Audit Report: Perfectly formed web development team - gotoAndPlay

Website: https://play.ee/
Date: 02.11.2026
Audit Coverage: 100% — all sources returned data
Confidence: high

Pages Audited (5 of 5):

- https://play.ee/
- https://play.ee/web-development-case-studies/
- https://play.ee/software-development-work-index/
- https://play.ee/wordpress-support-service/
- https://play.ee/web-development-in-estonia/

Summary of results

Overall Score: 87 / 100
Status: **Needs Improvement**

Site overall 87 is the mean of 5 pages. Scores range 86 (https://play.ee/web-development-case-studies/) → 88 (https://play.ee/wordpress-support-service/). Weakest page: Performance is excellent — PSI mobile 96 / desktop 100, LCP 1.8 s, CLS 0.000, TBT 0 ms, TTFB 4 ms — but Speed Index diverges sharply (4.14 s mobile vs 519 ms desktop), driven by render-blocking CSS (est. 1,050 ms savings) and 31 KiB unused CSS. Accessibility is 95–96 with only 2 axe violations, though the serious color-contrast failure spans 23+ nodes across the h1 and every card title/meta, which is the single largest quality drag. Security basics PASS with HSTS, nosniff, frame-ancestors, Referrer-Policy and Permissions-Policy all set; the weak CSP (frame-ancestors only, no default-src/object-src) and missing COOP/CORP are advanced-tier gaps, and with no auth/payments/UGC/commerce signals they stay low priority. W3C reports 4 errors (rocketlazyloadscript data blocks carrying defer) and the image inventory is clean (56 images, 0 missing alt, all SVG so lazy-loading is n/a). No CrUX field data exists for this URL, so scoring rests on lab metrics only.

Per-page scores

 Needs Improvement · https://play.ee/

Score	Performance	Accessibility	Best Practices	SEO	Security
88	98	96	100	92	PASS

 Needs Improvement · https://play.ee/web-development-case-studies/

Score	Performance	Accessibility	Best Practices	SEO	Security
86	96	96	100	100	PASS

 Needs Improvement · https://play.ee/software-development-work-index/

Score	Performance	Accessibility	Best Practices	SEO	Security
87	97	96	100	100	PASS

 Needs Improvement · https://play.ee/wordpress-support-service/

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

 Needs Improvement · https://play.ee/web-development-in-estonia/

Score	Performance	Accessibility	Best Practices	SEO	Security
86	96	96	100	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://play.ee/	**98** / 100	**1.81 s** / 485 ms	0.001 / **0.006**
https://play.ee/web-development-case-studies/	**96** / 100	**1.81 s** / 519 ms	0.000 / **0.005**
https://play.ee/software-development-work-index/	**97** / 100	**1.80 s** / 497 ms	0.000 / **0.003**

URL	Performance (M / D)	LCP (M / D)	CLS (M / D)
https://play.ee/wordpress-support-service/	**96** / 100	**1.80 s** / 485 ms	0.000 / **0.001**
https://play.ee/web-development-in-estonia/	**96** / 100	**1.83 s** / 486 ms	0.000 / **0.004**

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 br.
Images lazy-loaded	N/A	No raster elements found (50 SVGs excluded).
Hero image eagerly loaded	N/A	No raster elements found (50 SVGs 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 (50 SVGs 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. Fix color contrast on headings and card meta text `Accessibility`

- **Impact:** WCAG 1.4.3 (AA), readability across 23+ nodes
- **Problem:** axe reports a serious color-contrast violation on `h1 > .heading__main` plus 22 more nodes including every `.card__title.h4` and `.card__meta`; PSI's `color-contrast` audit also scores 0.00 with weight 7.
- **Solution:** Darken the affected text colors to reach ≥4.5:1 against their backgrounds:

```
.heading__main,
.card__title.h4 { color: #1a1a1a; }        /* was too light on
white */
.card__meta    { color: #595959; }        /* ≥4.5:1 on #fff */
```

  Then re-run axe; 8 additional nodes were flagged "incomplete" (alpha/gradient backgrounds) and need manual verification.

