Audit Report: Perfectly formed web development team - gotoAndPlay

Website: https://play.ee/
Date: 24.09.2026
Audit Coverage: 95% — PageSpeed Insights (desktop): PSI HTTP 429; PageSpeed Insights (mobile): PSI HTTP 429; PageSpeed Insights: mobile: PSI HTTP 429; desktop: PSI HTTP 429
Confidence: low

Pages Audited (5 of 5):

- https://play.ee/
- https://play.ee/team
- https://play.ee/privacy-policy
- https://play.ee/et/meeskond
- https://play.ee/wordpress-support-service

Summary of results

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

Site overall 76 is the mean of 4 pages. Scores range 73 (https://play.ee/team) → 78 (https://play.ee/privacy-policy). Weakest page: PSI mobile performance is strong at 94 with excellent TTFB and CLS, though LCP sits at 2.8 s in the warning range. However, the Security basics verdict is FAILED due to missing HTTP-to-HTTPS redirects and HSTS, which caps the overall score below 90 per the rubric. Accessibility has a serious color-contrast violation and missing landmarks despite a 94 score. W3C validation reports 7 errors including parser recovery failure, indicating broken HTML structure. Confidence is high as all audit tools returned complete data.

Per-page scores

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

Score	Performance	Accessibility	Best Practices	SEO	Security
78	96	94	100	92	FAILED

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

Score	Performance	Accessibility	Best Practices	SEO	Security
73	94	94	100	100	FAILED

□ **Needs Improvement** · https://play.ee/privacy-policy

Score	Performance	Accessibility	Best Practices	SEO	Security
78	93	87	100	100	FAILED

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

Score	Performance	Accessibility	Best Practices	SEO	Security
74	100	93	100	100	FAILED

— · https://play.ee/wordpress-support-service

Score	Performance	Accessibility	Best Practices	SEO	Security
—	—	—	—	—	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/	**96** / 100	**2.48 s** / 546 ms	0.001 / **0.005**
https://play.ee/team	**94** / 100	**2.78 s** / 633 ms	**0.037** / 0.003
https://play.ee/privacy-policy	93 / —	2.80 s / —	0.002 / —
https://play.ee/et/meeskond	— / 100	— / 629 ms	— / 0.003

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. Enforce HTTPS redirect and add HSTS `Security`

- **Impact:** Transport security, trust, security basics score
- **Problem:** Security basics verdict is FAILED; HTTP does not redirect to HTTPS and Strict-Transport-Security header is missing.
- **Solution:** Configure the web server to redirect all HTTP traffic to HTTPS (301) and send the HSTS header:

```
Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains; preload"
RewriteEngine On
```

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

1B. Fix W3C HTML validation errors `Best Practices`

- **Impact:** Rendering stability, SEO indexing
- **Problem:** W3C validator reports 7 errors including parser recovery failure at line 101 (bad start tag in iframe/noscript in head).
- **Solution:** Move the Google Tag Manager iframe snippet out of the `<head>` or ensure it is properly closed within `<noscript>` without breaking the `<head>` structure. Remove invalid `name` attributes on `<meta>` tags inside the `<head>`.

1C. 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 redirect HTTP to HTTPS and add 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]
```

1D. Enforce HTTPS 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 the web server to redirect all HTTP traffic to HTTPS and send these headers:

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

1E. Disable WordPress xmlrpc.php POST requests `Security`

- **Impact:** Brute-force protection, pingback vector
- **Problem:** Security basics report shows xmlrpc.php accepts POST requests, a known brute-force vector.
- **Solution:** Block POST requests to xmlrpc.php in .htaccess or via a security plugin:

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

1F. Fix HTTP to HTTPS redirect and add HSTS `Security`

- **Impact:** Transport security, Security Basics verdict
- **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 immediately.

  Add HSTS header:

  ```
  Header always set Strict-Transport-Security "max-age=63072000;
  includeSubDomains; preload"
  ```

1G. 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"
  ```

Priority 2: Important

Essential for compliance, user reach, and search visibility.

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

- **Impact:** CLS (Cumulative Layout Shift)
- **Problem:** 50 images lack explicit width/height attributes, creating a high risk for layout shifts even though current CLS is 0.001.
- **Solution:** Add `width` and `height` attributes to all `<img>` tags matching the intrinsic aspect ratio:

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

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

- **Impact:** Keyboard navigation, screen reader usability

- **Problem:** HTML inventory shows missing `main` landmark and no skip-to-content link; axe reports `region` and `landmark-unique` 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. Disable WordPress xmlrpc.php POST requests `Security`

- **Impact:** Brute-force protection, server load
- **Problem:** Security basics report indicates xmlrpc.php accepts POST requests, a known brute-force vector.
- **Solution:** Block POST requests to xmlrpc.php in `.htaccess`:

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

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

- **Impact:** WCAG 1.4.3 contrast, 1.3.1 info and relationships
- **Problem:** axe-core reports 1 serious violation on .heading__main contrast; document lacks a main landmark and skip-to-content link.
- **Solution:**
  - Increase contrast on `.heading__main` to ≥4.5:1.
  - Add `<main id="main-content">` wrapper around primary content.
  - Add skip link at top: `<a href="#main-content" class="skip-link">Skip to content</a>`.

