Audit Report: Privacy Policy - estply

Website: https://estply.com/en/privacy-policy/
Date: 29.10.2026
Audit Coverage: 100% — all sources returned data
Confidence: high

Pages Audited (1 of 1):

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

Summary of results

Overall Score: 83 / 100
Status: **Needs Improvement**

Performance is strong — PSI mobile 92 / desktop 98, CLS 0.000, TBT 0 ms, TTFB 3 ms — but mobile LCP is 3.3 s versus 0.7 s on desktop, a >1.5 s divergence driven by a hero image served with loading="lazy" and 720 ms of render-blocking resources plus 32 KiB unused CSS. Accessibility splits between tools: axe-core reports 0 violations across 40 passes, yet Lighthouse mobile scores 91 with `button-name` (weight 10, score 0) and `target-size` (weight 7, score 0) failing outright, and axe left 7 color-contrast nodes undetermined. SEO sits at 85 on both strategies due to a missing meta description, non-descriptive "here" links, and absent structured data. Security basics verdict is PASS, but the CSP is only `frame-ancestors 'self'` with no default-src or object-src, and Referrer-Policy, Permissions-Policy, COOP and CORP are all missing — low-risk here since the page has no auth, payments, or user content. W3C reports 8 errors, mostly WP Rocket `text/rocketlazyloadscript` data blocks carrying defer/async plus two srcset candidates missing width descriptors.

Per-page scores

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

Score	Performance	Accessibility	Best Practices	SEO	Security
83	92	91	100	85	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	**92** / 98	**3.30 s** / 721 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.
Reasonable number of image sizes	N/A	Too few raster images to evaluate srcset width variety.

Item	Status	Detail
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. Load the hero image eagerly with high fetch priority `Performance`

- **Impact:** LCP (mobile 3.3 s vs desktop 0.7 s)
- **Problem:** The optimized-web checklist flags the hero image (`uploads/2026/08/6c041c12fb0c531933532c08c2749a1a836d595a.png`) as `loading="lazy"` with no `fetchpriority`, and mobile LCP is 3.3 s against a 2.5 s good threshold.
- **Solution:**
  - Remove `loading="lazy"` from the hero/LCP image (or set `loading="eager"`).
  - Add `fetchpriority="high"` and a `preload` hint:

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

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

1B. Give every button an accessible name `Accessibility`

- **Impact:** WCAG 4.1.2 Name, Role, Value; Lighthouse accessibility score
- **Problem:** Lighthouse reports `button-name` at score 0.00 with weight 10 — "Buttons do not have an accessible name" — even though axe-core returned 0 violations, so the failure is real but tool-dependent (likely cookie-banner or icon buttons).
- **Solution:**
  - Audit every `<button>` in the rendered DOM (including the cookie consent banner injected by CookieYes).
  - Add visible text or an `aria-label`:

```
<button type="button" aria-label="Close cookie banner">
  <svg aria-hidden="true">…</svg>
</button>
```

- Re-run Lighthouse after the consent script has executed to confirm the fix.

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Eliminate render-blocking CSS and unused CSS `Performance`

- **Impact:** FCP, LCP, Speed Index (mobile 2.94 s)
- **Problem:** PSI flags `render-blocking-insight` with an estimated 720 ms saving and `unused-css-rules` with 32 KiB of unused CSS; the page loads 2 external stylesheets plus 3 inline blocks (9.6 KB).
- **Solution:**
  - Inline critical above-the-fold CSS and defer the rest:

```
<link rel="preload" href="/wp-content/themes/.../style.css"
as="style" onload="this.onload=null;this.rel='stylesheet'">
<noscript><link rel="stylesheet" href="/wp-
content/themes/.../style.css"></noscript>
```

  - Enable WP Rocket's "Remove Unused CSS" (Optimize CSS delivery) feature.
  - Split per-template stylesheets so the privacy-policy page does not load theme CSS it never uses.

2B. Add a meta description `SEO`

- **Impact:** SEO score (85), SERP snippet quality
- **Problem:** The document has no meta description (`meta-description` audit score 0.00) while title, canonical, hreflang and Open Graph tags are all present.
- **Solution:** Add a unique 120–155 character description per language variant:

```
<meta name="description" content="How Estply collects, uses and
protects personal data submitted through our website and contact
forms.">
```

  Set it in the SEO plugin so the `et` and `x-default` variants get their own translated copy.

