Audit Report: Perfectly formed web development team - gotoAndPlay

Website: https://play.ee/
Date: 02.10.2026
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: 77 / 100
Status: □ **Needs Improvement**

Site overall 77 is the mean of 5 pages. Scores range 74 (https://play.ee/) → 79 (https://play.ee/web-development-case-studies/). Weakest page: PSI mobile performance is excellent at 94 with strong Core Web Vitals, but the Security basics verdict is FAILED. These are high-level hygiene checks on HTTP headers, TLS, cookies and a few well-known exposed paths. PASS means the basics are in place. It does **not** mean the site is secure: application logic, authentication, plugins, server configuration and dependencies are not tested here. Treat FAILED as a must-fix and PASS as a starting point, not a certificate. W3C validation shows 7 errors including structural issues in the head, and accessibility lacks a skip-link and main landmark. The overall score reflects high technical performance weighed down by critical security hygiene gaps and HTML validity issues.

Per-page scores

□ **Needs Improvement** · https://play.ee/

Score	Performance	Accessibility	Best Practices	SEO	Security
74	94	94	100	92	FAILED

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

Score	Performance	Accessibility	Best Practices	SEO	Security
79	93	90	96	100	FAILED

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

Score	Performance	Accessibility	Best Practices	SEO	Security
78	93	90	96	100	FAILED

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

Score	Performance	Accessibility	Best Practices	SEO	Security
76	93	91	100	100	FAILED

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

Score	Performance	Accessibility	Best Practices	SEO	Security
78	93	94	100	100	FAILED

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/	**94** / 100	**2.54 s** / 608 ms	0.000 / **0.007**
https://play.ee/web-development-case-studies/	**93** / 100	**2.58 s** / 578 ms	0.000 / **0.005**
https://play.ee/software-development-work-index/	**93** / 100	**2.61 s** / 531 ms	0.000 / **0.003**

URL	Performance (M / D)	LCP (M / D)	CLS (M / D)
https://play.ee/wordpress-support-service/	**93** / 100	**2.57 s** / 629 ms	0.000 / **0.002**
https://play.ee/web-development-in-estonia/	**93** / 100	**2.58 s** / 567 ms	0.000 / **0.005**

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 (37 SVGs, 13 placeholders excluded).
Hero image eagerly loaded	N/A	No raster elements found (37 SVGs, 13 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 (37 SVGs, 13 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. Fix HTTP redirect and add HSTS `Security`

- **Impact:** Transport security, HTTPS enforcement
- **Problem:** Security basics FAILED: HTTP does not redirect to HTTPS and Strict-Transport-Security header is missing.
- **Solution:** Configure server to redirect all HTTP traffic to HTTPS and send HSTS header:

```
Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains; preload"
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

1B. Add X-Content-Type-Options header `Security`

- **Impact:** MIME sniffing protection
- **Problem:** Security basics FAILED: X-Content-Type-Options header is missing.
- **Solution:** Send the following header with all responses:

```
Header always set X-Content-Type-Options "nosniff"
```

1C. Disable xmlrpc.php POST requests `Security`

- **Impact:** Brute-force / Pingback vector
- **Problem:** WordPress security check failed: xmlrpc.php accepts POST requests (brute-force / pingback vector).
- **Solution:** Block POST requests to xmlrpc.php in .htaccess or via security plugin:

```
<Files xmlrpc.php>
  <Limit POST>
    deny from all
  </Limit>
</Files>
```

1D. Fix HTTP redirect and add baseline security headers `Security`

- **Impact:** Transport security, clickjacking, MIME sniffing
- **Problem:** Security Basics verdict is FAILED: HTTP does not redirect to HTTPS, HSTS is missing, and X-Content-Type-Options is missing.

- **Solution:** Configure server to force HTTPS and send required headers (Apache example):

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]

Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains"
Header always set X-Content-Type-Options "nosniff"
```

1E. Enforce HTTPS redirect and add HSTS `Security`

- **Impact:** Transport security, data integrity
- **Problem:** Security Basics FAILED: HTTP does not redirect to HTTPS and Strict-Transport-Security is missing.
- **Solution:** Configure server to redirect all HTTP traffic to HTTPS and send HSTS header:

