Audit Report: Avaleht - estply

Website: https://estply.com/
Date: 09.11.2026
Audit Coverage: 88% — Page skipped — rate limited by target (HTTP 429), Retry-After: 10 (https://estply.com/kontakt/); W3C Validator: W3C HTTP 429
Confidence: low

Pages Audited (4 of 5):

- https://estply.com/
- https://estply.com/en/privacy-policy/
- https://estply.com/privaatsuspoliitika/
- https://estply.com/kupsiste-poliitika/

Summary of results

Overall Score: 91 / 100
Status: □ □ **Excellent**

What is pulling the score down

Page score = weighted PageSpeed categories (Performance 40%, Accessibility 30%, Best Practices 15%, SEO 15%) minus at most 5 deduction points, and never below 90 when every category is ≥90 and Security basics PASS. Points = largest deduction on any page.

Signal	Points (max)	Pages	Evidence (weakest page)
Field TTFB	−2	4/4	CrUX mobile TTFB p75 914 ms (desktop 709 ms)
Advanced security headers	−1	4/4	Failing: CSP is strict, Referrer-Policy, Permissions-Policy, Cross-Origin-Opener-Policy, Cross-Origin-Resource-Policy
Image weight	−1	1/4	8 images totalling 2.66 MB, 6 of 7 raster images still JPEG/PNG

Site overall 91 is the mean of 4 pages; scores range 89 (https://estply.com/en/privacy-policy/) → 92 (https://estply.com/kupsiste-poliitika/).

Per-page scores

 Excellent · https://estply.com/

Score	Performance	Accessibility	Best Practices	SEO	Security
90	92	91	100	92	PASS

 Needs Improvement · https://estply.com/en/privacy-policy/

Score	Performance	Accessibility	Best Practices	SEO	Security
89	93	91	100	85	PASS

 Excellent · https://estply.com/privaatsuspoliitika/

Score	Performance	Accessibility	Best Practices	SEO	Security
92	96	91	100	92	PASS

 Excellent · https://estply.com/kupsiste-poliitika/

Score	Performance	Accessibility	Best Practices	SEO	Security
92	98	91	100	92	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://estply.com/	**92** / 94	**1.05 s** / 281 ms	0.000 / 0.000
https://estply.com/en/privacy-policy/	93 / 93	**1.05 s** / 281 ms	0.000 / 0.000
https://estply.com/privaatsuspoliitika/	96 / **95**	**1.05 s** / 281 ms	0.000 / **0.000**
https://estply.com/kupsiste-poliitika/	**98** / 100	**1.06 s** / 281 ms	0.000 / 0.000

Optimization Checklist

0 of 2 passing — 0 pass · 0 warn · 2 fail · 6 n/a

Item	Status	Detail
Page caching plugin / CDN active	**Fail**	No WordPress cache plugin marker or CDN edge cache detected on the document response.
Response compressed (gzip / brotli)	N/A	Response is not compressed but only 337 bytes — too small to matter.
Images lazy-loaded	N/A	No raster elements found.
Hero image eagerly loaded	**Fail**	Hero image has loading="lazy", which delays LCP (inferred from DOM order/size — Lighthouse LCP element unavailable). Use loading="eager" (or omit loading) and add fetchpriority="high".
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 2 raster images on the page — 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	N/A	No external scripts on the page.

Fixes

Priority 1: Critical

Immediate action — impacts user experience, search rankings, or site safety.

1A. Reduce server response time (TTFB) `Performance`

- **Impact:** TTFB, FCP, LCP, Speed Index
- **Problem:** CrUX mobile TTFB p75 is 914 ms and the document request took 4.01 s in the browser capture, while the lab run reports 3 ms — real users wait far longer than the lab suggests.
- **Solution:**
  - Confirm WP Rocket page cache is actually serving the canonical URL (warm the cache after deploy).
  - Add full-page/CDN edge caching in front of the origin so HTML is served from a POP near the user.
  - Check for slow plugins or uncached admin-ajax/REST calls on the homepage.
  - Verify the origin's PHP/DB latency; 914 ms p75 points at origin work, not network.

1B. Remove render-blocking and unused CSS `Performance`

- **Impact:** Speed Index, FCP, LCP
- **Problem:** Speed Index is 8.05 s on mobile (score 0.21) while LCP is only 1.1 s, and PSI's render-blocking insight estimates 710 ms of savings from 2 external stylesheets plus 9.6 KB of inline CSS.
- **Solution:**
  - Inline critical above-the-fold CSS and load the rest asynchronously:

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

  - Remove unused rules from the theme stylesheet (PSI flags unused CSS).
  - Investigate what paints late and pushes Speed Index to 8 s — likely a below-the-fold hero/CTA block or a late-loading consent/iframe script.

