** Supply-chain integrity (jsdelivr)
- **Problem:** 2 of 2 cross-origin scripts (cookieconsent and iframemanager from cdn.jsdelivr.net) lack an `integrity` attribute.
- **Solution:** Pin exact versions 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 still, self-host these two libraries so the third-party origin is removed entirely.

2E. Complete HSTS preload and add COOP/CORP `Security`

- **Impact:** Transport hardening, origin isolation
- **Problem:** HSTS is `max-age=31536000; includeSubDomains` but lacks `preload`, and 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"
```

  Then submit the domain at hstspreload.org.

2F. Strengthen Content-Security-Policy `Security`

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

- **Problem:** CSP is `frame-ancestors 'self';` only — no `default-src`, no `object-src 'none'`, so no script or resource restrictions are enforced.
- **Solution:** Add a nonce-based policy rather than a flat allowlist:

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

  Note: this is a WordPress site with a login at `/wp-login.php`, so a real XSS surface exists even though the audited page is a privacy policy.

2G. Add HSTS preload, COOP, and CORP headers `Security`

- **Impact:** Transport security, 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
  ```

  Then submit the domain at hstspreload.org.

2H. Move WordPress login off the default /wp-login.php path `Security`

- **Impact:** Brute-force / credential-stuffing surface
- **Problem:** Security basics flags `Login not on default path` as a warn — the login form is served on the default `/wp-login.php` path.
- **Solution:**
  - Rename the login URL (e.g., WPS Hide Login or a custom rewrite).
  - Add rate limiting / fail2ban and enable 2FA for all admin accounts.

2I. Reduce FCP and document latency `Performance`

- **Impact:** FCP, LCP, Speed Index
- **Problem:** Mobile FCP is 1.99 s (above the 1.8 s threshold); `document-latency-insight` estimates 570 ms savings and `network-dependency-tree-insight` is flagged high.
- **Solution:**
  - Add `<link rel="preconnect" href="https://cdn.jsdelivr.net" crossorigin>` (currently 0 preconnects).

- Inline critical CSS and defer the 3 external stylesheets.
- Flatten the dependency chain so the document is not blocked behind the jsdelivr scripts.

2J. Eliminate cookie-consent layout shift `Performance`

- **Impact:** CLS (0.085 mobile / 0.098 desktop)
- **Problem:** Lighthouse attributes the layout shift to `body.privacy-policy > div#cc-main > div.cm-wrapper > div.cm` — the cookie-consent banner.
- **Solution:** Reserve space for the banner or position it out of flow:

  ```
  #cc-main .cm-wrapper { position: fixed; bottom: 0; left: 0; right: 0; }
  ```

  Avoid injecting the banner into the document flow after first paint.

2K. Resolve the reCAPTCHA console errors and failed stylesheet `Best Practices`

- **Impact:** Console errors, Best Practices score, form reliability
- **Problem:** 5 console events including `requestStorageAccess: Permission denied` (×2), a CORS-blocked `styles__ltr.css` from gstatic (`net::ERR_FAILED`), and a failed preload — this is why PSI's `errors-in-console` audit fails.
- **Solution:**
  - The site runs `cookieconsent` + `iframemanager` from jsDelivr; the consent manager is likely blocking the third-party storage reCAPTCHA needs. Load reCAPTCHA only after consent is granted, or use a consent-exempt challenge.
  - The `styles__ltr.css` CORS failure is internal to reCAPTCHA — verify the site key's allowed domains in the Google reCAPTCHA admin console and that the key matches the production host.
  - Re-test the form end-to-end after the change; a broken challenge silently blocks submissions.

2L. Fix desktop CLS 0.247 `Performance`

- **Impact:** CLS (desktop)
- **Problem:** Desktop CLS is 0.247 (warning band, >0.1) while mobile CLS is 0.000 — a layout shift that only manifests at desktop viewport widths.
- **Solution:**
  - Reserve space for late-injected UI (cookie banner, reCAPTCHA badge) with fixed `min-height` / `aspect-ratio` instead of letting them push content.

    - Never insert elements above existing content after load; use `position: fixed` for the consent banner.
    - Check the theme's `global.js` for anything that toggles a class affecting layout on `DOMContentLoaded`.

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

- **Impact:** WCAG 1.3.1 / 4.1.2, 1.4.3
- **Problem:** The contact form has one `<textarea>` with no label (the other text inputs are labeled), and axe returned 3 'incomplete' color-contrast nodes it could not resolve.
- **Solution:**
    - Add an explicit label:

```
<label for="message">Message</label>
<textarea id="message" name="message"></textarea>
```

    - Manually check the 3 unresolved contrast nodes (axe skips alpha/gradient text) against 4.5:1 for body text and 3:1 for large text.