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains; preload"
```

1F. Enforce HTTPS and add baseline security headers `Security`

- **Impact:** Transport security, clickjacking, MIME sniffing
- **Problem:** Security basics FAILED: HTTP does not redirect to HTTPS, HSTS is missing, and X-Content-Type-Options is missing.
- **Solution:** Configure server to redirect all HTTP traffic to HTTPS and send these headers:

```
Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains"
Header always set X-Content-Type-Options "nosniff"
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

1G. Harden WordPress XML-RPC and Installer paths `Security`

- **Impact:** Brute-force protection, installation security
- **Problem:** xmlrpc.php accepts POST requests (brute-force vector) and /wp-admin/install.php is reachable.

- **Solution:** Block xmlrpc.php POST requests via .htaccess or plugin, and restrict access to install.php:

```
# Block xmlrpc.php POST
<Files xmlrpc.php>
  <Limit POST>
    deny from all
  </Limit>
</Files>
# Block install.php
<Files install.php>
  deny from all
</Files>
```

1H. Disable WordPress xmlrpc.php `Security`

- **Impact:** Brute-force protection
- **Problem:** xmlrpc.php accepts POST requests, creating a brute-force and pingback vector.
- **Solution:** Block xmlrpc.php in .htaccess or via security plugin:

```
<Files xmlrpc.php>
  Order Deny,Allow
  Deny from all
</Files>
```

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Fix W3C HTML validation errors `Best Practices`

- **Impact:** Parser recovery, SEO, Rendering
- **Problem:** W3C Validator reports 7 errors including bad start tag in iframe in noscript in head and stray end tags.
- **Solution:** Move the Google Tag Manager iframe out of the `<head>` or ensure it is properly closed. Remove stray `</noscript>` and `</head>` tags. Ensure meta tags are placed correctly within `<head>`.

2B. Add skip-to-content link and main landmark `Accessibility`

- **Impact:** Keyboard navigation, Screen reader flow
- **Problem:** HTML Inventory shows missing main landmark and missing skip-to-content link; axe-core reports region violations.
- **Solution:** Add a skip link at the top of the body:

```
<a href="#main-content" class="skip-link">Skip to content</a>
```

Wrap the primary content in `<main id="main-content">`.

2C. Add explicit width and height to images `Performance`

- **Impact:** CLS (Cumulative Layout Shift)
- **Problem:** HTML Inventory shows 50 images without width/height attributes, risking layout shifts despite current CLS score.
- **Solution:** Add `width` and `height` attributes to all `<img>` tags matching the intrinsic dimensions:

```
<img src="image.jpg" alt="..." width="800" height="600">
```

2D. Fix color contrast and add landmarks `Accessibility`

- **Impact:** WCAG 1.4.3 contrast, 1.3.1 info/relationships
- **Problem:** 1 serious axe violation on color-contrast (h1, cards) and missing main landmark/skip-to-content link.
- **Solution:**
  - Increase text contrast to ≥4.5:1 for `.heading__main` and card text.
  - Add `<main id="main">` wrapper around primary content.
  - Add skip link at top: `<a href="#main" class="skip-link">Skip to content</a>`.

2E. Resolve W3C HTML validation errors `SEO`

- **Impact:** Parsing consistency, SEO rendering
- **Problem:** 7 errors including parser recovery failure at line 101 (iframe/noscript in head, stray end tags).
- **Solution:**
  - Move `<noscript><iframe>...</iframe></noscript>` out of `<head>` into `<body>`.
  - Remove `name` attribute from `<meta>` tags where not allowed.
  - Ensure `<head>` closes before `<body>` starts.

2F. Fix color contrast on headings and spans `Accessibility`

- **Impact:** WCAG 1.4.3 compliance
- **Problem:** axe-core reports 1 serious violation: color-contrast on `h1 > .heading__main` and `p > span`.

