GYATA TABIRI

Inspired. Precise. Purposeful.

Improving an existing stylesheet without starting over

By Gyata Tabiri · Revised November 2, 2026

An untidy stylesheet may contain years of solved problems. Some declarations are obsolete; others protect a page nobody remembers to check. I would establish which is which before treating a shorter file as an improvement.

Agree on what should stay the same

Choose representative pages and states before changing shared rules: navigation, tables, forms, unusual content lengths, and narrow screens. Record the appearance and operate the controls. Without a baseline, a visual difference can be mistaken for either an intentional improvement or a harmless consequence.

Choose one source of uncertainty

If button colors have drifted, start there. Identify the roles the values serve and introduce a shared rule where the meaning agrees. A matching color on a warning and a decorative border does not necessarily mean they should become the same token.

Separate cleanup from redesign

A structural refactor and a visual refresh answer different questions. Combining them makes a regression harder to recognize and a review harder to discuss. I prefer a small change whose purpose is clear, followed by a separate visual decision when that is actually wanted.

Remove the old path when the new one is established

Leaving both implementations indefinitely can make the active rule harder to find. Once the replacement is verified, remove the obsolete declaration and keep a short explanation for any exception that remains. If a dependency is still uncertain, leave that uncertainty visible rather than declaring the cleanup complete.

The measure of the work is whether the next change becomes easier to understand. Fewer lines can be a consequence, but they are not enough to establish that the refactor was worthwhile.

Read alongside: Testing an explanation before changing a rule · Preserving the reasons behind a file.