1C. Add accessible names to buttons `Accessibility`

- **Impact:** Screen-reader and keyboard users; Lighthouse accessibility (weight 10 audit)
- **Problem:** Lighthouse `button-name` scores 0.00 with weight 10 — buttons on the page have no accessible name.
- **Solution:**
  - Give every `<button>` visible text or an `aria-label`:

```
<button type="button" aria-label="Ava menüü"><svg aria-
hidden="true">…</svg></button>
```

- Mark purely decorative icons `aria-hidden="true"` so they don't pollute the name.

1D. Enlarge touch targets `Accessibility`

- **Impact:** Mobile usability, WCAG 2.5.8
- **Problem:** Lighthouse `target-size` scores 0.00 (weight 7) — interactive elements are smaller than 24×24 CSS px with insufficient spacing.
- **Solution:**
  - Ensure every link/button is at least 44×44 px (24×24 px minimum per WCAG 2.5.8) including padding.
  - Add spacing between adjacent icon links in the header/footer nav so targets don't overlap.

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

- **Impact:** XSS defense-in-depth
- **Problem:** The CSP is only `frame-ancestors 'self'` — no `default-src`, no `object-src 'none'`, so no script or resource restriction exists, and the site exposes an upload surface.
- **Solution:** Deploy a nonce-based policy rather than a flat allowlist:

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

  - Roll out in `Content-Security-Policy-Report-Only` first and fix violations before enforcing.

1F. Load the hero image eagerly `Performance`

- **Impact:** LCP
- **Problem:** The first raster image (`uploads/2026/08/6c041c12…png`, 251×134) is served with `loading="lazy"` and no `fetchpriority`, which delays the LCP candidate.
- **Solution:**

```
<img src="…6c041c12….png" width="251" height="134" alt="…"
     loading="eager" fetchpriority="high" decoding="async">
```

  Keep `loading="lazy"` only on images genuinely below the fold.

1G. Stop returning HTTP 429 to legitimate requests `Best Practices`

- **Impact:** Crawlability, auditability, user access

- **Problem:** The audited URL returned status 429 with `retry-after: 10` and a 337-byte "429 Too Many Requests" body; the W3C validator was also rate-limited (HTTP 429).
- **Solution:**
  - Raise the rate-limit threshold and burst allowance for verified crawlers (Googlebot, PageSpeed Insights, W3C validator) and for normal browsing.
  - Allowlist known-good user agents / IP ranges at the WAF or ZoneOS layer instead of throttling them.
  - Return 429 only for genuinely abusive patterns, and make sure the error page is a real, styled HTML document with `lang`, viewport and meta description.
  - Monitor 429 rates per path so a single hot page cannot lock out all visitors.

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Serve images in WebP/AVIF at the right size `Performance`

- **Impact:** Page weight, LCP, Speed Index
- **Problem:** Images account for 2.66 MB of the 2.72 MB page; 6 of 7 raster images are still JPEG/PNG and PSI's image-delivery insight estimates 231 KiB of savings.
- **Solution:**
  - Convert the remaining JPEG/PNG uploads to WebP/AVIF (the hero is already WebP — extend that to the rest).
  - Keep the existing `srcset` / `sizes` and explicit `width` / `height`.
  - Re-check the 28 distinct srcset widths — trim to 4–5 meaningful breakpoints to reduce generation and cache churn.

2B. Add a meta description `SEO`

- **Impact:** Search snippet CTR
- **Problem:** The document has no meta description (`meta-description` audit fails) and SEO scores 92 on both mobile and desktop.
- **Solution:** Add a unique 120–155 character description per language version:

```
<meta name="description" content="ESTPLY toodab kvaliteetset
kasevineeri Jõgevamaal – kaasaegne vineeritehas, mis toetab Sinu
äri.">
```

2C. Add structured data `SEO`

- **Impact:** Rich results eligibility

- **Problem:** The single JSON-LD block parses but its `@type` is unrecognised, and Lighthouse's `structured-data` audit fails.
- **Solution:**
  - Replace the unknown block with a valid `Organization` + `WebSite` graph, and add `Product` / `BreadcrumbList` on product pages.
  - Validate with the Rich Results Test after deployment.

