

WordPress analysis

Frontend performance
optimization

© 2019 Codeconut Ltd. – Christian Oellers
www.codeconut.io

This is a Google Slides port
from the original document

Please excuse the typos due to incompatible
PDF engines in graphic software



%Client%

Frontend Performance Review

2019/08/13 - 1.0.0

Table of Contents

1. [About the Review](#)
2. [Onpage Performance Review](#)
3. [Bottlenecks & Challenges](#)
4. [Recommendations](#)
5. [Further Considerations](#)
6. [Terms & Definitions](#)
7. [Resources & Links](#)
8. [Disclaimer](#)

1. About the Review



Testing methods & Results focus

How & What we tested

How?

- Combine **12** analytics tools
- Test Desktop & Mobile homepage

What?

- Focus on frontend performance
(less on server or backend)
- Check basic HTML structures (SEO)



Preferences & Prioritisation

Evaluating solutions

Client preferences

1. Fast realisation of results
2. Low implementation costs
3. Low recurring maintenance cost
4. (Minimal changes to website layout or content needed)

Prioritisation

- Simple & high impact solutions first
- No costly and time intense manual code changes or refactoring
- Custom optimisations only if needed



Scores & Goals

Setting realistic goals &
Measuring results

About scores

- Don't aim for a 100% score just for the perfect number
- Don't correlate a faster page with a higher score^{*2 *3}

Goals

- Set viable speed improvements from the current state; e.g. 25–50% faster initial page load on all devices
- Improve gradually from there, given effort, cost and result are good fits



Reasoning

Prioritisation explained

What to optimise?

- **Frontend:** Offers the largest leverage to optimise big way
- **Backend:** Optimise WordPress + Plugins first (before customising any other code parts or configuration)
- **Server:** Hosting & Server – A shared server offers less speed and options for configuration

Optimisation is a continuous, iterative process.
Track ->Evaluate ->Optimise ->Repeat ...

2. Onpage Performance Review



Facts & Numbers

Average test results

Based on Google PageSpeed &
Google Lighthouse tests

Current status – Homepage

- **6–19 seconds** until content is visible
- **~55 seconds** until page is useable
- **17 MB** total page size with images
- **164+files** loaded at once
- **179+total** requests sent
- **17 unnecessary** 301 redirects

Exemplaric result. Values may vary depending on time, location, testing methodologies and cache settings.

**>50% visitors bounce if
page loads over 3 seconds**

Google benchmark / Mobile user behaviour^{*1}



...but is it true?

Don't trust all statistics

How to find out

- Implement own tracking solution
- Set custom goals to measure
- Find out how your customers 'tick'
(analytics, campaigns, surveys, ...)

Considerations

- Statistics might be too narrow, old or just not fit your users & industry
- Different user behaviour / Stronger internet & faster devices
- Really want the product no matter what (=willing to wait longer for it)



Why optimise?

General benefits

Faster websites...

- Can increase the CR % (your goals)
- Raise user satisfaction (likely more product & brand engagement)
- Impact SEO positively (performance included in ranking algorithms)

Long-term...

- Reduced costs of visiting for users (mobile data), maintenance & editing (staff) and hosting (server, CDN)



Goals

Overarching definitions

Thoughts & Considerations

- What is the desired outcome? (e.g. increased page load speed by x%; reduced file size from x to y MB ...)
- What are we willing to change or `sacrifice` for these goals? (e.g. changing layout; images, animations, ...)

A goodstart ...

1. Achieve 25% faster pageload.
2. Can we optimise or reduce more?
Go further and aim for $\geq 50\%$.



Our goals – 1/2

Current optimisation goals

1st Stage – Basics

1. Minify images(JPG, PNG, GIF, ...)
2. Minify +Combine source (CSS, JS)
3. Use caching(CSS, JS, HTML)
4. Implement image lazy loading

2nd Stage – Additional

1. Reduce & Remove content parts
 - a. Split content into subpages
2. Choose new file formats
 - a. Images =WebP?
 - b. GIF =MPEG4 Video?