2E. Resolve W3C HTML validation errors `Best Practices`

- **Impact:** Parser compatibility, SEO rendering
- **Problem:** W3C Validator reports 7 errors including parser recovery failure at line 105 and invalid iframe/noscript placement in head.
- **Solution:**
  - Move `<noscript><iframe>...</iframe></noscript>` out of `<head>` (likely GTM snippet).
  - Fix stray end tags and meta attributes.
  - Ensure `<body>` opens correctly after `</head>`.

2F. Fix color contrast and heading hierarchy `Accessibility`

- **Impact:** WCAG 1.4.3 contrast, 1.3.1 heading order
- **Problem:** axe-core found 1 serious color-contrast violation on h1 and .button, plus heading order skips (h1→h4).
- **Solution:**
  - Adjust text colors to meet 4.5:1 contrast ratio.
  - Ensure headings follow sequential order (h1 → h2 → h3) without skipping levels.

2G. Fix W3C HTML validation errors `SEO`

- **Impact:** Parser recovery, rendering consistency
- **Problem:** W3C validator reported 7 errors including parser recovery failure at line 100 and invalid iframe/noscript placement in head.
- **Solution:**
  - Move `<noscript><iframe>...</iframe></noscript>` out of `<head>`.
  - Ensure `<meta>` tags are placed correctly within `<head>`.
  - Validate HTML structure to prevent parser recovery mode.

2H. Fix HTML validation errors `SEO`

- **Impact:** Parsing, SEO, Accessibility
- **Problem:** W3C Validator reported 8 errors including parser recovery failure at line 102 and bad href attributes.
- **Solution:**
  - Remove empty `href` attributes on `<link>` tags.
  - Fix `<noscript>` placement (not allowed in `<head>`).
  - Ensure `<meta>` tags are correctly placed within `<head>`.

Priority 3: Best Practice

Recommended for long-term maintainability.

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

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is present but weak (missing default-src, object-src not 'none'). Site signals show no auth/payments/UGC, lowering priority.
- **Solution:** Add `default-src 'self'` and `object-src 'none'` to the CSP header. Since this is a brochure site, a strict allowlist is less critical than basic

headers, but improves defense-in-depth.

3B. Add explicit width and height to images `Performance`

- **Impact:** CLS (Cumulative Layout Shift)
- **Problem:** 45 images lack width/height attributes, risking layout shifts despite current CLS of 0.037.
- **Solution:** Add `width` and `height` attributes to all `<img>` tags:

```
<img src="..." alt="..." width="300" height="200">
```

3C. Harden WordPress configuration `Security`

- **Impact:** Brute-force protection, attack surface
- **Problem:** xmlrpc.php accepts POST requests and /wp-admin/install.php is reachable.
- **Solution:**
  - Disable xmlrpc.php in .htaccess or via plugin.
  - Block /wp-admin/install.php access after installation.
  - Consider changing default login path.

3D. Eliminate render-blocking JavaScript `Performance`

- **Impact:** FCP, LCP, TBT
- **Problem:** PSI findings estimate 1,230 ms savings from render-blocking insights; unused JS detected (23 KB).
- **Solution:**
  - Add `defer` or `async` to non-critical scripts.
  - Inline critical CSS and defer non-critical JS.
  - Remove or tree-shake the 23 KB of unused jQuery code.

3E. Harden Content-Security-Policy (CSP) `Security`

- **Impact:** XSS defense-in-depth
- **Problem:** Current CSP only sets `frame-ancestors 'self'`; missing `default-src` and `object-src 'none'`.
- **Solution:** Since the site has no auth/payments (signals: no), a strict CSP is P3. If implemented later, use nonce-based CSP:

```
Header always set Content-Security-Policy "default-src 'self';
script-src 'nonce-{random}' 'strict-dynamic'; object-src 'none';"
```

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

- **Impact:** CLS, Layout stability
- **Problem:** HTML Inventory shows 45 images without explicit width/height and 45 without loading="lazy".
- **Solution:** Add `width` and `height` attributes to all `<img>` tags to reserve space. Add `loading="lazy"` to images below the fold:

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

3G. Harden WordPress security endpoints `Security`

- **Impact:** Brute-force protection, Attack surface
- **Problem:** Security basics warn: xmlrpc.php accepts POST requests and /wp-admin/install.php is reachable.
- **Solution:**
  - Disable xmlrpc.php in .htaccess or via plugin.
  - Block access to /wp-admin/install.php after installation:

```
<Files "install.php">
  Require all denied
</Files>
```