Page Speed Improvements

STATUS: **COMPLETED**

Prepared by: Web Performance Team | Date: April 2026 | Version: 1.0

1. Executive Summary

This report documents the successful completion of the Page Speed Improvement initiative for our website. Through a combination of image optimisation, caching strategies, code minification, and server-side tuning, we have significantly improved load times and Core Web Vitals scores across all key pages.

2. Performance Metrics — Before vs After

All measurements were taken using Google Lighthouse (desktop) on the homepage, product listing, and checkout pages.

Metric	Before	After	Improvement
Performance Score	52 / 100	94 / 100	**+42 pts**
First Contentful Paint (FCP)	4.8 s	1.2 s	**-75%**
Largest Contentful Paint (LCP)	7.1 s	2.0 s	**-72%**
Total Blocking Time (TBT)	820 ms	110 ms	**-87%**
Cumulative Layout Shift (CLS)	0.31	0.04	**-87%**
Time to Interactive (TTI)	9.4 s	3.1 s	**-67%**
Page Weight (avg.)	5.6 MB	1.4 MB	**-75%**

Table 1 — Core Web Vitals and Lighthouse scores, desktop, homepage.

3. Optimisations Completed

3.1 Image Optimisation

- Converted all PNG/JPG assets to WebP format — average 68% size reduction.
- Implemented lazy-loading (loading='lazy') on all below-the-fold images.
- Added explicit width and height attributes to eliminate layout shift.
- Deployed a CDN image pipeline with automatic next-gen format serving.

3.2 CSS & JavaScript Minification

- Minified and tree-shook all JavaScript bundles using Rollup; removed 142 KB of unused code.

- Inlined critical CSS (above-the-fold) and deferred non-critical stylesheets.
- Eliminated three third-party scripts that were blocking the main thread.

3.3 Caching & Compression

- Configured long-lived Cache-Control headers (1 year) for all static assets.
- Enabled Brotli compression on the server; GZIP fallback for older clients.
- Implemented a Service Worker for offline caching of shell assets.

3.4 Server & Hosting

- Migrated to HTTP/2 and enabled server push for critical resources.
- Moved static assets to a globally distributed CDN (99th-percentile TTFB < 80 ms).
- Enabled Early Hints (103) on the main document request.

3.5 Font Optimisation

- Replaced three legacy web-fonts with a single variable font file.
- Added font-display: swap to all @font-face declarations.
- Preloaded the primary font file in <head> using <link rel='preload'>.

---

4. Tools & Testing Environment

Tool / Service	Purpose	Result
Google Lighthouse	Lab performance auditing	**Score 94/100**
WebPageTest.org	Real-browser waterfall analysis	**Pass**
Chrome DevTools	Coverage & runtime profiling	**Pass**
GTmetrix	Third-party monitoring	**Grade A**
Squoosh / Sharp	Image compression pipeline	**68% avg saving**
Rollup + Terser	JS bundle optimisation	**142 KB removed**

---

5. Recommendations & Next Steps

- Monitor Core Web Vitals via Google Search Console monthly to catch regressions early.
- Implement Real User Monitoring (RUM) using web-vitals.js to capture field data.
- Evaluate partial pre-rendering for the product listing page (target LCP < 1.5 s).
- Review and optimise third-party tag loading cadence via Tag Manager.
- Schedule quarterly Lighthouse audits after each major release.