Our goals – 2/2

Advanced possible optimisations

Tackle if needed

3rd Stage – Future

1. Choose performance-oriented & minimal WordPress theme
 - a. Less complex HTML +faster CSS
 - b. Requires content +layout changes
2. Change hosting solution to dedicated instead shared server (VPS?)
 - a. Better hardware (RAM, CPU, SSD)
 - b. Customisable config (PHP, MySQL)
 - c. Less load peak times and outages
3. Utilise commercial CDN or caching solutions (e.g. Cloudflare)
4. Get custom tailored plugins?
(e.g. without jQuery)

3. Bottlenecks & Challenges



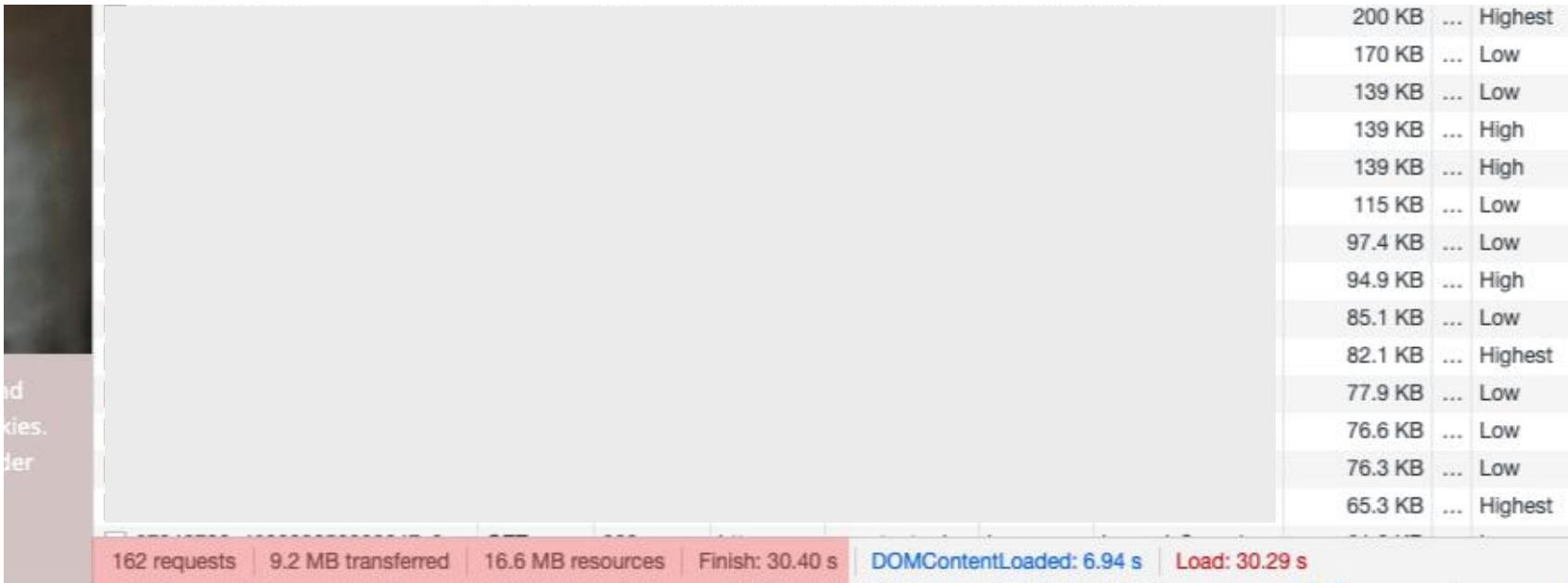
Topchallenges

Problems & Solutions

Top 6

1. Oversized +too many image files
2. No image 'lazy loading' enabled
 - a. All files load at once even if invisible
 - b. Too much assets loaded on 1st page
3. Scattered JS/CSS blocks rendering
 - a. Initial page appears partly empty for a while until the main image loads
4. Too many plugins +Scripts
 - a. e.g. YouTube player
5. No file compression(GZIP) enabled
6. Unnecessary 30x/http redirects
7. Partly slow server (lower priority)

1+2. Toomany initial requests +Large file sizes



1.1.GIF too large – Remove/Change/lazyload

▲ Defer offscreen images

12.15

s

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more.](#)



Install a [lazy-load WordPress plugin](#) that provides the ability to defer any offscreen images, or switch to a theme that provides that functionality. Also consider using the [AMP plugin](#).