1B. Fix color contrast on the H1 and lead paragraph `Accessibility`

- **Impact:** WCAG 1.4.3 (AA), axe serious violation, readability of the primary heading
- **Problem:** axe reports 1 serious `color-contrast` violation on `h1 > .heading__main` and `p:nth-child(1) > span`, and could not determine contrast for 9 further nodes (gradients/alpha).
- **Solution:**
  - Raise the foreground/background ratio to at least 4.5:1 for body text and 3:1 for large text (≥24 px, or ≥18.66 px bold).
  - Darken the heading color, e.g. `#8a8a8a` on white → `#595959` or darker.

```
.heading__main { color: #1a1a1a; }
.lead-text span { color: #4a4a4a; }
```

  - Manually verify the 9 undetermined nodes (alpha/gradient backgrounds) with a contrast checker.

1C. Fix color contrast on buttons and capability headings `Accessibility`

- **Impact:** WCAG 1.4.3 (AA), axe serious violation
- **Problem:** axe reports 1 serious `color-contrast` violation across 15 nodes, including `.button--tertiary > .button__inner > .button__text` and all four `.capabilities__heading > .heading__small` elements; 14 more nodes were flagged as incomplete.
- **Solution:**
  - Darken the tertiary button text/background pair to reach ≥4.5:1 (or ≥3:1 for large text).

- The `.heading__small` labels inside `.capabilities__heading` are the likely culprit — bump them from a light grey to something like `#595959` on white.
- Re-run axe after the change and manually verify the 14 "incomplete" nodes (alpha/gradient backgrounds are skipped by axe).

1D. Fix color contrast on headings and capability labels `Accessibility`

- **Impact:** WCAG 1.4.3 (AA), axe serious violation, 26 failing nodes
- **Problem:** axe reports 1 serious `color-contrast` violation covering 26 nodes — `.focus__heading > h1 > .heading__main` and all four `.capabilities__heading .heading__small` labels — with 86 more nodes axe could not determine (likely gradient/alpha backgrounds).
- **Solution:**
  - Raise foreground/background contrast to ≥4.5:1 for body text and ≥3:1 for text ≥24 px (or ≥18.66 px bold).
  - The hero `h1 .heading__main` and the `.capabilities__heading .heading__small` labels are the confirmed offenders — darken the text or lighten the overlay behind it.
  - Manually verify the 86 undetermined nodes (axe skips gradient/alpha backgrounds) with DevTools' contrast picker or the Accessibility pane.

```
/* example direction — verify against actual palette */
.focus__heading .heading__main { color: #1a1a1a; }
.capabilities__heading .heading__small { color: #4a4a4a; }
```

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Resolve color-contrast failures across 37 undetermined nodes `Accessibility`

- **Impact:** WCAG 1.4.3, PSI Accessibility score
- **Problem:** PSI fails `color-contrast` (score 0.00, weight 7) while axe returned 37 nodes it could not determine, meaning a large share of text may fall below 4.5:1.
- **Solution:**
  - Audit text/background pairs on the hero, footer, and card overlays with DevTools contrast picker.
  - Target ≥4.5:1 for body text and ≥3:1 for large text (≥24 px or ≥18.66 px bold).
  - Pay special attention to text over images/gradients, which axe skips:

```
.hero__overlay { background: rgba(0,0,0,.55); } /* raise opacity
until 4.5:1 */
```

2B. Stop serving the HTML document with cache-control: max-age=0 `Performance`

- **Impact:** TTFB on repeat visits, LCP
- **Problem:** The document response carries `cache-control: max-age=0` with `expires` equal to the request time, so every navigation revalidates the full 219 KB HTML despite WP Rocket being detected.
- **Solution:**
  - In WP Rocket, enable page caching for the front page and confirm the cache is warmed.
  - Ensure the origin emits a positive max-age for anonymous visitors, e.g.:

```
Cache-Control: public, max-age=600, s-maxage=3600, stale-while-
revalidate=86400
```

  - Verify with `curl -sI https://play.ee/ | grep -i cache-control` after deploy.

