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: 82 / 100
Status: □ □ **Needs Improvement**

Site overall 82 is the mean of 5 pages. Scores range 68 (https://playandnope.com/digital-solutions-contact) → 89 (https://playandnope.com/privacy-policy). Weakest page: PSI mobile performance is 57 (desktop 75) with LCP 6.7 s and TBT 660 ms — both heavy penalties — and TTI 11.3 s; the mobile/desktop LCP delta (6.7 s vs 0.6 s) is far beyond the 1.5 s threshold, so mobile-first scoring dominates. The root cause is third-party reCAPTCHA: 1.42 MB from www.gstatic.com, a 346 KB recaptcha__en.js, 176/170/170 KB of unused JS, and long tasks of 263/253/218/214 ms, against a 1.90 MB total page weight. Accessibility is otherwise strong (PSI 99/99, axe 1 minor violation) but the contact form has an unlabeled textarea and axe could not resolve 10 color-contrast nodes. Security basics PASS, yet the CSP is frame-ancestors-only (no default-src, no object-src 'none'), HSTS lacks preload, COOP/CORP are missing, and 3 of 4 cross-origin scripts lack SRI. W3C validation returned HTTP 429, so HTML validity is unscored — that gap lowers confidence to medium. Overall 68: solid SEO (100) and clean image handling (5 SVGs, all with dimensions and lazy loading) offset by a mobile performance profile dragged down by a single third-party script.

Per-page scores

 Needs Improvement · https://playandnope.com/

Score	Performance	Accessibility	Best Practices	SEO	Security
84	90	100	100	92	PASS

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

Score	Performance	Accessibility	Best Practices	SEO	Security
89	96	100	100	100	PASS

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

Score	Performance	Accessibility	Best Practices	SEO	Security
80	83	99	96	100	PASS

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

Score	Performance	Accessibility	Best Practices	SEO	Security
88	97	98	100	100	PASS

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

Score	Performance	Accessibility	Best Practices	SEO	Security
68	57	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** / 97	**3.10 s** / 525 ms	0.000 / **0.102**

URL	Performance (M / D)	LCP (M / D)	CLS (M / D)
https://playandnope.com/privacy-policy	96 / 96	**2.41 s** / 570 ms	0.000 / **0.112**
https://playandnope.com/contact	83 / **78**	**3.18 s** / 1.17 s	0.056 / **0.238**
https://playandnope.com/full-stack-digital-agency	**97** / 98	**2.27 s** / 558 ms	0.010 / **0.084**
https://playandnope.com/digital-solutions-contact	**57** / 75	**6.69 s** / 588 ms	0.056 / **0.239**

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.

Item	Status	Detail
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. Fix color contrast on navigation description text `Accessibility`

- **Impact:** WCAG 1.4.3 (AA), axe serious violation
- **Problem:** axe reports 1 serious `color-contrast` violation on `.menu-item-75 > .navigation__description`, and 8 further nodes could not be auto-determined (likely alpha/gradient backgrounds).
- **Solution:**
  - Raise the text/background contrast ratio to at least 4.5:1 (3:1 for ≥18.66 px bold or ≥24 px text).
  - Inspect the 8 undetermined nodes manually — axe skips elements with gradients or alpha transparency.

```
.navigation__description { color: #4a4a4a; /* verify ≥4.5:1 on the actual background */ }
```

1B. Cut reCAPTCHA JavaScript bloat and long tasks `Performance`

- **Impact:** TBT, TTI, LCP, page weight
- **Problem:** reCAPTCHA pulls 1.42 MB from gstatic.com with ~516 KB of unused JS across three recaptcha__en.js entries and 10 long tasks up to 175 ms; mobile TBT is 332 ms and TTI is 11.2 s.
- **Solution:** reCAPTCHA is the single largest performance cost on this page. Options, in order of effort:
  - **Defer loading until interaction**: only inject `https://www.google.com/recaptcha/api.js` when the user focuses the first form field, not on page load.
  - **Use reCAPTCHA v3 with** `render=explicit` and call `grecaptcha.execute()` on submit, so the 170 KB+ payload is not parsed during initial load.