URL	Size	Potential Savings
	1,929 KB	1,929 KB

3.1. Inline JS within HTML (plugins/theme)

```
3666      </div>
3667  </div>
3668  <div id="photoblocks-fancybox-2"></div>
3669  <script>
3670    var $ = jQuery;
3671    $(document).ready(function() {
3672      var $ = jQuery;
3673      $(document).ready(function() {
3674        jQuery(function() {
3675          var p = new PhotoBlocks({
3676            selector: "#photoblocks-2",
3677            columns: 1,
3678            padding: 10,
3679            resizer: '',
3680            image_quality: 80,
3681            disable_below: 320,
3682            imageFactor: 1.5,
3683            on: {
3684              before: function() {},
3685              after: function() {},
3686              refresh: function() {}
3687            },
3688            mobile_layout: [],
3689            lazy: false,
3690            debug: false
3691          });
3692        });
3693      });
3694    });
3695  });
3696 </script>
```

HTML

JS

3.2. Inline CSS within HTML (plugins/theme)

3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854

CSS

```
#photoblocks-4 .pb-block.pb-type-text .pb-description {  
}  
  
#photoblocks-4 .pb-overlay {  
    background: rgba(0, 0, 0, 0.75);  
}  
  
#photoblocks-4.pb-lift.show-empty-overlay .pb-block.pb-type-image:hover, #photoblock:  
    box-shadow: rgba(0, 0, 0, 0.75) 0 0 20px;  
}  
  
#photoblocks-4 .pb-block {  
}  
  
#photoblocks-4 .pb-block .pb-social button {  
    color: #ffffff;  
    font-size: 14px;  
}  
}  
</style>  
<div class="photoblocks-gallery pb-blur pb-lift pb-effect-fade" data-anim="fade" id="pho:  
    <div class="pb-blocks">  
        <div class="pb-block pb-landscape pb-width pb-type-image " style="" data-colspan:  
            
11333     <div class="cookie-notice-container">
11334         <span id="cn-notice-text">Um unsere Webseite für dich optimal zu gestalten und fortlaufend verbessern zu können, verl...
11335         <a href="#" id="cn-accept-cookie" data-cookie-set="accept" class="cn-set-cookie cn-button wp-default button">Verstehen</a>
11336         <a href="#" href="#" target="_blank" id="cn-more-info" class="cn-more-info cn-button wp-default button">Weiterlesen</a>
11337     </div>
11338 </div>
11339 <script type="text/javascript" defer src="https://www.rechner.club/wp-content/themes/rechnerclub/assets/js/main.js?ver=1.0.0">
11340 <script data-cfasync='false'>
11341     !function(t) {
11342         "use strict";
11343         t.loadCSS || (t.loadCSS = function() {});
11344     );
11345     var e = loadCSS.relpreload = {};
11346     if (e.support = function() {
11347         var e;
11348         try {
11349             e = t.document.createElement("link").relList.supports("preload")
11350         } catch (t) {
11351             e = !1
11352         }
11353     });
11354     var n = t.document.querySelectorAll("img");
11355     for (var r = 0; r < n.length; r++) {
11356         var i = n[r];
11357         if (i.getAttribute("src")) {
11358             var s = i.getAttribute("src");
11359             if (!e.support) {
11360                 i.setAttribute("src", s + "?ver=" + new Date().getTime());
11361             } else {
11362                 i.setAttribute("srcset", s + "?ver=" + new Date().getTime());
11363             }
11364         }
11365     }
11366     var o = t.querySelectorAll("img");
11367     for (var a = 0; a < o.length; a++) {
11368         var l = o[a];
11369         if (l.getAttribute("src")) {
11370             var c = l.getAttribute("src");
11371             if (!e.support) {
11372                 l.setAttribute("src", c + "?ver=" + new Date().getTime());
11373             } else {
11374                 l.setAttribute("srcset", c + "?ver=" + new Date().getTime());
11375             }
11376         }
11377     }
11378     var d = t.querySelectorAll("img");
11379     for (var f = 0; f < d.length; f++) {
11380         var h = d[f];
11381         if (h.getAttribute("src")) {
11382             var g = h.getAttribute("src");
11383             if (!e.support) {
11384                 h.setAttribute("src", g + "?ver=" + new Date().getTime());
11385             } else {
11386                 h.setAttribute("srcset", g + "?ver=" + new Date().getTime());
11387             }
11388         }
11389     }
11390     var p = t.querySelectorAll("img");
11391     for (var v = 0; v < p.length; v++) {
11392         var b = p[v];
11393         if (b.getAttribute("src")) {
11394             var w = b.getAttribute("src");
11395             if (!e.support) {
11396                 b.setAttribute("src", w + "?ver=" + new Date().getTime());
11397             } else {
11398                 b.setAttribute("srcset", w + "?ver=" + new Date().getTime());