2C. Eliminate render-blocking CSS and unused CSS `Performance`

- **Impact:** FCP, LCP, Speed Index
- **Problem:** PSI reports `render-blocking-insight` with an estimated 1,030 ms of savings and `unused-css-rules` wasting 28 KiB; FCP is 1.81 s on mobile.
- **Solution:**
  - Inline critical above-the-fold CSS and load the rest asynchronously:

```
<link rel="preload" href="/style.css" as="style"
onload="this.rel='stylesheet'">
```

  - Purge unused selectors with PurgeCSS or WP Rocket's "Remove Unused CSS" option.
  - Re-test FCP after the change; target ≤1.5 s on mobile.

2D. Replace 9 repeated "read more" links with descriptive text `SEO`

- **Impact:** SEO link-text audit, screen-reader navigation
- **Problem:** PSI fails `link-text` (9 links found) and the HTML inventory shows "read more" repeated 9 times, plus "case studies" ×2 and "privacy policy" ×2.
- **Solution:**

- Give each link unique, descriptive anchor text, e.g. "Read the Acme case study" instead of "read more".
- If the visible text must stay short, add context:

```
<a href="/work/acme" aria-label="Read the Acme case study">read more</a>
```

2E. Fix hreflang URLs pointing at http:// and add valid structured data `SEO`

- **Impact:** International targeting, rich results eligibility
- **Problem:** hreflang entries resolve to `http://play.ee/` and `http://play.ee/et/` rather than HTTPS, and the single JSON-LD block is typed "Unknown" while PSI fails `structuredData`.
- **Solution:**
  - Change hreflang to HTTPS and keep them reciprocal:

```
<link rel="alternate" hreflang="en" href="https://play.ee/">
<link rel="alternate" hreflang="et" href="https://play.ee/et/">
<link rel="alternate" hreflang="x-default" href="https://play.ee/">
```

  - Replace the untyped JSON-LD with a valid `Organization` (or `ProfessionalService`) schema including `name`, `url`, `logo`, and `sameAs`.

2F. Enlarge touch targets to 44×44 px `Accessibility`

- **Impact:** WCAG 2.5.8, mobile usability
- **Problem:** PSI fails the `tapTargets` SEO audit on mobile, and axe's manual-check list flags touch target sizing as unverified.
- **Solution:**
  - Ensure every interactive element (nav links, social icons, language switcher) has at least a 44×44 px hit area:

```
.icon-link { display:inline-flex; align-items:center; justify-content:center; min-width:44px; min-height:44px; }
```

  - Add spacing between adjacent targets so they do not overlap.

2G. Eliminate render-blocking CSS and trim unused CSS/JS `Performance`

- **Impact:** Speed Index, FCP, LCP
- **Problem:** Speed Index is 4.14 s on mobile versus 519 ms on desktop; PSI estimates 1,050 ms savings from render-blocking resources and 31 KiB from

unused CSS, plus 23 KB of unused jQuery.

- **Solution:**
  - Inline critical above-the-fold CSS and defer the rest (`<link rel="preload" as="style" onload="this.rel='stylesheet'">`).
  - Drop unused CSS rules (31 KiB) with a coverage pass.
  - Replace the 23 KB `jquery.min.js` dependency with vanilla JS or drop it if only used by legacy widgets.
  - Verify the `network-dependency-tree-insight` chain — the console also logs "Couldn't load preload assets".

2H. Eliminate render-blocking CSS and unused CSS to cut Speed Index
`Performance`

- **Impact:** Speed Index (3.97 s mobile), FCP, render-blocking-insight (980 ms est. savings)
- **Problem:** PSI flags `render-blocking-insight` with ~980 ms estimated savings and `unused-css-rules` with ~30 KiB savings; Speed Index is 3.97 s on mobile versus 497 ms on desktop.
- **Solution:**
  - Enable WP Rocket's **Remove Unused CSS** (or Optimize CSS Delivery) so only above-the-fold rules load synchronously.
  - Inline critical CSS for the hero and defer `global.c0787c93fa3aa6aefb86.min.css` (32.4 KB) with `media="print" onload="this.media='all'"` .
  - Audit the 3 inline stylesheets (9.6 KB) for rules that can be merged into the critical block.