2N. Fix heading-order skip (h1 → h3) `Accessibility`

- **Impact:** WCAG 1.3.1, axe moderate violation, PSI heading-order audit
- **Problem:** The heading outline jumps from h1 "Get to know us" straight to h3 "Full-stack digital agency powered by curiosity"; axe flags `.hero__description > h3` and PSI's heading-order audit scores 0.00.
- **Solution:**
    - Change the hero `<h3>` to `<h2>` so the sequence descends one level at a time.
    - Audit the remaining h3s (Design, Development, Other, etc.) and promote any that are top-level sections to h2.
    - Keep exactly one h1 per page.

2O. Fix document latency and page caching `Performance`

- **Impact:** TTFB, document latency (est. 660 ms savings), FCP
- **Problem:** PSI's document-latency-insight estimates 660 ms of savings, and the document response carries `cache-control: max-age=0` with an `expires` header equal to the request time — so every visit revalidates despite WP Rocket being detected.
- **Solution:**

- Confirm WP Rocket's page cache is actually serving this URL for anonymous visitors (check for the `<!-- This is WP Rocket ... -->` footer comment).
- Set a real `Cache-Control` for HTML, e.g. `public, max-age=600, s-maxage=86400, stale-while-revalidate=60`.
- Add `<link rel="preconnect">` for `https://cdn.jsdelivr.net` (currently 0 preconnects across 45 anchors).
- Warm the cache on deploy so the first visitor does not pay the cold-start cost.

2P. Reduce image delivery weight `Performance`

- **Impact:** LCP, page weight (552.5 KB total, 287.5 KB images)
- **Problem:** PSI's image-delivery-insight estimates 330 KiB of savings; the browser runtime capture shows 8 image requests totalling 287.5 KB out of 552.5 KB total page weight.
- **Solution:**
  - Serve WebP/AVIF with a `<picture>` fallback for the raster images that are actually loaded.
  - Add `srcset` / `sizes` so mobile devices do not download desktop-sized assets.
  - Preload the LCP image with `<link rel="preload" as="image" fetchpriority="high">`.
  - Re-check the hero image visually — the checklist heuristic could not identify a raster hero (44 SVGs excluded).

2Q. Harden 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 are enforced. The site runs WordPress with a login form at /wp-login.php, so an auth surface exists in practice 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';
```

    - Generate the nonce per request in the WordPress theme and inject it into every `<script>` tag.
    - Roll out in `Content-Security-Policy-Report-Only` first and fix violations before enforcing.

2R. Fix desktop CLS of 0.246 `Performance`

- **Impact:** CLS
- **Problem:** Desktop CLS is 0.246 (mobile 0.000) — above the 0.1 good threshold and likely caused by late-loading content or fonts.
- **Solution:** Reserve space for dynamic content:
    - Set explicit `width` / `height` or `aspect-ratio` on all media.
    - Preload the web font and use `font-display: optional` or `swap` with a matched fallback to avoid FOUT reflow.
    - Avoid injecting consent banners or cookie UI above existing content.

2S. Reduce desktop TBT of 448 ms `Performance`

- **Impact:** TBT, INP
- **Problem:** Desktop TBT is 448 ms (mobile 175 ms) with 7 long tasks, mostly from recaptcha__en.js.
- **Solution:** Break up long tasks and defer non-critical JS:
    - Load reCAPTCHA on interaction (see P1).
    - Split long handlers with `scheduler.yield()` or `setTimeout`.
    - Audit the 42 scripts; 2 are still render-blocking.

2T. Harden 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'`, so scripts and plugins are unrestricted.
- **Solution:** Add a nonce-based policy:

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

  The site has a contact form (user-content signal), so XSS hardening is worth prioritizing.

2U. Add Cross-Origin-Opener-Policy and Cross-Origin-Resource-Policy `Security`

- **Impact:** Origin isolation, cross-origin read protection
- **Problem:** COOP and CORP headers are both missing per the security headers scan.
- **Solution:**

```
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
```

2V. Fix failing tap-target audit `SEO`

- **Impact:** Mobile usability, SEO
- **Problem:** PSI flags `tapTargets` as failing on mobile.
- **Solution:** Ensure every interactive element is at least 44×44 px with adequate spacing:

```
.btn, nav a, button { min-width: 44px; min-height: 44px; }
```

2W. Fix structured data `SEO`

- **Impact:** Rich results, SEO
- **Problem:** PSI `structuredData` audit fails and the single JSON-LD block is typed `Unknown`.
- **Solution:** Add valid schema.org types (e.g. `Organization`, `LocalBusiness`, `ContactPage`) and validate with Google's Rich Results Test.

2X. Label the unlabeled textarea `Accessibility`

- **Impact:** WCAG 1.3.1, 4.1.2
- **Problem:** The contact form has a `<textarea>` with no associated label (hidden inputs are exempt).
- **Solution:**

```
<label for="message">Message</label>
<textarea id="message" name="message"></textarea>
```

2Y. Disambiguate repeated link text `Accessibility`

- **Impact:** WCAG 2.4.4
- **Problem:** Multiple anchors repeat the same text (e.g. "work" ×2, "about" ×2, "contact" ×2, "info@playandnope.com" ×3).
- **Solution:** Add `aria-label` or visually-hidden context to each repeated link:

```
<a href="/work" aria-label="View our work">work</a>
```

Priority 3: Best Practice

Recommended for long-term maintainability.

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

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is only `frame-ancestors 'self';` — no `default-src` and no `object-src 'none'`. Site signals show no auth, payments, user content or commerce, so the XSS blast radius is low and this is calibrated to P3.
- **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: 'unsafe-
inline';
  frame-ancestors 'self';
```

3B. Allow browser caching of the HTML document `Performance`

- **Impact:** Repeat-visit TTFB, LCP
- **Problem:** The document response is served with `cache-control: max-age=0` despite WP Rocket being active, forcing revalidation on every navigation.
- **Solution:**
  - In WP Rocket, enable browser caching for HTML with a short TTL (e.g. 10 minutes) plus `stale-while-revalidate`.
  - Keep `Vary: Accept-Encoding` (already present) so the gzip variant is cached correctly.

3C. Verify the undetermined color-contrast node `Accessibility`

- **Impact:** WCAG 1.4.3 contrast
- **Problem:** axe-core reports 0 violations but 1 incomplete node for color-contrast that it could not evaluate (likely alpha or gradient text).
- **Solution:**
  - Inspect the flagged element in DevTools and confirm the computed contrast is ≥4.5:1 for body text and ≥3:1 for large text.
  - If it fails, darken the text color or add a solid background behind the gradient.

3D. Reduce technology disclosure and move the login path `Best Practices`

- **Impact:** Attack-surface reduction
- **Problem:** The `server: Apache` header is disclosed and the WordPress login form is served on the default `/wp-login.php` path.
- **Solution:**
  - Suppress the banner: `ServerTokens Prod` and `ServerSignature Off` in Apache.
  - Restrict or rename the login endpoint (e.g. a WAF rule or a custom login slug) and add rate limiting plus 2FA.

3E. Suppress server technology disclosure and add X-Permitted-Cross-Domain-Policies `Security`

- **Impact:** Information disclosure
- **Problem:** Response headers disclose `Server: Apache`, and `X-Permitted-Cross-Domain-Policies` is missing.
- **Solution:**

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

3F. Disambiguate repeated link text `SEO`

- **Impact:** Anchor-text quality, screen-reader navigation
- **Problem:** 10 link texts repeat verbatim (e.g., "home" ×2, "contact" ×2, "development" ×2, "info@playandnope.com" ×2), which weakens anchor context.
- **Solution:** Add `aria-label` or visually-hidden context to each duplicate, e.g.:

```
<a href="/contact/" aria-label="Contact Play & NOPE
Alliance">contact</a>
```

3G. Trim unused CSS and legacy JavaScript `Performance`

- **Impact:** Render-blocking bytes, parse time
- **Problem:** PSI flags `unused-css-rules` (13 KiB savings) and `legacy-javascript-insight` (12 KiB savings); 5 inline stylesheets total 11.5 KB.
- **Solution:**
  - Purge unused selectors from the theme CSS (WP Rocket can do this).
  - Raise the browserslist target / drop legacy polyfills so modern browsers get the smaller bundle.

3H. Complete the remaining header hardening `Security`

- **Impact:** Transport security, origin isolation
- **Problem:** HSTS lacks the `preload` directive, COOP and CORP 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=63072000;
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 once the preload directive is live.

3I. Move the WordPress login off the default path `Security`

- **Impact:** Brute-force / credential-stuffing surface
- **Problem:** The security checklist warns that the login form is served on the default `/wp-login.php` path.
- **Solution:**
  - Rename the login URL (e.g. WPS Hide Login) and add rate limiting / 2FA.
  - Keep `xmlrpc.php` blocked (already passing) and retain the existing user-enumeration protection.