2D. Add Referrer-Policy, Permissions-Policy, COOP and CORP `Security`

- **Impact:** Information leakage, cross-origin isolation
- **Problem:** Referrer-Policy, Permissions-Policy, Cross-Origin-Opener-Policy and Cross-Origin-Resource-Policy are all missing from the response.
- **Solution:**

```
Header always set Referrer-Policy "strict-origin-when-cross-origin"
Header always set Permissions-Policy "geolocation=(), camera=(),
microphone=()"
Header always set Cross-Origin-Opener-Policy "same-origin"
Header always set Cross-Origin-Resource-Policy "same-origin"
```

2E. Enable HSTS preload `Security`

- **Impact:** First-visit downgrade protection
- **Problem:** HSTS is set with `max-age=31536000; includeSubDomains` but lacks the `preload` directive.
- **Solution:**
  - Add `preload` to the header: `Strict-Transport-Security: max-age=31536000; includeSubDomains; preload`.
  - Submit the domain at hstspreload.org (this is effectively irreversible — confirm all subdomains are HTTPS-only first).

2F. Add Subresource Integrity to third-party scripts `Security`

- **Impact:** Supply-chain tampering
- **Problem:** The one cross-origin script (`cdn.jsdelivr.net/.../iframemanager.js`) has no `integrity` attribute.
- **Solution:**

```
<script
src="https://cdn.jsdelivr.net/gh/orestbida/iframemanager@1.3.0/dist
/iframemanager.js"
        integrity="sha384-…" crossorigin="anonymous" defer>
</script>
```

    - Generate the hash from the pinned 1.3.0 file and update it whenever the version changes.

2G. Make link text descriptive `Accessibility`

- **Impact:** Screen-reader navigation, anchor-text relevance
- **Problem:** Repeated vague anchor text appears across the page ("meist" ×2, "tooted" ×2, "kontakt" ×2, "liitu meiega" ×2, "ressursid" ×2), which is ambiguous out of context.
- **Solution:**
    - Rewrite anchors to describe the destination, e.g. "Meist" → "ESTPLY ettevõttest".
    - Where the visible text must stay short, add `aria-label` with the fuller description.

2H. Cache the HTML document `Performance`

- **Impact:** TTFB, repeat-visit load time
- **Problem:** The document response carries no `cache-control` header, so caching behaviour is unpredictable; the checklist also fails page caching / CDN.
- **Solution:**
    - Send an explicit policy for HTML, e.g. `Cache-Control: public, max-age=600, stale-while-revalidate=60` behind a CDN, or `no-cache` with a strong `ETag` if the page is personalised.
    - Re-verify on the real policy page — the captured response was a 429 error document.

2I. Verify color contrast manually `Accessibility`

- **Impact:** WCAG 1.4.3 contrast
- **Problem:** axe-core returned 0 violations but flagged 8 nodes as incomplete for color contrast, so contrast is unverified.
- **Solution:**
    - Check the 8 flagged nodes in DevTools with the contrast picker; target ≥4.5:1 for body text and ≥3:1 for large text.
    - Pay attention to text over gradients or semi-transparent backgrounds, which axe cannot evaluate.

2J. Protect the WordPress login and installer `Security`

- **Impact:** Brute-force and credential-stuffing exposure

- **Problem:** The login form is served on the default `/wp-login.php` path (warn), which is the first URL every automated credential-stuffing bot tries.
- **Solution:**
  - Move or mask the login path (a plugin or an nginx/Apache rewrite), or put it behind HTTP basic auth / IP allowlisting.
  - Enforce rate limiting and 2FA on `/wp-login.php`.
  - Keep the existing good results: `xmlrpc.php` is blocked, user enumeration is blocked, and no installer or config backup is exposed.

2K. Set the html lang attribute `Accessibility`

- **Impact:** WCAG 3.1.1, screen-reader pronunciation
- **Problem:** axe-core reports a serious `html-has-lang` violation and the captured document has `Lang: not set` (charset windows-1252).
- **Solution:**
  - Declare the document language and a UTF-8 charset:

```
<html lang="et">
<head><meta charset="utf-8"></head>
```

  - Confirm on the real policy page, since the capture was the 429 error document.

Priority 3: Best Practice

Recommended for long-term maintainability.

3A. Hide server version headers `Security`

- **Impact:** Attack-surface reconnaissance
- **Problem:** The response discloses `Server: Apache / ZoneOS`, which the security basics check flags as a warning.
- **Solution:**

```
ServerTokens Prod
ServerSignature Off
```

  - Also strip any `X-Powered-By` header at the origin or edge.

3B. Add main landmark and landmark regions `Accessibility`

- **Impact:** WCAG 1.3.1, keyboard/screen-reader navigation
- **Problem:** axe-core reports `landmark-one-main` and `region` violations; the captured document has no nav, main, header or footer landmarks and no skip link.

- **Solution:**
  - Wrap primary content in `<main>` and navigation in `<nav>`.
  - Add a skip link as the first focusable element:

```
<a class="skip-link" href="#main">Otse sisu juurde</a>
```