2I. Give repeated "visit the website" links unique accessible names
`Accessibility`

- **Impact:** WCAG 2.4.4 link purpose, screen-reader navigation, SEO anchor diversity
- **Problem:** 21 anchors read "visit the website", 21 read "visit website", and 15 read "read more about the project" — indistinguishable out of context.
- **Solution:**
  - Add a project-specific `aria-label` to each card link:

```
<a href="/case-study/tallink/" aria-label="Visit the Tallink case
study website">visit the website</a>
```

  - Or extend the visible text to include the client name.

- Same treatment for the 15 "read more about the project" and 3 social links.

2J. Tighten the Content-Security-Policy beyond frame-ancestors `Security`

- **Impact:** XSS defense-in-depth, script injection surface
- **Problem:** CSP is `frame-ancestors 'self';` only — no `default-src`, no `object-src 'none'`, so no script or resource restrictions are enforced.
- **Solution:** Add the missing directives while keeping the existing framing rule:

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

  - Roll out in Report-Only first to catch violations from WP Rocket and the jsdelivr bundles.
  - Note: this site has no auth, payments or user content, so CSP is defense-in-depth rather than a P1 exposure.

2K. Add Subresource Integrity to the two jsdelivr scripts `Security`

- **Impact:** Supply-chain tampering risk on third-party CDN scripts
- **Problem:** 2 of 2 cross-origin scripts lack an `integrity` attribute — `cookieconsent.umd.js` and `iframemanager.js` are both loaded from cdn.jsdelivr.net.
- **Solution:** Pin versions and add 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`.
  - Alternatively self-host both libraries so they fall under `script-src 'self'`.

2L. Eliminate render-blocking CSS and trim unused CSS `Performance`

- **Impact:** Speed Index, FCP, LCP
- **Problem:** PSI flags `render-blocking-insight` with est. 1,110 ms savings and `unused-css-rules` with 29 KiB unused; mobile Speed Index is 4.04 s vs 485 ms on desktop.
- **Solution:**

- Inline the critical above-the-fold CSS and load the rest with `<link rel="preload" as="style" onload="this.rel='stylesheet'">`.
- Purge unused selectors from the theme CSS (WP Rocket's "Remove Unused CSS" option can do this).
- Add `<link rel="preload">` for the hero font/asset — the console already warns "Couldn't load preload assets".
- Drop the 23 KB jQuery bundle if the theme's remaining scripts don't depend on it.

2M. Fix heading structure and vague repeated link text `Accessibility`

- **Impact:** Document outline, screen-reader navigation, SEO
- **Problem:** The page has 2 `<h1>` elements ("Your worry-free WordPress website" and "additional services to upgrade your online presence"), and "get in touch" appears as link text 14 times with no disambiguation.
- **Solution:**
  - Demote the second `<h1>` ("additional services…") to `<h2>` so the page has exactly one `<h1>`.
  - Give each repeated "get in touch" link a distinct accessible name, e.g. `<a href="/contact/" aria-label="Get in touch about WordPress support">`.
  - Same for the repeated social links ("view our instagram feed" ×3, etc.) — add `aria-label` naming the destination.

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

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is present but only sets `frame-ancestors 'self';` — no `default-src`, no `object-src 'none'`, and no script/resource restrictions. The site also serves a WordPress login at the default `/wp-login.php` path, so an admin session is a real XSS target even though the page itself has no auth/payment/UGC signals.
- **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'; upgrade-insecure-requests
```

  - Generate a fresh nonce per request and add it to every `<script>` tag.
  - Roll out in `Content-Security-Policy-Report-Only` first to catch breakage from the jsdelivr and fast.fonts.net dependencies.

2O. Cut render-blocking CSS and unused CSS to fix mobile Speed Index `Performance`

- **Impact:** Speed Index (4.03 s mobile), FCP (1.83 s), render-blocking (est. 1,050 ms)
- **Problem:** PSI flags `render-blocking-insight` with est. 1,050 ms savings and `unused-css-rules` with 29 KiB unused; mobile Speed Index is 4.03 s versus 564 ms on desktop.
- **Solution:**
  - Inline critical above-the-fold CSS and load the rest asynchronously:

```
<link rel="preload" href="/style.css" as="style"
onload="this.onload=null;this.rel='stylesheet'">
<noscript><link rel="stylesheet" href="/style.css"></noscript>
```

  - Purge unused selectors from the theme CSS (29 KiB reported) — WP Rocket's "Remove Unused CSS" option can do this.
  - Also address the console warning `Couldn't load preload assets` — a `<link rel=preload>` is pointing at a resource that never resolves.

2P. Remove `defer` from WP Rocket lazyload data-block scripts `Best Practices`

- **Impact:** W3C validity (4 errors)
- **Problem:** W3C reports 4 errors, all the same rule: a `<script type="text/rocketlazyloadscript" … defer>` data block must not carry `defer` (first at line 2395).
- **Solution:**
  - In WP Rocket → File Optimization → JavaScript, disable "Delay JavaScript execution" for these handles, or
  - Filter the tag to drop the attribute:

```
add_filter( 'rocket_delay_js_exclusions', function( $excluded ) {
    $excluded[] = 'jquery-js';
    return $excluded;
} );
```

  - Re-run the W3C validator to confirm 0 errors.

2Q. Use a single h1 and disambiguate repeated link text `SEO`

- **Impact:** Document outline, anchor-text relevance
- **Problem:** The page has 2 `<h1>` elements ("our strong suit" plus a second) and repeated vague anchors — "view our instagram feed" ×3, "visit our facebook page"

×3, "visit our linkedin page" ×3, "eng" ×3, "est" ×3.

- **Solution:**
  - Keep exactly one `<h1>`; demote the second to `<h2>`.
  - Give each repeated social link a distinct accessible name, e.g. `<a href="…" aria-label="View our Instagram feed – project photos">`.
  - Replace the bare `eng` / `est` language switcher text with `aria-label="English"` / `aria-label="Eesti"`.

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'`, so no script or resource restrictions exist. Site signals show no auth, payments, UGC, or commerce, so exploitation value is low.
- **Solution:**
  - Add a baseline policy now and move to nonces when scripts are consolidated:

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

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

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

3B. Add SRI to cross-origin scripts and set COOP/CORP `Security`

- **Impact:** Supply-chain integrity, origin isolation
- **Problem:** Both cross-origin scripts (jsDelivr cookieconsent and iframemanager) lack `integrity` attributes, and Cross-Origin-Opener-Policy / Cross-Origin-Resource-Policy headers are missing.
- **Solution:**
  - Add SRI hashes to the two jsDelivr tags:

```
<script
src="https://cdn.jsdelivr.net/gh/orestbida/cookieconsent@3.0.1/dist
/cookieconsent.umd.js"
```

```
        integrity="sha384-…" crossorigin="anonymous" defer>
</script>
```

- Send `Cross-Origin-Opener-Policy: same-origin` and `Cross-Origin-Resource-Policy: same-origin`.

3C. Add HSTS preload and reduce technology disclosure `Security`

- **Impact:** TLS downgrade resistance, fingerprinting surface
- **Problem:** HSTS is `max-age=31536000; includeSubDomains` without `preload`, the `Server: Apache` header is exposed, and the WordPress login is served on the default `/wp-login.php` path.
- **Solution:**
  - Append `preload` to HSTS and submit the domain at hstspreload.org:

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

  - Suppress the server banner (`ServerTokens Prod` in Apache).
  - Move the login form off `/wp-login.php` or front it with rate limiting / 2FA.

3D. Fix 4 W3C errors from defer on a data-block script `Best Practices`

- **Impact:** HTML validity
- **Problem:** The validator reports 4 errors: a `<script type="text/rocketlazyloadscript">` data block carries a `defer` attribute, first at line 2065.
- **Solution:**
  - Remove `defer` from the Rocket Loader data-block script tags, or change the type to a real JavaScript MIME type if the script should execute:

```
<script type="text/rocketlazyloadscript" id="jquery-js" data-
rocket-src="…"></script>
```

3E. Drop unused jQuery and investigate the preload warning `Performance`

- **Impact:** JS bytes, console cleanliness
- **Problem:** `jquery.min.js` wastes 23 KB of unused JavaScript, and the runtime capture logged "Couldn't load preload assets".
- **Solution:**
  - Audit whether jQuery is still required; if not, remove the enqueue entirely.
  - If it is required, load it only on pages that use it.

- Fix or remove the `<link rel="preload">` that fails to resolve — check the exact URL in DevTools Network → Priority.

3F. Add explicit dimensions to the one unsized image `Performance`

- **Impact:** CLS
- **Problem:** 1 of 50 images lacks `width` / `height` attributes; the rest are SVGs with intrinsic sizing.
- **Solution:**
  - Add `width` and `height` (or an `aspect-ratio` in CSS) to the unsized element so the browser reserves space before paint.

3G. Add SRI to cross-origin scripts and enable HSTS preload `Security`

- **Impact:** Supply-chain integrity, transport hardening
- **Problem:** Both cross-origin scripts (jsdelivr iframemanager and cookieconsent) lack `integrity` attributes, and HSTS is `max-age=31536000; includeSubDomains` without `preload`.
- **Solution:**
  - Add `integrity="sha384-…" crossorigin="anonymous"` to each jsdelivr `<script>`.
  - Append `; preload` to HSTS once you're confident every subdomain is HTTPS-only, then submit at hstspreload.org.

3H. Fix invalid `defer` on Rocket Loader data-block scripts `Best Practices`

- **Impact:** HTML validity, script execution order
- **Problem:** W3C reports 4 errors: `<script type="text/rocketlazyloadscript" … defer>` — a data block type must not carry `defer` (first at line 1869).
- **Solution:** Remove `defer` from the `text/rocketlazyloadscript` blocks and let WP Rocket's loader handle timing:

```
<script type="text/rocketlazyloadscript" id="jquery-js" data-
rocket-src="…jquery.min.js"></script>
```

3I. Move the `<aside>` out of its parent landmark `Accessibility`

- **Impact:** WCAG 1.3.1 landmark structure
- **Problem:** axe flags `landmark-complementary-is-top-level` (moderate): the `<aside>` is nested inside another landmark, so screen-reader landmark navigation is muddled.

- **Solution:** Move the `<aside>` to be a direct child of `<body>` (sibling of `<main>`), or change it to a `<div>` if it is decorative rather than complementary content.

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

- **Impact:** Repeat-visit TTFB, back/forward navigation
- **Problem:** The document response sends `cache-control: max-age=0`, forcing revalidation on every navigation despite WP Rocket being active.
- **Solution:** Set a short but non-zero HTML cache lifetime with a stale-while-revalidate window:

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

3K. Add valid structured data and disambiguate repeated link text `SEO`

- **Impact:** Rich results eligibility, link purpose
- **Problem:** PSI's `structuredData` and `tapTargets` audits fail, the single JSON-LD block is typed "Unknown", and link text repeats ("view our instagram feed" ×3, "visit our facebook page" ×3, "eng"/"est" ×3).
- **Solution:**
  - Emit a typed schema.org graph (e.g. `Organization` + `CollectionPage` / `ItemList` for the case-study index) instead of the untyped block.
  - Add `aria-label` to repeated social/language links, e.g. `<a href="…" aria-label="gotoAndPlay on Instagram">view our instagram feed</a>`.
  - Increase tap target size to ≥44×44 px for the failing `tapTargets` audit.

3L. Fix the 4 W3C errors from the rocketlazyloadscript pattern `Best Practices`

- **Impact:** HTML validity, validator error count
- **Problem:** W3C reports 4 errors: a `script` element with `type="text/rocketlazyloadscript"` (a data block, not a JS MIME type) must not carry `defer`, first at line 2616.
- **Solution:**
  - In WP Rocket, disable **Delay JavaScript execution** for the affected inline snippets, or
  - Remove the `defer` attribute from those `type="text/rocketlazyloadscript"` tags so the validator no longer flags them.
  - Re-run the W3C validator after the change; target 0 errors.

3M. Drop or defer the unused jQuery bundle `Performance`

- **Impact:** Unused JavaScript (23 KB), main-thread work
- **Problem:** PSI reports 23 KB wasted from `jquery.bfe1bb19d13b3c17b682.min.js` (30.8 KB transferred in the runtime capture).
- **Solution:**
  - Audit theme scripts for jQuery dependencies and migrate remaining ones to vanilla JS.
  - If jQuery is only needed for the cookie-consent/iframe-manager widgets, load it on interaction rather than on page load.
  - Otherwise keep it deferred and drop any jQuery plugins the theme no longer uses.

3N. Emit valid structured data with a known @type `SEO`

- **Impact:** Rich results eligibility, `structuredData` modeled audit
- **Problem:** The single JSON-LD block is reported as valid but its type resolves to "Unknown", and PSI's modeled `structuredData` audit fails.
- **Solution:**
  - Declare an explicit `@type` (e.g. `Organization`, `WebSite`, `BreadcrumbList`, `CollectionPage`) and `@context: https://schema.org`.
  - Validate with the Rich Results Test and Schema Markup Validator after the change.

3O. Flatten the heading outline and label the unheaded section `Accessibility`

- **Impact:** Document outline, screen-reader navigation
- **Problem:** The page has 1 `h1`, 30 `h2` and zero `h3`–`h6`, and the W3C validator warns that a `<section class="section section--no-padding content-1-1">` at line 2410 lacks a heading.
- **Solution:**
  - Introduce `h3` for project sub-details (client, year, stack) so the outline is not a flat list of 30 siblings.
  - Either add an `h2` to the flagged section or change it to a `<div>` if it is purely presentational.

3P. Set a real cache lifetime on the HTML document `Performance`

- **Impact:** Repeat-visit TTFB, cache hit ratio

- **Problem:** The document response sends `cache-control: max-age=0` with an `expires` equal to the request time, so browsers revalidate the HTML on every navigation despite WP Rocket being active.
- **Solution:**
  - Serve the HTML with a short but non-zero TTL, e.g. `Cache-Control: public, max-age=600, s-maxage=3600, stale-while-revalidate=60`.
  - Keep `Vary: Accept-Encoding` and purge the page from the WP Rocket cache on publish.

3Q. Add COOP/CORP, HSTS preload, and hide the Server header `Security`

- **Impact:** Origin isolation, HSTS preload eligibility, technology disclosure
- **Problem:** `Cross-Origin-Opener-Policy` and `Cross-Origin-Resource-Policy` are missing, HSTS lacks the `preload` directive, and `Server: Apache` discloses the stack.
- **Solution:**

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

  - Submit the domain at hstspreload.org once the preload directive is live.
  - Low priority here: no auth or payment surface, so origin isolation is hardening rather than a live exposure.

3R. Verify touch target sizes flagged by the tapTargets audit `Accessibility`