- **Consider a lighter alternative** (hCaptcha, Cloudflare Turnstile, or a honeypot + server-side rate limit) if the form volume is low.
- Verify Gravity Forms is not enqueueing reCAPTCHA twice — the same `recaptcha__en.js` URL appears three times in the unused-JS list.

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

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is `frame-ancestors 'self';` only — no `default-src` and no `object-src 'none'`, so scripts, styles and plugins are unrestricted. The page contains a `<textarea>` (user-content signal), which raises the value of a real CSP.
- **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';
  form-action 'self';
```

  Generate a fresh nonce per request and add it to every `<script>` tag. Keep `frame-ancestors 'self'` (already present) and add `object-src 'none'` immediately — that is a one-line change.

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

- **Impact:** WCAG 1.4.3 (AA), axe serious violation
- **Problem:** axe flags a serious `color-contrast` violation on `.menu-item-74 > .navigation__description` and `.menu-item-75 > .navigation__description`; 9 further nodes were left 'incomplete' because axe could not compute contrast.
- **Solution:**
  - Darken the description text or lighten its background until the ratio is ≥4.5:1 for normal text (≥3:1 for ≥18.66 px bold / ≥24 px).
  - Re-run axe after the change and manually verify the 9 'incomplete' nodes (alpha/gradient backgrounds are skipped by the automated check).

```
.navigation__description { color: #4a4a4a; /* verify against actual
bg */ }
```

1E. Stop loading reCAPTCHA on page load — defer it until the form is used `Performance`

- **Impact:** TBT, LCP, TTI, page weight
- **Problem:** recaptcha__en.js is 346 KB and www.gstatic.com accounts for 1.42 MB of the 1.90 MB page; Lighthouse attributes 176/170/170 KB of unused JS and long tasks of 263/253/218/214 ms to it, driving mobile TBT to 660 ms and TTI to 11.3 s.
- **Solution:** Load reCAPTCHA only when the user actually engages the form:
  - Keep `render=explicit` and call `grecaptcha.render()` on first focus/click of the form, not on DOMContentLoaded.
  - Inject the script tag dynamically instead of shipping it in the initial HTML.

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

  - If the form is low-risk, evaluate a lighter alternative (Cloudflare Turnstile) that ships far less JS.

1F. Fix mobile LCP (6.7 s) — preload the LCP asset and remove render-blocking scripts `Performance`

- **Impact:** LCP, FCP, Speed Index
- **Problem:** Mobile LCP is 6.7 s vs 0.6 s on desktop, FCP 1.99 s, Speed Index 3.66 s, and the HTML inventory still shows 2 render-blocking scripts in the head out of 42 total.
- **Solution:**
  - Identify the LCP element in DevTools (Performance panel → LCP marker) and preload it:

```
<link rel="preload" as="image" href="/path/to/lcp.webp"
fetchpriority="high">
```

- Inline critical above-the-fold CSS and defer the rest.
- Add `defer` (or `async`) to the 2 remaining render-blocking scripts.
- Preload the primary Roboto woff2 (39.2 KB, 148 ms) with `rel="preload" as="font" crossorigin`.

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Reduce mobile LCP (3.1 s) and Speed Index (4.21 s) `Performance`

- **Impact:** LCP, Speed Index, FCP, TBT
- **Problem:** Mobile LCP is 3.1 s vs 0.5 s on desktop, Speed Index 4.21 s vs 0.668 s, and PSI flags `forced-reflow-insight` **(high)** plus `legacy-javascript-insight` (high, ~12 KiB savings); jQuery produces a 50 ms long task.
- **Solution:**
  - Eliminate forced synchronous layout: batch DOM reads before writes, avoid `offsetHeight` / `getBoundingClientRect()` inside loops.
  - Drop legacy JS transpilation targets (browserslist) to remove the ~12 KiB of polyfill overhead.
  - Defer or replace jQuery-dependent code where possible; the 50 ms long task originates from `jquery.f5f8c1b60d5ba64e00fb.min.js`.
  - Preload the actual LCP element and add `<link rel="preconnect">` for `cdn.jsdelivr.net` (currently 0 preconnects on 49 anchors).

2B. Add explicit width/height to the 5 dimension-less images `Performance`

- **Impact:** CLS (desktop 0.102)
- **Problem:** HTML inventory shows 5 of 30 images without explicit width/height, and desktop CLS is 0.102 — just over the 0.1 good threshold.
- **Solution:** Add intrinsic dimensions (or an aspect-ratio) so the browser can reserve space:

```
<img src="…" width="440" height="138" alt="…" loading="lazy"
decoding="async">
```

```
img { aspect-ratio: attr(width) / attr(height); }
```

2C. Fix tap targets, structured data and non-descriptive link text `SEO`

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

- **Problem:** PSI SEO is 92 with failing audits `tapTargets`, `structuredData` (JSON-LD block present but type reported as Unknown) and `link-text` (4 links without descriptive text); the HTML inventory also shows "read more" repeated ×4.
- **Solution:**
  - Enlarge tap targets to ≥48×48 px with adequate spacing (WCAG 2.5.8).
  - Replace the generic JSON-LD `@type` with a real schema (`Organization`, `WebSite`, `Service`) and validate it.
  - Rewrite vague anchors, e.g. `read more` → `Read more about Full-stack digital solutions`; add `aria-label` where the visible text must stay short.

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

- **Impact:** Supply-chain integrity (jsdelivr CDN)
- **Problem:** Security basics reports 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:

```
<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`. Better still, self-host these two libraries.

2E. Add the HSTS preload directive `Security`

- **Impact:** Transport security hardening
- **Problem:** Security basics advanced check `HSTS preload` fails: `strict-transport-security: max-age=31536000; includeSubDomains` is missing `preload`.
- **Solution:**

```
Strict-Transport-Security: max-age=63072000; includeSubDomains;
preload
```

  Then submit the domain at hstspreload.org — note the commitment is effectively permanent, so confirm all subdomains are HTTPS-only first.

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

- **Impact:** Document latency, FCP, LCP

- **Problem:** The runtime capture shows two document requests — `/privacy-policy` (482 ms, 0 B) redirecting to `/privacy-policy/` (641 ms, 21.1 KB) — matching PSI's `document-latency-insight` (high, ~540 ms estimated savings).
- **Solution:**
  - Point all internal links and the canonical/OG URLs at the trailing-slash form so the redirect never fires.
  - If the redirect is server-side, resolve it at the edge (CDN rule) rather than in WordPress.
  - Verify with `curl -sI https://playandnope.com/privacy-policy` — expect a single 200, no 301.

2G. Cache the HTML document `Performance`

- **Impact:** TTFB, repeat-visit LCP, document latency
- **Problem:** The document response sets `cache-control: max-age=0` with an `expires` timestamp already in the past, so browsers revalidate the HTML on every navigation even though WP Rocket is active.
- **Solution:** Serve a short browser TTL with background revalidation for static pages like this one:

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

  In WP Rocket, confirm "Cache Lifespan" is set and that the privacy-policy URL is not excluded from page caching.

2H. Fix desktop CLS 0.112 `Performance`

- **Impact:** CLS
- **Problem:** Desktop CLS is 0.112 (mobile 0.000), inside the 0.1–0.25 warning band, while the page loads 3 web fonts (105.3 KB) and a deferred cookie-consent banner from cdn.jsdelivr.net.
- **Solution:**
  - Add `font-display: optional` (or `swap` plus a matched fallback with `size-adjust`) to the three font faces.
  - Reserve space for the cookie-consent banner before its script runs (fixed height container or `min-height` on the wrapper).
  - Re-measure with Lighthouse desktop after each change to confirm CLS drops below 0.1.

2I. Improve mobile FCP and the critical request chain `Performance`

- **Impact:** FCP, LCP, Speed Index

- **Problem:** Mobile FCP is 1.96 s (desktop 528 ms) and PSI flags `network-dependency-tree-insight` as high priority; the HTML inventory reports 0 preconnect hints despite 6 requests to cdn.jsdelivr.net.
- **Solution:**
  - Add preconnect/dns-prefetch for the CDN origin:

```
<link rel="preconnect" href="https://cdn.jsdelivr.net" crossorigin>
```

  - Inline the critical CSS for the above-the-fold region and defer the rest.
  - Self-host the cookie-consent CSS/JS so it is not a third-party round trip on the critical path.