3J. Fix the typo in the contact page H2 `Best Practices`

- **Impact:** Content credibility
- **Problem:** The heading outline shows `h2: Let's start theconverstation` — a missing space and a misspelling of 'conversation'.
- **Solution:** Correct the heading text to "Let's start the conversation" in the page editor.

3K. Add HSTS preload directive `Security`

- **Impact:** HSTS coverage for first-visit and subdomain requests
- **Problem:** HSTS is set to `max-age=31536000; includeSubDomains` but lacks the `preload` directive, so the domain cannot be submitted to the browser preload list.
- **Solution:**
  - Append `preload` to the header:

```
Strict-Transport-Security: max-age=31536000; includeSubDomains;
```

```
preload
```

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

3L. Add Cross-Origin-Opener-Policy and Cross-Origin-Resource-Policy `Security`

- **Impact:** Cross-origin isolation, Spectre-class side-channel mitigation
- **Problem:** Both COOP and CORP are missing from the response headers; the security-basics advanced tier flags both as fail.
- **Solution:**

```
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-origin
```

  - Verify no legitimate cross-origin popups or embeds break before enforcing.

3M. Add Subresource Integrity to jsdelivr scripts `Security`

- **Impact:** Supply-chain integrity for third-party JS
- **Problem:** 2 of 2 cross-origin scripts (cookieconsent and iframemanager from cdn.jsdelivr.net) lack an `integrity` attribute; jsdelivr accounts for 6 requests / 28.6 KB.
- **Solution:**
  - Add `integrity` and `crossorigin="anonymous"` to each CDN script tag:

```
<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`.
  - Pin the exact version (already done: `@3.0.1`, `@1.3.0`) so hashes stay valid.

3N. Reduce technology disclosure `Best Practices`

- **Impact:** Attack-surface reconnaissance
- **Problem:** The `Server: Apache` response header discloses the web server, and the WordPress login form is served on the default `/wp-login.php` path (security-basics warn on both).
- **Solution:**

    - Suppress the server banner: `ServerTokens Prod` and `ServerSignature Off` in Apache config.
    - Move or rename the login path (e.g. a plugin that rewrites `/wp-login.php` to a custom slug) and add rate limiting / 2FA.
    - Keep the existing protections (xmlrpc blocked, no user enumeration, no readme.html) in place.

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

- **Impact:** Rich results eligibility, mobile usability
- **Problem:** PSI's SEO category fails `structuredData` and `tapTargets`; the single JSON-LD block is reported with type "Unknown", so no recognized schema type is emitted.
- **Solution:**
    - Replace the generic JSON-LD with a typed graph, e.g. `Organization` + `WebSite` + `BreadcrumbList`:

```
{"@context":"https://schema.org","@type":"Organization","name":"Pla
y and Nope","url":"https://playandnope.com/"}
```

    - Validate with Google's Rich Results Test.
    - Increase tap target sizes to at least 48×48 px with adequate spacing for the flagged mobile elements.

3P. Suppress server technology disclosure `Security`

- **Impact:** Information disclosure
- **Problem:** The `Server: Apache` response header discloses the web server software.
- **Solution:** In Apache config:

```
ServerTokens Prod
ServerSignature Off
```

3Q. Move WordPress login off the default path `Security`

- **Impact:** Brute-force surface
- **Problem:** The login form is served on the default `/wp-login.php` path.
- **Solution:** Rename the login URL (e.g. WPS Hide Login), add rate limiting / 2FA, and restrict `/wp-login.php` by IP where possible.

3R. Fix presentation-role-conflict on reCAPTCHA iframe `Accessibility`

- **Impact:** WCAG 4.1.2
- **Problem:** axe reports a minor `presentation-role-conflict` on `iframe[title="reCAPTCHA"]`.
- **Solution:** Ensure the reCAPTCHA iframe is not marked `role="presentation"` while retaining a `title`; if the conflict comes from a wrapper, remove the conflicting role.

3S. Fix heading typo `Best Practices`

- **Impact:** Content quality
- **Problem:** The H2 reads "Let's start theconverstation" — missing space and misspelled "conversation".
- **Solution:** Correct to "Let's start the conversation".

3T. Eliminate remaining render-blocking scripts `Performance`

- **Impact:** FCP, LCP
- **Problem:** 2 render-blocking scripts remain among the 42 total scripts.
- **Solution:** Add `defer` or `async` to the remaining render-blocking scripts, or move them to the end of `<body>`.