

ELEMENTOR PRO PERFORMANCE & PAGE SPEED GUIDE

Technical Blueprint for Achieving 90+ Core Web Vitals & PageSpeed Scores

Target PageSpeed Score:	90+ Mobile / 98+ Desktop	Date Revised:	August 2026
Framework:	Elementor Pro + WordPress Core	Author:	Empirican Technical SEO & Web Performance Team

1. EXECUTIVE SUMMARY & CORE BENCHMARKS

Elementor Pro offers powerful visual page building, but default configurations can generate excessive DOM nodes, unoptimized CSS/JS assets, and render-blocking resources. This guide outlines the exact technical procedures required to consistently reach **90+ Google PageSpeed Insights (PSI)** scores and pass Google Core Web Vitals on Elementor Pro websites.

Core Web Vital Metric	Target Threshold	Primary Cause in Elementor	Optimization Fix
LCP (Largest Contentful Paint)	< 2.5 seconds	Unoptimized Hero Images & Background Videos	Preload Hero assets, convert to WebP/AVIF, omit lazy loading on fold.
INP (Interaction to Next Paint)	< 200 ms	Heavy JS execution, menu triggers, widget scripts	Delay non-critical JS execution, dequeue unused Elementor scripts.
CLS (Cumulative Layout Shift)	< 0.1	Unsize images, dynamic icon font loading, FOIT/FOUT	Explicit height/width attributes, CSS aspect-ratio, local SVG icons.

2. ELEMENTOR CORE SETTINGS & FEATURE FLAGS

In the WordPress Admin dashboard, navigate to **Elementor > Settings > Features** and configure the following native performance switches:

Feature Flag	Recommended Setting	Impact on Page Speed
Flexbox Container / Grid	Active	Reduces DOM depth by up to 50% by eliminating wrapper divs.
Inline Font Icons (eicons)	Active	Renders icons inline as SVG instead of loading bulky font files.
Improved CSS Loading	Active	Loads CSS asynchronously and only for rendered widgets on page.

Feature Flag	Recommended Setting	Impact on Page Speed
Improved Asset Loading	Active	Prevents JS libraries from loading if their corresponding widget isn't used.
Optimized DOM Output	Active	Strips legacy CSS classes and extra section container wrappers.

3. DOM REDUCTION & STRUCTURE BEST PRACTICES

Container & Layout Rules

- **Flatten DOM Hierarchy:** Limit total DOM nodes to under 800 and depth to under 32 elements per page.
- **Use Flexbox Containers instead of Sections/Columns:** Never nest containers deeper than 2 layers. Use direction (row/column) and gap settings rather than empty spacer widgets.
- **Eliminate Unnecessary Widgets:** Combine text headings and subheadings inside a single Text Editor widget or HTML widget when custom markup permits.

4. ASSET & FONT OPTIMIZATION STRATEGY

Font & Icon Optimization

- **Disable Google Fonts API:** Self-host WOFF2 font files locally via WP Rocket or Perfmatters to eliminate external render-blocking roundtrips to `fonts.gstatic.com`.
- **CSS Font Display:** Enforce `font-display: swap;` across all typography styles.
- **Font Awesome:** Disable Font Awesome in Elementor settings if custom SVG icons can be used instead.

Asset Unloading & Script Management

Dequeue unused scripts using a script manager (e.g., Perfmatters or Asset CleanUp). Common unnecessary assets in Elementor pages include:

- `elementor-dialog` and `swiper.js` on pages without sliders or modal popups.
- `wp-embed.min.js` and Gutenberg CSS block styles on custom-built Elementor landing pages.
- WooCommerce scripts (`cart-fragments.js`) on non-ecommerce pages.

5. CACHING, IMAGE COMPRESSION & SERVER STACK

Optimization Layer	Recommended Solution / Stack	Key Configuration
Server Hosting	LiteSpeed / Nginx / Cloudflare Enterprise	HTTP/3 enabled, Redis Object Caching, PHP 8.2+.
Page Caching Plugin	WP Rocket / Litespeed Cache	Critical CSS generation, Delay JS execution, Minify CSS/JS.

Optimization Layer	Recommended Solution / Stack	Key Configuration
Image Optimization	ShortPixel / Imagify / Cloudflare Polish	Convert to WebP/AVIF format, auto-scale down large uploads.
CDN / Edge Caching	Cloudflare APO (Automatic Page Optimization)	Cache full HTML at edge servers to reduce TTFB to < 150ms.

6. VERIFICATION & MONITORING WORKFLOW

Testing Checklist

1. Test using **Google PageSpeed Insights** in Incognito mode to bypass admin-level cookies.
2. Run audits via **WebPageTest.org** (Mobile 4G profile) to analyze waterfall requests and CPU execution time.
3. Monitor Core Web Vitals real-user metric (RUM) data in Google Search Console's CWV report.