11399             }
11400         }
11401     }
11402     var m = t.querySelectorAll("img");
11403     for (var u = 0; u < m.length; u++) {
11404         var k = m[u];
11405         if (k.getAttribute("src")) {
11406             var j = k.getAttribute("src");
11407             if (!e.support) {
11408                 k.setAttribute("src", j + "?ver=" + new Date().getTime());
11409             } else {
11410                 k.setAttribute("srcset", j + "?ver=" + new Date().getTime());
11411             }
11412         }
11413     }
11414     var y = t.querySelectorAll("img");
11415     for (var x = 0; x < y.length; x++) {
11416         var z = y[x];
11417         if (z.getAttribute("src")) {
11418             var A = z.getAttribute("src");
11419             if (!e.support) {
11420                 z.setAttribute("src", A + "?ver=" + new Date().getTime());
11421             } else {
11422                 z.setAttribute("srcset", A + "?ver=" + new Date().getTime());
11423             }
11424         }
11425     }
11426     var L = t.querySelectorAll("img");
11427     for (var T = 0; T < L.length; T++) {
11428         var S = L[T];
11429         if (S.getAttribute("src")) {
11430             var R = S.getAttribute("src");
11431             if (!e.support) {
11432                 S.setAttribute("src", R + "?ver=" + new Date().getTime());
11433             } else {
11434                 S.setAttribute("srcset", R + "?ver=" + new Date().getTime());
11435             }
11436         }
11437     }
11438     var V = t.querySelectorAll("img");
11439     for (var U = 0; U < V.length; U++) {
11440         var W = V[U];
11441         if (W.getAttribute("src")) {
11442             var X = W.getAttribute("src");
11443             if (!e.support) {
11444                 W.setAttribute("src", X + "?ver=" + new Date().getTime());
11445             } else {
11446                 W.setAttribute("srcset", X + "?ver=" + new Date().getTime());
11447             }
11448         }
11449     }
11450     var Z = t.querySelectorAll("img");
11451     for (var B = 0; B < Z.length; B++) {
11452         var D = Z[B];
11453         if (D.getAttribute("src")) {
11454             var C = D.getAttribute("src");
11455             if (!e.support) {
11456                 D.setAttribute("src", C + "?ver=" + new Date().getTime());
11457             } else {
11458                 D.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11459             }
11460         }
11461     }
11462     var N = t.querySelectorAll("img");
11463     for (var F = 0; F < N.length; F++) {
11464         var G = N[F];
11465         if (G.getAttribute("src")) {
11466             var H = G.getAttribute("src");
11467             if (!e.support) {
11468                 G.setAttribute("src", H + "?ver=" + new Date().getTime());
11469             } else {
11470                 G.setAttribute("srcset", H + "?ver=" + new Date().getTime());
11471             }
11472         }
11473     }
11474     var P = t.querySelectorAll("img");
11475     for (var I = 0; I < P.length; I++) {
11476         var Q = P[I];
11477         if (Q.getAttribute("src")) {
11478             var R = Q.getAttribute("src");
11479             if (!e.support) {
11480                 Q.setAttribute("src", R + "?ver=" + new Date().getTime());
11481             } else {
11482                 Q.setAttribute("srcset", R + "?ver=" + new Date().getTime());
11483             }
11484         }
11485     }
11486     var O = t.querySelectorAll("img");
11487     for (var J = 0; J < O.length; J++) {
11488         var K = O[J];
11489         if (K.getAttribute("src")) {
11490             var L = K.getAttribute("src");
11491             if (!e.support) {
11492                 K.setAttribute("src", L + "?ver=" + new Date().getTime());
11493             } else {
11494                 K.setAttribute("srcset", L + "?ver=" + new Date().getTime());
11495             }
11496         }
11497     }
11498     var M = t.querySelectorAll("img");
11499     for (var H = 0; H < M.length; H++) {
11500         var I = M[H];
11501         if (I.getAttribute("src")) {
11502             var J = I.getAttribute("src");
11503             if (!e.support) {
11504                 I.setAttribute("src", J + "?ver=" + new Date().getTime());
11505             } else {
11506                 I.setAttribute("srcset", J + "?ver=" + new Date().getTime());
11507             }
11508         }
11509     }
11510     var N = t.querySelectorAll("img");
11511     for (var K = 0; K < N.length; K++) {
11512         var L = N[K];
11513         if (L.getAttribute("src")) {
11514             var M = L.getAttribute("src");
11515             if (!e.support) {
11516                 L.setAttribute("src", M + "?ver=" + new Date().getTime());