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

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is only `frame-ancestors 'self';` — the security basics table marks "CSP is strict" as fail for missing `default-src` and `object-src` not set to `'none'`.
- **Solution:** Add a nonce-based policy 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';
```

  Roll it out in report-only mode first (`Content-Security-Policy-Report-Only`) to catch WordPress plugin violations before enforcing.

2K. Fix desktop CLS (0.238) `Performance`

- **Impact:** CLS
- **Problem:** Desktop CLS is 0.238 (warning band) versus 0.056 on mobile; Lighthouse attributes the shift to `body.wp-singular > div#page > main#main-content > div.section`.
- **Solution:**
  - Reserve explicit height/width on the shifting `.section` container so late-arriving content does not push the layout.
  - If the shift comes from a cookie-consent banner or reCAPTCHA badge, render a placeholder of the same size in the initial HTML.
  - Avoid injecting content above existing content after load; use `transform` animations instead of layout-affecting properties.
  - Re-run PSI desktop after the change to confirm CLS drops below 0.1.

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

- **Impact:** Best Practices, console errors
- **Problem:** 5 console events were captured, including two `requestStorageAccess: Permission denied` errors, a CORS block on `styles__ltr.css` from gstatic.com, and a `net::ERR_FAILED` for the same resource.
- **Solution:**
  - The CORS failure on `recaptcha__en.js`'s companion stylesheet is inside Google's own iframe — verify it is not caused by a local `Cross-Origin-Resource-Policy` or `Cross-Origin-Embedder-Policy` header you set.
  - The `requestStorageAccess` denials are typically third-party-cookie blocking; they are benign for reCAPTCHA but pollute the console. Confirm the form still submits successfully in a private window.
  - Re-run the browser capture after any CSP change to confirm the error count drops.

2M. Enable document caching `Performance`

- **Impact:** TTFB, repeat visits
- **Problem:** The document response sends `cache-control: max-age=0` with `expires` equal to `date`, so every visit revalidates the HTML even though WP Rocket is detected.
- **Solution:**
  - In WP Rocket, confirm "Cache lifespan" is set to a positive value and that the contact page is not excluded from the cache.
  - Verify the origin does not override WP Rocket's headers with `max-age=0`.
  - Confirm the cache is warmed for the canonical URL after deploy so the first visitor does not pay the cold-start cost.

2N. Add HSTS preload directive `Security`

- **Impact:** HSTS coverage
- **Problem:** HSTS is `max-age=31536000; includeSubDomains` but lacks the `preload` directive, so the domain cannot be submitted to the browser preload list.
- **Solution:** Append `preload` and submit the domain at https://hstspreload.org:

```
Strict-Transport-Security: max-age=63072000; includeSubDomains;
preload
```

Note: preload is a long-term commitment — confirm every subdomain serves HTTPS before submitting.

2O. Label the unlabeled textarea in the contact form `Accessibility`

- **Impact:** WCAG 3.3.2, 4.1.2
- **Problem:** Form 1 contains a `<textarea>` with no associated label (the 12 hidden inputs without labels are exempt from labelling requirements).
- **Solution:** Associate a visible `<label>` with the textarea:

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

  If the field is intentionally hidden, add `aria-hidden="true"` and `tabindex="-1"` instead.

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

- **Impact:** WCAG 1.3.1, PSI heading-order audit (weight 3)
- **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 fails at weight 3.
- **Solution:**
  - Change the hero subtitle from `<h3>` to `<h2>` so the outline descends one level at a time.
  - Audit the rest of the outline (h1 ×1, h2 ×7, h3 ×12) for other implicit skips introduced by visual styling rather than semantics.

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

- **Impact:** Repeat-visit TTFB, LCP, FCP
- **Problem:** The document response is served with `cache-control: max-age=0` and `expires` equal to the request time, so returning visitors always re-download the 210 KB decoded HTML even though WP Rocket is active.
- **Solution:**
  - In WP Rocket → Cache → Browser cache, enable HTML caching with a short TTL (e.g. 10 minutes) and let the CDN handle longer TTLs.
  - Verify with `curl -I` that `cache-control` returns a positive `max-age` and that `vary: Accept-Encoding` is preserved.

2R. Reduce image delivery cost (463 KiB) `Performance`

- **Impact:** LCP, page weight, Speed Index

- **Problem:** PSI's `image-delivery-insight` estimates 463 KiB of savings; the runtime capture shows 8 image requests totalling 287.5 KB out of 560.6 KB total page weight.
- **Solution:**
  - Serve correctly sized variants via `srcset` / `sizes` and modern formats (WebP/AVIF) where raster images are used.
  - The page currently ships 44 SVG images; check that any large SVG hero/illustration assets are minified (SVGO) and gzip/brotli-compressed.
  - Confirm the actual LCP element in DevTools Performance panel — the checklist heuristic could not identify a raster hero.

2S. Eliminate forced reflow and shorten the critical request chain `Performance`

- **Impact:** FCP (1.97 s mobile), Speed Index (2.64 s), document latency
- **Problem:** PSI flags `forced-reflow-insight` **(high)** and `document-latency-insight` (high, est. 660 ms savings); mobile FCP is 1.97 s versus 538 ms on desktop.
- **Solution:**
  - Batch DOM reads/writes in JS to avoid layout thrashing (read all geometry first, then write).
  - Reduce the critical request chain: preload the LCP resource and inline critical CSS.
  - Defer non-critical theme scripts further; 24 scripts are already defer/async, so focus on execution order rather than blocking.

2T. Complete 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. Site signals show no auth, payments, or user content, so this is calibrated to P2 rather than P1.
- **Solution:**
  - Add `default-src 'self'` and `object-src 'none'` at minimum, then tighten `script-src` with nonces:

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

    - Note the page loads scripts from `cdn.jsdelivr.net`; allowlist or self-host them before enabling `default-src 'self'`.

2U. Label the unlabeled textarea and verify contrast on 10 unresolved nodes `Accessibility`

- **Impact:** WCAG 4.1.2, 1.4.3, 1.4.1
- **Problem:** Form 1 contains a `textarea` with no label, and axe returned 10 color-contrast nodes plus 1 link-in-text-block node as "incomplete" (could not determine).
- **Solution:**
    - Add a visible `<label for="…">` (or `aria-label`) to the textarea; the other three text inputs are already labeled.
    - Run a contrast checker on the 10 flagged nodes (likely gradient/alpha text) and raise any below 4.5:1 to at least 4.5:1.
    - Underline in-body links or add a non-color affordance so they are distinguishable without color.

2V. Harden the Content-Security-Policy (default-src, object-src 'none') `Security`

- **Impact:** XSS defense-in-depth
- **Problem:** The CSP is `frame-ancestors 'self';` only — no `default-src`, no `object-src 'none'`, and no script/resource restrictions, so it provides no XSS mitigation. The `hasUserContent` signal is triggered by a contact-form `<textarea>`, not persistent stored content, so this is calibrated to P2 rather than P1.
- **Solution:** Deploy a nonce-based policy 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';
```

    - Emit a fresh nonce per response and add it to every `<script>` tag.
    - If the contact form submissions are stored and later rendered in wp-admin, treat this as P1.

2W. Fix console errors from reCAPTCHA (CORS, storage access, ERR_FAILED) `Best Practices`

- **Impact:** Best Practices errorsInConsole, form reliability
- **Problem:** Browser runtime captured 5 console events: two `requestStorageAccess: Permission denied`, a CORS-blocked XHR to gstatic `styles__ltr.css`, and `net::ERR_FAILED`; PSI's `errors-in-console` audit fails.

- **Solution:**
  - The CORS/ERR_FAILED errors originate inside Google's reCAPTCHA iframe and are usually benign, but they indicate the widget is loading in a degraded state.
  - Verify the reCAPTCHA site key is registered for `playandnope.com` (including any www/non-www variants) in the reCAPTCHA admin console.
  - If the errors persist, switch to a lighter challenge (Turnstile) or a server-side honeypot + rate limit.

