Audit Report: Avaleht - estply

Website: https://estply.com/
Date: 09.11.2026
Audit Coverage: □ 88% — W3C Validator: validator.w3.org returned HTTP 429
Confidence: medium

Pages Audited (5 of 5):

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

Summary of results

Overall Score: 93 / 100
Status: □ **Excellent**

What is pulling the score down

Points are for the weakest page; Pages = how many of 5 pages share the signal.

Signal	Points	Pages	Evidence
Accessibility 91	−2.7	5/5	weighted 30% of 9 points lost
SEO 85	−2.3	5/5	weighted 15% of 15 points lost
Field TTFB	−2	5/5	CrUX mobile TTFB p75 914 ms (desktop 709 ms)
Advanced security headers	−1	5/5	Failing: CSP is strict, Referrer-Policy, Permissions-Policy, Cross-Origin-Opener-Policy, Cross-Origin-Resource-Policy
Page score (weakest: https://estply.com/en/privacy-policy/)	**92**		100 − 8.0 → 92

Site overall 93 is the mean of 5 pages; scores range 92 (https://estply.com/en/privacy-policy/) → 93 (https://estply.com/kontakt/). Confidence is medium: W3C Validator: validator.w3.org returned HTTP 429.

Per-page scores

 Excellent · https://estply.com/

Base	Score	Performance	Accessibility	Best Practices	SEO	Security
96	93	100	91	100	92	PASS

 Excellent · https://estply.com/en/privacy-policy/

Base	Score	Performance	Accessibility	Best Practices	SEO	Security
95	92	100	91	100	85	PASS

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

Base	Score	Performance	Accessibility	Best Practices	SEO	Security
96	93	100	91	100	92	PASS

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

Base	Score	Performance	Accessibility	Best Practices	SEO	Security
95	92	98	91	100	92	PASS

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

Base	Score	Performance	Accessibility	Best Practices	SEO	Security
96	93	98	92	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/	100 / 100	**1.05 s** / 315 ms	0.000 / 0.000

URL	Performance (M / D)	LCP (M / D)	CLS (M / D)
https://estply.com/en/privacy-policy/	100 / 100	**1.05 s** / 281 ms	0.000 / 0.000
https://estply.com/privaatsuspoliitika/	100 / 100	**1.06 s** / 281 ms	0.000 / 0.000
https://estply.com/kupsiste-poliitika/	**98** / 100	**1.06 s** / 282 ms	0.000 / 0.000
https://estply.com/kontakt/	**98** / 100	**1.06 s** / 281 ms	0.000 / 0.000

Optimization Checklist

4 of 5 passing — 4 pass · 0 warn · 1 fail · 3 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	**Pass**	All non-hero raster images use loading="lazy".
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.

Item	Status	Detail
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. Reduce server response time (TTFB) `Performance`

- **Impact:** TTFB, FCP, LCP
- **Problem:** CrUX mobile TTFB p75 is 914 ms (rated 'average') and desktop 709 ms, while the Lighthouse lab TTFB is only 3 ms — the gap points at origin/cache-miss latency for real users, not at the HTML itself.
- **Solution:**
  - Confirm WP Rocket page cache is actually served for first-time visitors (check `X-Cache` / `Cache-Control` on a cold request; the response currently returns `cache-control: max-age=0`).
  - Warm the cache after every deploy and purge, and enable a full-page CDN in front of the origin.
  - Check PHP/opcache and database latency on cache misses; the document is only 87 KB decoded, so origin work dominates.

1B. Add accessible names to buttons `Accessibility`

- **Impact:** WCAG 4.1.2, screen-reader navigation
- **Problem:** Lighthouse `button-name` scores 0.00 with weight 10 on mobile — buttons have no accessible name, which is the main driver of the mobile Accessibility score of 91 vs 96 on desktop.
- **Solution:**
  - Give every icon-only button visible text or an `aria-label`:

```
<button type="button" aria-label="Open menu">…</button>
```

    - For buttons that only contain an `<svg>`, add `aria-hidden="true"` to the icon and a label on the button.

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

- **Impact:** FCP, Speed Index, LCP
- **Problem:** `render-blocking-insight` estimates 570 ms of savings and `unused-css-rules` 33 KiB; Speed Index is 3.83 s on mobile against a 1.06 s FCP.
- **Solution:**
    - Inline the critical above-the-fold CSS and load `global.6a324cad….css` (34.3 KB) asynchronously:

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

    - Purge unused selectors from the theme stylesheet (WP Rocket → Remove Unused CSS).
    - Flatten the critical request chain flagged by `network-dependency-tree-insight` (preconnect/preload the jsdelivr script).

1D. Load the hero image eagerly `Performance`

- **Impact:** LCP
- **Problem:** The first/largest image (uploads/2026/08/6c041c12…png, 251×134) carries loading="lazy" and no fetchpriority, delaying the LCP candidate.
- **Solution:**
    - Remove `loading="lazy"` from the hero image (or set `loading="eager"`).
    - Add `fetchpriority="high"`:

```
<img src="…png" width="251" height="134" alt="…"
fetchpriority="high" decoding="async">
```

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

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, mobile data cost
- **Problem:** 8 image requests total 2.66 MB of the 2.72 MB page; PSI's `image-delivery-insight` estimates 231 KiB of savings even though all images already

use srcset and WebP.

- **Solution:**
  - Audit the largest uploads and cap the served width to the rendered slot (28 distinct srcset widths exist — trim the ladder).
  - Re-encode the remaining JPEG/PNG uploads to WebP/AVIF and set `quality` around 75–80.
  - Keep `fetchpriority="high"` on the hero and `loading="lazy"` on everything below the fold.

