Audit Report: Vineeri kvaliteediklassid selgeks - estply

Website: https://estply.com/en/2026/07/06/vineeri-kvaliteediklassid-selgeks/
Date: 29.10.2026
Audit Coverage: 100% — all sources returned data
Confidence: medium

Pages Audited (1 of 1):

- https://estply.com/en/2026/07/06/vineeri-kvaliteediklassid-selgeks/

Summary of results

Overall Score: 73 / 100
Status: Needs Improvement

Performance is split: mobile PSI 78 vs desktop 93, with excellent LCP (2.0 s), TBT (0 ms) and TTFB (3 ms), but a heavy CLS of 0.482 on mobile (desktop 0.143) traced almost entirely to the footer element shifting 0.481 — that single metric is the largest score drag. Accessibility sits at 91 mobile / 96 desktop: axe reports only one moderate violation (missing h1), but Lighthouse independently fails `button-name` (weight 10) and `target-size` (weight 7), and the HTML inventory confirms 0 h1 elements. Image assets are light (2 raster images, both with dimensions and alt attributes) but the hero is lazy-loaded, delaying LCP, and the document is served with `cache-control: max-age=0` despite WP Rocket being present. Security basics PASS (HSTS, nosniff, framing, TLS, SPF/DMARC all good), though CSP is only `frame-ancestors 'self'`, HSTS lacks preload, and Referrer-Policy/Permissions-Policy/COOP/CORP are absent — with no auth, payments, UGC or commerce signals these are P3 per the rubric. W3C reports 8 errors, mostly WP Rocket `text/rocketlazyloadscript` attributes plus two `srcset` candidates missing width descriptors. Confidence is medium because the CrUX field block returns 0 ms p75 for every metric while labelling them 'slow', which is internally inconsistent and unusable.

Per-page scores

 Needs Improvement · https://estply.com/en/2026/07/06/vineeri-kvaliteediklassid-selgeks/

Score	Performance	Accessibility	Best Practices	SEO	Security
73	78	91	100	100	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	**78** / 93	**1.95 s** / 745 ms	**0.482** / 0.143

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. Eliminate the footer layout shift driving CLS 0.482 `Performance`

- **Impact:** CLS, mobile Performance score
- **Problem:** Mobile CLS is 0.482 (desktop 0.143), with `div.main__footer > footer.footer` responsible for a 0.481 shift — far above the 0.25 heavy-penalty threshold.
- **Solution:**
  - Reserve space for the footer before it renders: give `footer.footer` a `min-height` matching its rendered height, or render it server-side instead of injecting it after load.
  - If the shift comes from a late-loading cookie banner or lazyloaded block above the footer, load it with `position: fixed` or reserve a placeholder of identical height.
  - Add `contain: layout` on the footer wrapper and preload the footer's font/logo so its box does not resize.
  - Re-test with Lighthouse mobile until CLS < 0.1.

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Load the hero image eagerly with high fetch priority `Performance`

- **Impact:** LCP, Speed Index
- **Problem:** The optimized-web checklist fails 'Hero image eagerly loaded': `6c041c12…836d595a.png` has `loading="lazy"` and no `fetchpriority`, delaying the LCP candidate.

- **Solution:**

```
<img src="/wp-
content/uploads/2026/08/6c041c12fb0c531933532c08c2749a1a836d595a.pn
g"
     alt="…" width="251" height="134"
     loading="eager" fetchpriority="high" decoding="async">
```

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

2B. Stop serving the HTML document with max-age=0 `Performance`

- **Impact:** Repeat-visit TTFB, caching
- **Problem:** The document response returns `cache-control: max-age=0` with an `expires` timestamp equal to the request time, so browsers revalidate the HTML on every navigation despite WP Rocket being active.
- **Solution:**
  - In WP Rocket (or the origin), enable HTML/browser caching so the document is served with a short but non-zero TTL, e.g. `Cache-Control: public, max-age=600, s-maxage=3600`.
  - Keep `Vary: Accept-Encoding` and purge the cache on publish.
  - Verify with `curl -I` that the canonical URL returns the new header.

