Audit Report: Perfectly formed web development team - gotoAndPlay

Website: https://play.ee/
Date: 26.10.2026
Audit Coverage: 100% — all sources returned data
Confidence: high

Pages Audited (1 of 1):

- https://play.ee/

Summary of results

Overall Score: 86 / 100
Status: **Needs Improvement**

Strong, well-built site: PSI mobile 94 / desktop 100, with clean Core Web Vitals (LCP 2.5 s, CLS 0.000, TBT 0 ms, TTFB 44 ms) and a very light 171 KB page weight served brotli-compressed behind WP Rocket. The main performance drag is mobile-only: Speed Index 4.04 s vs 580 ms on desktop, with PSI estimating 1,280 ms of savings from render-blocking resources and a deep network dependency tree. Accessibility is close to clean — axe-core reports 0 violations across 36 passes — but Lighthouse still fails `color-contrast` (weight 7) and axe could not determine contrast on 37 nodes, so the real contrast state is unverified. SEO sits at 92 with failing `tapTargets`, `structuredData`, and `link-text` (9 non-descriptive links, including "read more" ×9), plus hreflang entries pointing at `http://` URLs. Security basics PASS, but advanced checks fail: CSP is `frame-ancestors 'self'` only (no `default-src`, no `object-src 'none'`), HSTS lacks preload, COOP/CORP are missing, and both cross-origin jsDelivr scripts lack SRI — these keep the score out of the 90s alongside 4 W3C errors.

Per-page scores

 Needs Improvement · https://play.ee/

Score	Performance	Accessibility	Best Practices	SEO	Security
86	94	96	100	92	PASS

PageSpeed Insights — Mobile vs Desktop

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

Strategy	Performance (M / D)	LCP (M / D)	CLS (M / D)
Mobile vs Desktop	**94** / 100	**2.52 s** / 570 ms	0.000 / **0.007**

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. Resolve color-contrast failures and verify the 37 undetermined nodes `Accessibility`

- **Impact:** WCAG 1.4.3 contrast, PSI Accessibility score
- **Problem:** Lighthouse fails `color-contrast` (score 0.00, weight 7) on mobile and desktop, while axe-core returned 37 nodes it could not evaluate for contrast — so the real extent is unknown.
- **Solution:**
  - Run axe DevTools or Chrome DevTools contrast picker on the 37 undetermined nodes (likely text over gradients/alpha backgrounds).
  - Raise any failing text to ≥4.5:1 (≥3:1 for large text ≥24 px or ≥18.66 px bold).
  - For text over images/gradients, add a solid overlay or `text-shadow` so the computed background is deterministic.

1B. Eliminate render-blocking resources and flatten the network dependency tree `Performance`

- **Impact:** Speed Index, FCP, LCP (mobile)
- **Problem:** PSI flags `render-blocking-insight` with an estimated 1,280 ms of savings and `network-dependency-tree-insight` as high priority; mobile Speed Index is 4.04 s vs 580 ms on desktop.
- **Solution:**
  - Inline critical CSS for above-the-fold content and load the rest with `media="print" onload="this.media='all'"`.
  - Add `<link rel="preload">` for the LCP font/hero asset and `<link rel="preconnect">` for `fast.fonts.net` and `cdn.jsdelivr.net`.
  - Defer or move the remaining render-blocking script out of the critical path.
  - Audit the 22 scripts (only 3 defer, 0 async) — most can be deferred or bundled.

Priority 2: Important

Essential for compliance, user reach, and search visibility.

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

- **Impact:** XSS defense-in-depth, admin session protection
- **Problem:** CSP is only `frame-ancestors 'self';` — no `default-src`, no `object-src 'none'`, and no script/resource restrictions. A WordPress login form is served at the default `/wp-login.php` path, so an XSS would have a real admin session to steal.
- **Solution:** Deploy 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:;
  style-src 'self' 'unsafe-inline'; img-src 'self' data: https:;
  frame-ancestors 'self'; form-action 'self';
```

Roll it out in `Content-Security-Policy-Report-Only` first, fix violations, then enforce.

2B. Complete HSTS and add COOP/CORP `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:**

```
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 after confirming all subdomains are HTTPS-only.