11517             } else {
11518                 L.setAttribute("srcset", M + "?ver=" + new Date().getTime());
11519             }
11520         }
11521     }
11522     var O = t.querySelectorAll("img");
11523     for (var J = 0; J < O.length; J++) {
11524         var I = O[J];
11525         if (I.getAttribute("src")) {
11526             var K = I.getAttribute("src");
11527             if (!e.support) {
11528                 I.setAttribute("src", K + "?ver=" + new Date().getTime());
11529             } else {
11530                 I.setAttribute("srcset", K + "?ver=" + new Date().getTime());
11531             }
11532         }
11533     }
11534     var P = t.querySelectorAll("img");
11535     for (var H = 0; H < P.length; H++) {
11536         var I = P[H];
11537         if (I.getAttribute("src")) {
11538             var J = I.getAttribute("src");
11539             if (!e.support) {
11540                 I.setAttribute("src", J + "?ver=" + new Date().getTime());
11541             } else {
11542                 I.setAttribute("srcset", J + "?ver=" + new Date().getTime());
11543             }
11544         }
11545     }
11546     var Q = t.querySelectorAll("img");
11547     for (var G = 0; G < Q.length; G++) {
11548         var F = Q[G];
11549         if (F.getAttribute("src")) {
11550             var H = F.getAttribute("src");
11551             if (!e.support) {
11552                 F.setAttribute("src", H + "?ver=" + new Date().getTime());
11553             } else {
11554                 F.setAttribute("srcset", H + "?ver=" + new Date().getTime());
11555             }
11556         }
11557     }
11558     var R = t.querySelectorAll("img");
11559     for (var E = 0; E < R.length; E++) {
11560         var D = R[E];
11561         if (D.getAttribute("src")) {
11562             var C = D.getAttribute("src");
11563             if (!e.support) {
11564                 D.setAttribute("src", C + "?ver=" + new Date().getTime());
11565             } else {
11566                 D.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11567             }
11568         }
11569     }
11570     var S = t.querySelectorAll("img");
11571     for (var B = 0; B < S.length; B++) {
11572         var A = S[B];
11573         if (A.getAttribute("src")) {
11574             var C = A.getAttribute("src");
11575             if (!e.support) {
11576                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11577             } else {
11578                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11579             }
11580         }
11581     }
11582     var T = t.querySelectorAll("img");
11583     for (var D = 0; D < T.length; D++) {
11584         var C = T[D];
11585         if (C.getAttribute("src")) {
11586             var B = C.getAttribute("src");
11587             if (!e.support) {
11588                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11589             } else {
11590                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11591             }
11592         }
11593     }
11594     var U = t.querySelectorAll("img");
11595     for (var F = 0; F < U.length; F++) {
11596         var E = U[F];
11597         if (E.getAttribute("src")) {
11598             var G = E.getAttribute("src");
11599             if (!e.support) {
11600                 E.setAttribute("src", G + "?ver=" + new Date().getTime());
11601             } else {
11602                 E.setAttribute("srcset", G + "?ver=" + new Date().getTime());
11603             }
11604         }
11605     }
11606     var V = t.querySelectorAll("img");
11607     for (var H = 0; H < V.length; H++) {
11608         var I = V[H];
11609         if (I.getAttribute("src")) {
11610             var J = I.getAttribute("src");
11611             if (!e.support) {
11612                 I.setAttribute("src", J + "?ver=" + new Date().getTime());
11613             } else {
11614                 I.setAttribute("srcset", J + "?ver=" + new Date().getTime());
11615             }
11616         }
11617     }
11618     var W = t.querySelectorAll("img");
11619     for (var G = 0; G < W.length; G++) {
11620         var F = W[G];
11621         if (F.getAttribute("src")) {
11622             var H = F.getAttribute("src");
11623             if (!e.support) {
11624                 F.setAttribute("src", H + "?ver=" + new Date().getTime());
11625             } else {
11626                 F.setAttribute("srcset", H + "?ver=" + new Date().getTime());
11627             }
11628         }
11629     }
11630     var X = t.querySelectorAll("img");
11631     for (var E = 0; E < X.length; E++) {
11632         var D = X[E];
11633         if (D.getAttribute("src")) {