2X. Set a real browser cache lifetime on the HTML document `Performance`

- **Impact:** Repeat-visit TTFB, LCP
- **Problem:** The document response carries `cache-control: max-age=0` with `expires` equal to the request date, so the HTML is revalidated on every navigation despite WP Rocket being active.
- **Solution:**
  - In WP Rocket → Cache → Browser cache, enable browser caching for HTML.
  - Or set at the origin:

```
Header set Cache-Control "public, max-age=600, stale-while-
revalidate=86400"
```

  - Keep `Vary: Accept-Encoding` (already present).

2Y. Preconnect to third-party origins `Performance`

- **Impact:** LCP, FCP, connection setup
- **Problem:** The page loads from 4 third-party origins (www.gstatic.com 1.42 MB, fonts.gstatic.com 78.4 KB, cdn.jsdelivr.net 28.6 KB, www.google.com) but the HTML inventory reports 0 preconnect links.
- **Solution:** Add early hints in `<head>`:

```
<link rel="preconnect" href="https://www.gstatic.com" crossorigin>
<link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
<link rel="preconnect" href="https://cdn.jsdelivr.net" crossorigin>
```

  - Better still, self-host the two jsdelivr libraries and the Roboto font to remove two origins entirely.

Priority 3: Best Practice

Recommended for long-term maintainability.

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

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is present but weak: `frame-ancestors 'self';` only — no `default-src`, no `object-src 'none'`, and no script/resource restrictions. Site signals show no auth, payments, user content or commerce, so the XSS blast radius is low and this is calibrated to P3.
- **Solution:** Move to a nonce-based policy 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';
```

  Note the two jsdelivr scripts will need nonces or self-hosting.

3B. Set COOP and CORP, and stop disclosing the server banner `Security`

- **Impact:** Origin isolation, information disclosure
- **Problem:** `Cross-Origin-Opener-Policy` and `Cross-Origin-Resource-Policy` are missing, and the response discloses `Server: Apache`; the WordPress login form is also served on the default `/wp-login.php` path.
- **Solution:**

```
Header always set Cross-Origin-Opener-Policy "same-origin"
Header always set Cross-Origin-Resource-Policy "same-origin"
Header always unset Server
ServerSignature Off
```

  Also move the login form off `/wp-login.php` (custom slug or a login plugin) to cut automated credential-stuffing noise.

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

- **Impact:** Repeat-visit TTFB, LCP
- **Problem:** The document response sends `cache-control: max-age=0` with `expires` equal to the request date, forcing revalidation on every navigation despite WP Rocket being active.
- **Solution:** Serve a short but non-zero TTL with revalidation support:

```
Cache-Control: public, max-age=60, stale-while-revalidate=300
```

Keep the WP Rocket page cache for the origin and let the CDN/browser absorb repeat views.

3D. Re-run W3C HTML validation `Best Practices`

- **Impact:** HTML validity coverage
- **Problem:** The W3C Validator returned HTTP 429 (rate limited), so HTML validity is unverified for this run — 88% audit coverage.
- **Solution:** Re-run the validator after the rate limit resets, or validate locally:

```
npx html-validate https://playandnope.com/
```

Given 152 KB of HTML and 22 scripts, expect a handful of nesting/attribute warnings worth triaging.

3E. Add HSTS preload `Security`

- **Impact:** Transport security
- **Problem:** HSTS is `max-age=31536000; includeSubDomains` but lacks the `preload` directive, which the security basics table marks as fail.
- **Solution:**
  - Append `preload` to the header:

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

  - Confirm every subdomain is HTTPS-only, then submit the domain at hstspreload.org.

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

- **Impact:** Origin isolation, cross-origin data leaks
- **Problem:** Both COOP and CORP are missing (advanced-tier fails in the security basics table).
- **Solution:**

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

Use `same-site` for CORP if any subdomain legitimately embeds these assets.

3G. Disambiguate repeated link text and verify unresolved contrast `Accessibility`

- **Impact:** WCAG 2.4.4, 1.4.3
- **Problem:** The HTML inventory lists 34 anchors with repeated identical text ("home" ×2, "work" ×2, "contact" ×2, "info@playandnope.com" ×2, etc.), and axe returned 9 incomplete color-contrast nodes plus 1 link-in-text-block node it could not resolve.
- **Solution:**
  - Give each duplicate link a distinct accessible name:

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

  - Manually check the 9 unresolved contrast nodes (likely gradient/alpha text) against 4.5:1 for body copy and 3:1 for large text.
  - Ensure in-text links are distinguishable by more than color (underline or weight).

3H. Fix structured data `SEO`

- **Impact:** Rich results eligibility
- **Problem:** PSI flags `structuredData` as a failing SEO audit and the single JSON-LD block is typed "Unknown", so no valid schema.org entity is being emitted.
- **Solution:**
  - Replace the block with a valid type, e.g. `WebPage` / `Organization`:

```
{"@context":"https://schema.org","@type":"WebPage","name":"Privacy Policy","url":"https://playandnope.com/privacy-policy/"}
```

  - Validate with the Rich Results Test and the Schema Markup Validator.

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

- **Impact:** JS/CSS bytes, FCP
- **Problem:** PSI estimates 12 KiB savings from legacy JavaScript and 13 KiB from unused CSS rules; the runtime capture shows 103.6 KB of script and 23.0 KB of stylesheet bytes.
- **Solution:**
  - Add a modern `browserslist` target so the build stops shipping transpiled polyfills.
  - Purge unused CSS rules at build time (or use WP Rocket's Remove Unused CSS).
  - Re-check the 22 scripts for duplicates — jQuery plus jquery-migrate are both loaded.

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

- **Impact:** Brute-force surface
- **Problem:** The security basics table warns that the login form is served on the default `/wp-login.php` path, which is the first target for credential-stuffing bots.
- **Solution:**
  - Rename the login URL (e.g. WPS Hide Login) and block direct `/wp-login.php` access at the edge.
  - Enforce 2FA for all accounts and rate-limit `/wp-login.php` and `wp-json/wp/v2/users`.
  - Keep the existing xmlrpc.php block in place.

3K. Suppress the Server header `Security`

- **Impact:** Technology disclosure
- **Problem:** The response discloses `Server: Apache`, which aids targeted fingerprinting.
- **Solution:**

  ```
  ServerTokens Prod
  ServerSignature Off
  ```

  Or strip the header at the reverse proxy / CDN edge.

3L. Resolve the presentation-role-conflict on the reCAPTCHA iframe `Accessibility`

- **Impact:** WCAG 4.1.2
- **Problem:** axe reports one minor `presentation-role-conflict` violation on `iframe[title="reCAPTCHA"]` — the element is marked presentational but is focusable.
- **Solution:**
  - If the iframe is decorative, add `tabindex="-1"` and `aria-hidden="true"` alongside `role="presentation"`.
  - If it is interactive (reCAPTCHA is), remove `role="presentation"` instead.
  - This is inside Google's injected markup — check whether a local wrapper is adding the conflicting role.

3M. Verify color contrast on the 10 incomplete nodes `Accessibility`

- **Impact:** WCAG 1.4.3

- **Problem:** axe could not determine contrast for 10 nodes (likely alpha-transparency or gradients) and flagged 1 `link-in-text-block` node as incomplete.
- **Solution:**
  - Manually check the 10 nodes with a contrast checker against the rendered background.
  - Ensure in-text links are distinguishable by more than color alone (underline, weight, or an icon).
  - Re-run axe after any theme color change to confirm.

3N. Disambiguate repeated link text `SEO`

- **Impact:** Link purpose, crawl clarity
- **Problem:** 43 anchors include repeated vague text such as "home" ×2, "work" ×2, "about" ×2, "contact" ×2, and "info@playandnope.com" ×3.
- **Solution:**
  - Add `aria-label` to icon-only or repeated links, e.g. `<a href="/work/" aria-label="View our work">work</a>`.
  - For the repeated email link, vary the visible text or add context via `aria-describedby`.

3O. Fix the failing structuredData audit `SEO`

- **Impact:** Rich results eligibility
- **Problem:** PSI reports `structuredData` as a failing SEO audit, and the single JSON-LD block on the page has an unrecognised type.
- **Solution:**
  - Validate the JSON-LD block with Google's Rich Results Test.
  - Use a recognised schema type (`Organization`, `LocalBusiness`, `ContactPage`, or `WebSite`) with the required properties.
  - Add `@id` and `url` fields so the entity can be linked across pages.

3P. Add HSTS preload directive `Security`

- **Impact:** HSTS preload list eligibility
- **Problem:** `strict-transport-security: max-age=31536000; includeSubDomains` is set but lacks `preload`, so the domain cannot be submitted to the HSTS preload list.
- **Solution:**
  - Append `preload` and submit at https://hstspreload.org once you are confident every subdomain is HTTPS-only:

```
Strict-Transport-Security: max-age=63072000; includeSubDomains;
preload
```

3Q. Add COOP and CORP headers `Security`

- **Impact:** Cross-origin isolation, side-channel mitigation
- **Problem:** `cross-origin-opener-policy` and `cross-origin-resource-policy` are both missing; the tool rates these low severity and the site has no auth or payment surface.
- **Solution:**

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

  - Test after deploy: COOP can break OAuth popups and cross-origin window messaging.

3R. Add Subresource Integrity to cross-origin scripts `Security`

- **Impact:** Supply-chain tampering risk
- **Problem:** 2 of 2 cross-origin scripts (from `cdn.jsdelivr.net`) lack an `integrity` attribute, so a compromised CDN could inject arbitrary JS.
- **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`.
  - Alternatively self-host the two jsdelivr libraries.

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

- **Impact:** Link purpose, rich-result eligibility
- **Problem:** The inventory lists repeated vague anchors ("home" ×2, "work" ×2, "about" ×2, "contact" ×2, "design & innovation" ×2) and PSI's `structuredData` audit fails; the single JSON-LD block is typed "Unknown".
- **Solution:**

  - Add `aria-label` or visually hidden context to repeated anchors so each link's purpose is unique out of context.

- Replace the unrecognised JSON-LD with a valid `Organization` / `LocalBusiness` schema and validate it in the Rich Results Test.
- Also review the failing `tapTargets` audit — ensure interactive elements meet the 44×44 px minimum.

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

- **Impact:** Mobile usability, rich results
- **Problem:** PSI's modeled audits `tapTargets` and `structuredData` both fail, and the single JSON-LD block has type "Unknown".
- **Solution:**
  - Ensure every interactive element is at least 44×44 px with adequate spacing (WCAG 2.5.8).
  - Replace the unrecognized JSON-LD with a valid `Organization` / `ContactPage` graph:

```
{"@context":"https://schema.org","@type":"ContactPage","name":"Cont
act","url":"https://playandnope.com/digital-solutions-contact/"}
```

3U. Add HSTS preload, COOP and CORP headers `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 at hstspreload.org once the preload directive is live.

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

- **Impact:** Attack surface, fingerprinting
- **Problem:** The response discloses `Server: Apache`, and the login form is served on the default `/wp-login.php` path (both flagged as warnings).
- **Solution:**
  - `ServerTokens Prod` and `ServerSignature Off` in Apache.
  - Move the login to a custom path (e.g. via a security plugin) and add rate limiting / 2FA.

- Optionally set `X-Permitted-Cross-Domain-Policies: none`.

3W. Disambiguate repeated link text and fix the heading typo `SEO`

- **Impact:** Link purpose, content quality
- **Problem:** The page repeats identical anchor text ('home' ×2, 'work' ×2, 'contact' ×2, 'about' ×2, 'development' ×2, etc.) and the h2 reads "Let's start theconverstation".
- **Solution:**
  - Add `aria-label` or visually hidden context to repeated nav links, e.g. `<a href="/work" aria-label="View our work">work</a>`.
  - Correct the h2 to "Let's start the conversation".