2C. Enlarge touch targets to at least 24×24 px (44×44 recommended) `Accessibility`

- **Impact:** WCAG 2.5.8 Target Size; Lighthouse accessibility score
- **Problem:** Lighthouse `target-size` scores 0.00 with weight 7 — "Touch targets do not have sufficient size or spacing" — and axe could not verify 7 color-contrast nodes.
- **Solution:**
  - Increase padding/min-height on footer and inline links to at least 24×24 CSS px, ideally 44×44:

```
.footer a, .legal-links a { display:inline-block; min-height:44px;
padding:10px 8px; }
```

  - Manually verify the 7 undetermined color-contrast nodes (gradients/alpha) with a contrast checker.

2D. Fix invalid srcset candidates missing width descriptors `Best Practices`

- **Impact:** W3C validation (2 of 8 errors), responsive image selection
- **Problem:** W3C reports "No width specified for image …" for two `srcset` values on the images at lines 536 and 556, where a `sizes` attribute is present.
- **Solution:** Every candidate in a `srcset` used with `sizes` must carry a width descriptor:

```
<img src="…-251.png"
     srcset="…-251.png 251w, …-502.png 502w"
     sizes="(max-width: 600px) 100vw, 251px"
     width="251" height="134" alt="">
```

Priority 3: Best Practice

Recommended for long-term maintainability.

3A. Remove defer/async from WP Rocket lazyload data blocks `Best Practices`

- **Impact:** W3C validation (6 of 8 errors)
- **Problem:** W3C reports 5 errors for `<script type="text/rocketlazyloadscript" … defer>` and 1 for the same pattern with `async` — a data block must not carry `defer` / `async`.
- **Solution:**
  - WP Rocket rewrites `src` to `data-rocket-src` and changes `type` to `text/rocketlazyloadscript`; the `defer` / `async` attributes are then invalid on a non-JS MIME type.

- Either let WP Rocket manage the delay (drop the manual `defer` / `async` attributes from the original tags) or exclude those specific scripts from Delay JS execution in WP Rocket settings.
- Also drop the redundant `type="text/javascript"` on the inline script at line 384 (2 warnings).

3B. Strengthen 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, or user content, so this is a low-urgency hardening item rather than an exposure.
- **Solution:**
  - Add at minimum:

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

  - If you later add a login or user content, move to a nonce + `'strict-dynamic'` policy rather than a flat allowlist.

3C. Add Referrer-Policy, Permissions-Policy, COOP and CORP `Security`

- **Impact:** Privacy leakage, 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:**

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

3D. Add HSTS preload and SRI on the CDN script `Security`

- **Impact:** Transport hardening, third-party script integrity
- **Problem:** HSTS is `max-age=31536000; includeSubDomains` but lacks `preload`, and the single cross-origin script from cdn.jsdelivr.net has no `integrity` attribute.
- **Solution:**
  - Append `; preload` to the HSTS header and submit the domain at hstspreload.org.

- Add SRI to the iframemanager script:

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

3E. Harden WordPress entry points `Security`

- **Impact:** Attack surface
- **Problem:** Security basics reports `/wp-admin/install.php` reachable and the login form served on the default `/wp-login.php` path.
- **Solution:**
  - Block the installer in `.htaccess`:

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

  - Move or protect the login path (e.g. a custom login slug plugin) and add rate limiting / 2FA.

3F. Disambiguate vague link text and add structured data `SEO`

- **Impact:** SEO score (85), link-text audit
- **Problem:** `link-text` scores 0.00 with 2 non-descriptive links, and the inventory shows "here" ×2 plus repeated "about us", "products", "resources"; the single JSON-LD block has an unrecognised type.
- **Solution:**
  - Replace "here" with descriptive anchor text, e.g. `<a href="/en/contact/">contact our data protection officer</a>`.
  - Add `aria-label` where the visible text must stay short.
  - Give the JSON-LD block a recognised `@type` (e.g. `WebPage` / `Organization`) so it is eligible for rich results.

3G. Allow browser caching of the HTML document `Performance`

- **Impact:** Repeat-visit TTFB, back/forward navigation
- **Problem:** The document response sends `cache-control: max-age=0` with an `expires` timestamp equal to the request time, so repeat visitors re-download the HTML even though WP Rocket caches it server-side.
- **Solution:**

- Serve a short but non-zero browser TTL for anonymous HTML, e.g. `Cache-Control: public, max-age=600, s-maxage=3600`.
- Keep `max-age=0` only for logged-in or personalised responses.