2B. Enlarge touch targets `Accessibility`

- **Impact:** WCAG 2.5.8, mobile usability
- **Problem:** Lighthouse `target-size` scores 0.00 (weight 7) on mobile — interactive elements are smaller than the 24×24 px minimum with insufficient spacing.
- **Solution:**
  - Ensure at least 44×44 px hit areas for nav links, social icons and footer links:

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

  - Add spacing between adjacent targets so their hit areas do not overlap.

2C. Add a meta description `SEO`

- **Impact:** Search snippet CTR
- **Problem:** The document has no `<meta name="description">`; Lighthouse `meta-description` fails and SEO is 92 on both mobile and desktop.
- **Solution:** Add a unique, ~150-character description per page, e.g.:

```
<meta name="description" content="ESTPLY toodab kvaliteetset kasevineeri Jõgevamaa kaasaegses tehases – usaldusväärne Eesti vineer tootjatele ja edasimüüjatele.">
```

2D. Add structured data `SEO`

- **Impact:** Rich results eligibility
- **Problem:** Lighthouse `structuredData` fails; the single JSON-LD block parses as type 'Unknown', so no Organization/Product markup is recognised.
- **Solution:**
  - Emit a valid `Organization` (and `WebSite`) graph with `name`, `url`, `logo`, `sameAs`.

    - Add `Product` / `Offer` markup on product pages and `BreadcrumbList` where applicable.
    - Validate with the Rich Results Test after deployment.

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

- **Impact:** XSS mitigation
- **Problem:** The CSP is only `frame-ancestors 'self'` — `default-src` is missing and `object-src` is not `'none'`, so it restricts nothing about scripts or resources. The only user-content signal is an anchor whose href contains 'upload', so treat this as P2 rather than P1.
- **Solution:**
    - Start in report-only mode, then enforce:

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

    - Prefer nonce + `'strict-dynamic'` over a flat host allowlist.
    - Allow `cdn.jsdelivr.net` only if the iframemanager script stays.

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

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

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

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

- **Impact:** Supply-chain tampering risk
- **Problem:** The single cross-origin script (`cdn.jsdelivr.net/.../iframemanager@1.3.0`) has no `integrity` attribute.
- **Solution:**
    - Pin the version and add a hash:

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

- Better still, self-host the 2.7 KB file and drop the third-party origin.

2H. Make link text descriptive `SEO`

- **Impact:** Link purpose (WCAG 2.4.4), crawl context
- **Problem:** Lighthouse `link-text` fails with 2 links, and the inventory finds vague repeated text such as "here" ×2.
- **Solution:**
  - Replace "here" with the destination, e.g. "read our cookie policy".
  - Where the visible text must stay short, add `aria-label` with the full purpose.

Priority 3: Best Practice

Recommended for long-term maintainability.

3A. Enable HSTS preload `Security`

- **Impact:** First-visit downgrade protection
- **Problem:** HSTS is set to `max-age=31536000; includeSubDomains` but lacks the `preload` directive, so the very first request can still be downgraded.
- **Solution:**
  - Append `preload` to the header: `Strict-Transport-Security: max-age=31536000; includeSubDomains; preload`.
  - Submit the domain at hstspreload.org once every subdomain is HTTPS-only.

3B. Hide server version headers `Security`

- **Impact:** Attack-surface reconnaissance
- **Problem:** The response discloses `Server: Apache / ZoneOS`, which the security scan flags as technology disclosure.
- **Solution:**

```
ServerTokens Prod
ServerSignature Off
Header unset X-Powered-By
```

3C. 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, which invites automated brute-force attempts.
- **Solution:**
  - Put `/wp-login.php` behind a custom path or HTTP basic auth at the edge.
  - Add rate limiting / fail2ban and enforce strong passwords plus 2FA.
  - Keep `xmlrpc.php` blocked (already passing) and continue blocking user enumeration.

3D. Verify color contrast manually `Accessibility`

- **Impact:** WCAG 1.4.3 readability
- **Problem:** axe-core returned 0 violations but flagged 9 nodes as 'incomplete' for color contrast (gradients/alpha backgrounds it cannot compute).
- **Solution:**
  - Check each flagged element with DevTools' contrast picker against the actual rendered background.
  - Target ≥4.5:1 for body text and ≥3:1 for large text; adjust the palette rather than adding text shadows.

3E. Cache the HTML document `Performance`

- **Impact:** Repeat-visit TTFB
- **Problem:** The document is served with `cache-control: max-age=0` and no ETag, so every repeat visit revalidates against the origin — consistent with the 914 ms field TTFB.
- **Solution:**
  - Let WP Rocket serve a short shared-cache TTL at the edge, e.g. `Cache-Control: public, max-age=600, s-maxage=86400, stale-while-revalidate=60`.
  - Add an `ETag` so conditional requests can return 304 instead of a full 87 KB body.

3F. Add descriptive alt text to content images `Accessibility`

- **Impact:** WCAG 1.1.1 non-text content
- **Problem:** All 22 images technically have an `alt` attribute, but the listed team/photo thumbnails use empty `alt=""` — correct only if they are purely decorative.
- **Solution:**

    - Confirm each empty `alt` is decorative; if the photo conveys a person or product, add a short description.
    - Keep `alt=""` for genuinely decorative graphics so screen readers skip them.

3G. Fix the unused preload credentials mismatch `Best Practices`

- **Impact:** Wasted preload, console noise
- **Problem:** The browser warns that the preload for `rings.svg` is unused because the request credentials mode does not match.
- **Solution:**
    - Add `crossorigin` to the preload link so it matches the consuming request:

```
<link rel="preload" href=".../rings.svg" as="image"
type="image/svg+xml" crossorigin>
```

    - Or remove the preload if the asset is not on the critical path.