- **Impact:** WCAG 2.5.8 target size, mobile usability
- **Problem:** PSI's modeled `tapTargets` audit fails on mobile, and axe's manual-check list flags 44×44 px minimum targets.
- **Solution:**
  - Audit the header nav, social icons and language switcher on a 375 px viewport.
  - Ensure each interactive element has at least a 44×44 px hit area, e.g. `padding: 12px; min-width: 44px; min-height: 44px;`.
  - Re-run PSI mobile after the change to confirm `tapTargets` passes.

3S. Fix the 5 W3C validation errors from WP Rocket's lazy-load script type `Best Practices`

- **Impact:** HTML validity

- **Problem:** W3C reports 5 errors, all of the form "A `script` element with a `type` attribute … must not have a `defer` / `async` attribute" — 4 `defer` and 1 `async` on `<script type="text/rocketlazyloadscript">` blocks (first at line 350).
- **Solution:** WP Rocket rewrites script tags to `type="text/rocketlazyloadscript"` and keeps the original `async` / `defer` attributes, which is invalid HTML. Options:
  - Enable WP Rocket's "Delay JavaScript execution" off, or use its native defer handling instead of the type-swap.
  - If you must keep the swap, strip `async` / `defer` from the rewritten tags via a small output-buffer filter so the markup validates.

3T. Add width/height to the 16 images missing dimensions and enlarge tap targets `Accessibility`

- **Impact:** CLS risk, WCAG 2.5.8 target size
- **Problem:** 16 of 31 images have no explicit `width` / `height`, and PSI's modeled `tapTargets` audit fails on mobile.
- **Solution:**
  - Add intrinsic `width` / `height` (or `aspect-ratio` in CSS) to every `<img>` and inline SVG wrapper so the browser can reserve space.
  - Ensure interactive elements are at least 44×44 px with adequate spacing; the icon-only social links are the likely offenders.

3U. Complete the advanced header set: HSTS preload, COOP, CORP, SRI `Security`

- **Impact:** Transport hardening, origin isolation, supply-chain integrity
- **Problem:** HSTS is `max-age=31536000; includeSubDomains` but lacks `preload`; `Cross-Origin-Opener-Policy` and `Cross-Origin-Resource-Policy` are missing; 2 of 2 cross-origin scripts (jsdelivr) lack `integrity` attributes.
- **Solution:**

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

  - Add `integrity="sha384-…"` and `crossorigin="anonymous"` to the two `cdn.jsdelivr.net` script tags (iframemanager and cookieconsent).
  - Submit the domain at hstspreload.org once the header is live.
  - Also strip or genericize the `Server: Apache` response header.

3V. Harden the WordPress login path and validate structured data `SEO`

- **Impact:** Brute-force exposure, rich-result eligibility
- **Problem:** The login form is served on the default `/wp-login.php` path, and PSI's modeled `structuredData` audit fails even though one JSON-LD block is present (type reported as "Unknown").
- **Solution:**
  - Move the login to a custom path (e.g. via WPS Hide Login) and add rate limiting / 2FA on `/wp-login.php`.
  - Replace the generic JSON-LD block with explicit `@type` values — `Organization`, `Service`, and `FAQPage` / `Offer` for the pricing tiers — and validate with Google's Rich Results Test.

3W. Harden the existing CSP (default-src, object-src 'none') `Security`

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is present but only `frame-ancestors 'self';` — no `default-src` and no `object-src 'none'`, so it restricts nothing but framing. Site signals show no auth, payments, UGC or commerce, so this is hardening rather than an active exposure.
- **Solution:** Add the missing directives, ideally with a nonce + `'strict-dynamic'` for scripts rather than a flat allowlist:

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

  - Roll out in Report-Only first to catch breakage from the jsDelivr and fast.fonts.net third parties.

3X. Add SRI to cross-origin scripts, COOP/CORP, and HSTS preload `Security`

- **Impact:** Supply-chain integrity, origin isolation, HSTS coverage
- **Problem:** 2 of 2 cross-origin scripts (jsDelivr cookieconsent and iframemanager) lack `integrity`; COOP and CORP headers are missing; HSTS has `max-age=31536000; includeSubDomains` but no `preload`; the `Server: Apache` header discloses the stack.
- **Solution:**

```
<script
src="https://cdn.jsdelivr.net/gh/orestbida/cookieconsent@3.0.1/dist
/cookieconsent.umd.js"
```

```
        integrity="sha384-…" crossorigin="anonymous" defer>
</script>
```

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

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

3Y. Validate the JSON-LD block and check tap targets `SEO`

- **Impact:** Rich results eligibility, mobile usability
- **Problem:** PSI's modeled audits flag `structuredData` and `tapTargets` as failing, and the single JSON-LD block is reported with type "Unknown" rather than a recognized schema.org type.
- **Solution:**
  - Inspect the JSON-LD block and set an explicit `@type` (e.g. `Organization`, `WebSite`, `Service`) with required properties.
  - Test with Google's Rich Results Test and Schema Markup Validator.
  - Audit interactive elements for the 44×44 px minimum touch target (WCAG 2.5.8) on mobile.