2C. Give buttons accessible names and enlarge touch targets `Accessibility`

- **Impact:** WCAG 4.1.2, 2.5.8; Lighthouse Accessibility
- **Problem:** Lighthouse mobile fails `button-name` (score 0.00, weight 10) and `target-size` (score 0.00, weight 7), pulling mobile Accessibility to 91.
- **Solution:**
  - Add visible text or `aria-label` to every `<button>`, e.g. `<button aria-label="Close cookie banner">`.
  - Ensure interactive targets are at least 24×24 CSS px (44×44 recommended) with adequate spacing:

```
.icon-button { min-width: 44px; min-height: 44px; }
```

  - Re-run axe and Lighthouse to confirm both audits pass.

2D. Add a single level-one heading `Accessibility`

- **Impact:** WCAG 1.3.1 / 2.4.6, document outline, SEO
- **Problem:** axe reports `page-has-heading-one` (moderate) and the HTML inventory counts h1 ×0 through h6 ×0 — the page has no heading structure at all.

- **Solution:**
  - Wrap the article title in exactly one `<h1>`: `<h1>Vineeri kvaliteediklassid selgeks</h1>`.
  - Promote section titles to `<h2>` / `<h3>` in document order so the outline is meaningful.
  - Do not use styled `<div>`s in place of headings.

Priority 3: Best Practice

Recommended for long-term maintainability.

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

- **Impact:** XSS defense-in-depth
- **Problem:** CSP is present but only `frame-ancestors 'self'` — no `default-src`, no `object-src 'none'`, and no script/resource restrictions. Site signals show no auth, payments, user content or commerce, so this is a P3 hardening item rather than an active exposure.
- **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:;
  frame-ancestors 'self'; upgrade-insecure-requests
```

  Roll out in report-only mode first, then enforce.

3B. Add missing low-cost security headers and HSTS preload `Security`

- **Impact:** Referrer leakage, feature abuse, origin isolation
- **Problem:** Referrer-Policy, Permissions-Policy, Cross-Origin-Opener-Policy and Cross-Origin-Resource-Policy are all missing, and HSTS (`max-age=31536000; includeSubDomains`) lacks the `preload` directive.
- **Solution:**

```
Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains; preload"
Header always set Referrer-Policy "strict-origin-when-cross-origin"
Header always set Permissions-Policy "camera=(), microphone=(),
geolocation=()"
Header always set Cross-Origin-Opener-Policy "same-origin"
Header always set Cross-Origin-Resource-Policy "same-site"
```

  Submit the domain at hstspreload.org once the header is live.

3C. Fix W3C srcset errors and WP Rocket script type attributes `Best Practices`

- **Impact:** HTML validity, responsive image selection
- **Problem:** W3C reports 8 errors: two `srcset` candidates lack width descriptors (lines 399 and 419), and six `text/rocketlazyloadscript` data blocks carry `defer` / `async` attributes that are invalid for non-JavaScript types.
- **Solution:**
  - Add width descriptors to the srcset candidates:

```
<img srcset="/…/6c041c12….png 251w" sizes="(max-width: 600px)
100vw, 251px" …>
```

  - For the WP Rocket lazyload artifacts, either let the plugin rewrite them to real `<script type="text/javascript">` tags before output, or exclude those specific scripts from lazyload so the emitted markup is valid.

3D. Add SRI to the jsDelivr script and harden WordPress entry points `Security`

- **Impact:** Supply-chain integrity, admin attack surface
- **Problem:** The single cross-origin script (`cdn.jsdelivr.net/…/iframemanager.js`) has no `integrity` attribute, `/wp-admin/install.php` is reachable, and the login form is served on the default `/wp-login.php` path.
- **Solution:**
  - Add SRI plus CORS to the CDN tag:

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

  - Block the installer in `.htaccess` : `RewriteRule ^wp-admin/install\.php$ - [F,L]` .
  - Restrict or rename the login path and add rate limiting / 2FA.