

WordPress Performance Audit Worksheet

A practical system for diagnosing a slow WordPress directory, marketplace, membership site, or WooCommerce store before anyone changes plugins, caches, scripts, or hosting.

Do not optimize a score. Remove delay from the journeys that run the business, and prove those journeys still work.

01

Define the slow journey

02

Rank likely bottlenecks

03

Audit plugin ownership

04

Fix one layer

05

Verify speed and function

PRINTABLE WORKSHEET

Accessible digital version available in the complete guide

1. Define the slow journey

Replace “the site is slow” with reproducible statements. Record three journeys, including the user, page or action, device, cache state, observed delay, and business outcome at risk.

Site / environment	Audit owner	Date

Three slow business journeys

Journey	Conditions and observed problem	Outcome at risk	Priority 1-5
Journey 1			
Journey 2			
Journey 3			

Rule: Priority: 1 = low, 5 = critical. A homepage score is not the answer to a search, dashboard, booking, or checkout problem.

2. Rank likely bottlenecks

S+ is our first audit on mature, plugin-heavy sites. Measurements from the affected business journey still override this default order.

Tier	Bottleneck	First evidence	Best first move
S+	Redundant plugins / overlapping tools	Responsibility and asset map	Keep one owner per job; remove the rest safely
S	Missing or incorrect page cache	Hit/miss headers and exclusions	Verify public hits and protected routes
S	Global JavaScript and third parties	Waterfall, Coverage, long tasks	Remove, then conditionally load
S	Oversized main/listing media	Main image, dimensions, transfer	Resize, variants, compress, prioritize
S	Slow uncached request	Query, hook, API, transaction profile	Fix the measured component
A	Background job or origin pressure	Queues, CPU, memory, workers	Fix ownership before upgrading
A	CSS, fonts, builder output, DOM	Coverage and rendering trace	Remove unused route-level work
B	CDN / edge delivery	Distance and cache headers	Improve delivery, not app logic

Baseline evidence

- | | |
|--|--|
| ☐ Same URL, user, device, location, and cache condition | ☐ Field and lab data recorded separately |
| ☐ Initial HTML time and cache signal captured | ☐ Browser waterfall and long tasks captured |
| ☐ Dynamic request profiled by component | ☐ Business action success or failure recorded |

Order: Remove work, control where remaining work runs, cache safe output, then tune delivery.

3. Audit plugin ownership

Plugin count is not the target. Assign every active plugin or component a business function, scope, measured cost, and decision. Duplicate this page when more rows are needed.

Plugin / component	Business function	Where needed	Measured cost	Decision

Overlap check

- ☐ Cache layers have named scope and purge path
- ☐ Only one CSS/JS rewrite owner
- ☐ Only one lazy-load or image/CDN rewrite owner
- ☐ Analytics and pixels are not duplicated
- ☐ Only one database-cleanup schedule
- ☐ Deactivated plugins left no unexplained jobs or data

Removal gate:

Confirm business and data ownership, disable on staging, clear caches, retest affected journeys, then delete confirmed orphaned data.

4. Fix one layer

Make one reversible change tied to measured evidence. Do not change hosting, caching, plugins, scripts, and images together and then guess what helped.

Measured problem

What evidence connects this layer to the slow journey?

One change

What exactly will change, where, and who owns it?

Scope and exclusions

Which routes, users, cookies, jobs, or integrations must remain outside the change?

Rollback

What result triggers a revert, and how will the prior state be restored?

Acceptance tests

Which business journeys and technical signals must pass before the change stays?

Change rule:

Capture evidence before, make one bounded change, rerun the same test, and keep or revert based on the result.

5. Verify speed and function

Functional acceptance

- ☐ Navigation, search, filters, sorting, and maps work
- ☐ Listings, forms, consent, email, and webhooks work
- ☐ No cross-user leakage appears in tested personalized journeys and cache states
- ☐ Registration, login, reset, and account actions work
- ☐ Booking, cart, checkout, payment, and returns work
- ☐ Admin and scheduled operations remain usable

Performance acceptance

- ☐ Original complaint improved under comparable conditions
- ☐ Dynamic journeys meet documented targets
- ☐ Payload, long tasks, or execution improved where causal
- ☐ Public cacheable pages show verified hits
- ☐ Main content is discovered promptly and not lazy-loaded
- ☐ Origin remains stable under representative load

Change record

Change	Evidence before	Evidence after	Regression result	Keep / revert

Need specialist review?

Request a WordPress performance review before changing another plugin, cache layer, or host.

directorism.com/contact?type=wordpress-performance