2C. Replace vague repeated link text with descriptive labels `SEO`

- **Impact:** SEO (link-text audit), screen-reader link purpose (WCAG 2.4.4)
- **Problem:** PSI fails `link-text` on 9 links; the inventory shows "read more" ×9, "view our instagram feed" ×3, "visit our facebook page" ×3, "visit our linkedin page" ×3, and "case studies" ×2.
- **Solution:**
  - Rewrite each "read more" to name its destination, e.g. `<a href="/work/acme">Read the Acme case study</a>`.
  - If the visible text must stay short, add `aria-label`:

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

  - Social links: use `aria-label="gotoAndPlay on Instagram"` etc. so each is unique.

2D. Fix tap targets and add valid structured data `SEO`

- **Impact:** SEO score, mobile usability, rich results eligibility

- **Problem:** PSI SEO is 92 with `tapTargets` and `structuredData` both failing; the single JSON-LD block is typed "Unknown" rather than a recognized schema.org type.
- **Solution:**
  - Ensure every interactive element is ≥48×48 px with ≥8 px spacing (WCAG 2.5.8 / Lighthouse tap targets).
  - Replace the JSON-LD with a valid `Organization` + `WebSite` graph:

```
{"@context":"https://schema.org","@type":"Organization","name":"got
oAndPlay","url":"https://play.ee/","logo":"https://play.ee/logo.png
"}
```

  - Validate with Google's Rich Results Test.

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

- **Impact:** Supply-chain tampering risk
- **Problem:** Both cross-origin scripts from `cdn.jsdelivr.net` (iframemanager 1.3.0, cookieconsent 3.0.1) lack an `integrity` attribute.
- **Solution:**

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

  Generate the hash with `openssl dgst -sha384 -binary file.js | openssl base64 -A`. Better still, self-host both libraries.

Priority 3: Best Practice

Recommended for long-term maintainability.

3A. Fix the 4 W3C validation errors on lazyloaded scripts `Best Practices`

- **Impact:** HTML validity
- **Problem:** W3C reports 4 errors, all the same: a `<script type="text/rocketlazyloadscript">` data block carries a `defer` attribute, first at line 2065.
- **Solution:** WP Rocket's `text/rocketlazyloadscript` is a data block, not a JavaScript MIME type, so `defer` is invalid. Either drop `defer` from those tags or switch to a real module type:

```
<script type="text/rocketlazyloadscript" data-rocket-src="...">
</script>
```

3B. Trim unused JavaScript and enable document caching `Performance`

- **Impact:** Page weight, repeat-visit TTFB
- **Problem:** `jquery.min.js` ships 23 KB of unused JS, and the document response is `cache-control: max-age=0` with an `expires` equal to the request time, so repeat visits re-download the 219 KB HTML.
- **Solution:**
  - Drop jQuery if the theme's remaining scripts don't depend on it, or load only the modules actually used.
  - Let WP Rocket serve a short browser cache for the HTML, e.g. `Cache-Control: public, max-age=60, s-maxage=600, stale-while-revalidate=86400`.

3C. Point hreflang alternates at HTTPS URLs `SEO`

- **Impact:** International targeting, canonical consistency
- **Problem:** The hreflang set uses `http://play.ee/` and `http://play.ee/et/` while the canonical and final URL are `https://`, creating a protocol mismatch for alternate-language pages.
- **Solution:**

```
<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/"
/>
```

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

- **Impact:** Attack-surface reduction
- **Problem:** The `Server: Apache` header discloses the web server, and the WordPress login form is served on the default `/wp-login.php` path.
- **Solution:**
  - Set `ServerHeader off` / `ServerTokens Prod` in Apache to stop version/banner leakage.
  - Put `/wp-login.php` behind a custom path or IP allowlist, and add rate limiting plus 2FA.