- **Solution:** Increase contrast ratio to at least 4.5:1 for normal text. Adjust CSS:

```
.heading__main { color: #333333; } /* Example darker shade */
```

2G. Eliminate render-blocking resources `Performance`

- **Impact:** LCP, FCP, Speed Index
- **Problem:** PSI reports render-blocking insight with estimated savings of 1,250 ms, despite checklist passing.
- **Solution:** Defer non-critical CSS/JS or inline critical CSS. Check for blocking scripts in `<head>`:

```
<script src="..." defer></script>
<link rel="preload" href="critical.css" as="style">
```

2H. Add X-Content-Type-Options header `Security`

- **Impact:** MIME sniffing protection
- **Problem:** Security Basics FAILED: X-Content-Type-Options header is missing.
- **Solution:** Add the following header to prevent MIME type sniffing:

```
Header always set X-Content-Type-Options "nosniff"
```

2I. Fix color contrast and link names `Accessibility`

- **Impact:** WCAG 1.4.3 contrast, 2.4.4 link purpose
- **Problem:** 2 serious axe violations: color-contrast on .button--tertiary and headings; link-name missing on logo grid links.
- **Solution:**
  - Increase contrast ratio for `.button--tertiary` text to ≥4.5:1.
  - Add `aria-label` to logo links: `<a href="" aria-label="Partner Name"><img ...></a>`.

2J. Correct HTML structure and landmarks `SEO`

- **Impact:** Document outline, screen reader navigation
- **Problem:** W3C reports 7 errors including 2 h1 elements, missing `main` landmark, and parser recovery failure at line 107.
- **Solution:**
  - Ensure only one `<h1>` per page.
  - Wrap main content in `<main>`.

- Fix `<noscript><iframe>` nesting in `<head>` (move to `<body>` or remove if unnecessary).

2K. Fix color contrast and add main landmark `Accessibility`

- **Impact:** WCAG 1.4.3, 1.3.1
- **Problem:** 1 serious color-contrast violation on headings; document lacks a landmark and skip link.
- **Solution:**
  - Increase text contrast to ≥4.5:1 (e.g., darken text or lighten background).
  - Wrap primary content in `<main>` and add a skip link: `<a href="#main-content" class="skip-link">Skip to content</a>`.

Priority 3: Best Practice

Recommended for long-term maintainability.

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

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is present but weak: missing default-src and object-src 'none'. Site signals show no auth/payments/UGC.
- **Solution:** Since the site has no auth or user content, a strict CSP is lower priority but recommended for defense-in-depth:

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

3B. Eliminate render-blocking resources `Performance`

- **Impact:** LCP, FCP, Speed Index
- **Problem:** PSI estimates 1,270 ms savings from render-blocking insight; 1 render-blocking script detected.
- **Solution:**
  - Add `defer` or `async` to non-critical scripts.
  - Inline critical CSS if possible.
  - Use `<link rel="preload">` for critical fonts/scripts.

3C. Add explicit dimensions to images `Best Practices`

- **Impact:** CLS, Layout stability
- **Problem:** 56 images lack width/height attributes, risking layout shifts on load.
- **Solution:**

- Add `width` and `height` attributes to all `<img>` tags matching intrinsic dimensions.
- Use CSS `aspect-ratio` if dimensions vary dynamically.

3D. Harden WordPress XML-RPC and CSP `Security`

- **Impact:** Brute-force protection, XSS defense
- **Problem:** xmlrpc.php accepts POST requests (brute-force vector); CSP is weak (missing default-src).
- **Solution:** Disable XML-RPC in `.htaccess` or via plugin. Strengthen CSP:

```
<Files xmlrpc.php>
  Order deny,allow
  Deny from all
</Files>
Header set Content-Security-Policy "default-src 'self'; object-src
'none';"
```

3E. Add image dimensions and lazy loading `Performance`

- **Impact:** CLS prevention, bandwidth savings
- **Problem:** 31 images lack explicit width/height and loading="lazy" attributes.
- **Solution:** Add `width` and `height` attributes to all `<img>` tags to reserve space. Add `loading="lazy"` to images below the fold:

```
<img src="image.jpg" alt="..." width="300" height="200"
loading="lazy">
```

3F. Optimize LCP and render-blocking resources `Performance`

- **Impact:** LCP, FCP, Speed Index
- **Problem:** LCP 2.6 s; PSI estimates 1,280 ms savings from render-blocking insight.
- **Solution:**
  - Preload LCP image if it is an image.
  - Defer non-critical JavaScript.
  - Inline critical CSS if render-blocking stylesheets are detected.