11634             var C = D.getAttribute("src");
11635             if (!e.support) {
11636                 D.setAttribute("src", C + "?ver=" + new Date().getTime());
11637             } else {
11638                 D.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11639             }
11640         }
11641     }
11642     var Y = t.querySelectorAll("img");
11643     for (var B = 0; B < Y.length; B++) {
11644         var A = Y[B];
11645         if (A.getAttribute("src")) {
11646             var C = A.getAttribute("src");
11647             if (!e.support) {
11648                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11649             } else {
11650                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11651             }
11652         }
11653     }
11654     var Z = t.querySelectorAll("img");
11655     for (var D = 0; D < Z.length; D++) {
11656         var C = Z[D];
11657         if (C.getAttribute("src")) {
11658             var B = C.getAttribute("src");
11659             if (!e.support) {
11660                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11661             } else {
11662                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11663             }
11664         }
11665     }
11666     var P = t.querySelectorAll("img");
11667     for (var H = 0; H < P.length; H++) {
11668         var I = P[H];
11669         if (I.getAttribute("src")) {
11670             var G = I.getAttribute("src");
11671             if (!e.support) {
11672                 I.setAttribute("src", G + "?ver=" + new Date().getTime());
11673             } else {
11674                 I.setAttribute("srcset", G + "?ver=" + new Date().getTime());
11675             }
11676         }
11677     }
11678     var Q = t.querySelectorAll("img");
11679     for (var F = 0; F < Q.length; F++) {
11680         var E = Q[F];
11681         if (E.getAttribute("src")) {
11682             var D = E.getAttribute("src");
11683             if (!e.support) {
11684                 E.setAttribute("src", D + "?ver=" + new Date().getTime());
11685             } else {
11686                 E.setAttribute("srcset", D + "?ver=" + new Date().getTime());
11687             }
11688         }
11689     }
11690     var R = t.querySelectorAll("img");
11691     for (var D = 0; D < R.length; D++) {
11692         var C = R[D];
11693         if (C.getAttribute("src")) {
11694             var B = C.getAttribute("src");
11695             if (!e.support) {
11696                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11697             } else {
11698                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11699             }
11700         }
11701     }
11702     var S = t.querySelectorAll("img");
11703     for (var B = 0; B < S.length; B++) {
11704         var A = S[B];
11705         if (A.getAttribute("src")) {
11706             var C = A.getAttribute("src");
11707             if (!e.support) {
11708                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11709             } else {
11710                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11711             }
11712         }
11713     }
11714     var T = t.querySelectorAll("img");
11715     for (var A = 0; A < T.length; A++) {
11716         var C = T[A];
11717         if (C.getAttribute("src")) {
11718             var B = C.getAttribute("src");
11719             if (!e.support) {
11720                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11721             } else {
11722                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11723             }
11724         }
11725     }
11726     var U = t.querySelectorAll("img");
11727     for (var C = 0; C < U.length; C++) {
11728         var B = U[C];
11729         if (B.getAttribute("src")) {
11730             var A = B.getAttribute("src");
11731             if (!e.support) {
11732                 B.setAttribute("src", A + "?ver=" + new Date().getTime());
11733             } else {
11734                 B.setAttribute("srcset", A + "?ver=" + new Date().getTime());
11735             }
11736         }
11737     }
11738     var V = t.querySelectorAll("img");
11739     for (var A = 0; A < V.length; A++) {
11740         var C = V[A];
11741         if (C.getAttribute("src")) {
11742             var B = C.getAttribute("src");
11743             if (!e.support) {
11744                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11745             } else {
11746                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11747             }
11748         }
11749     }
11750     var W = t.querySelectorAll("img");
11751     for (var B = 0; B < W.length; B++) {
11752         var A = W[B];
11753         if (A.getAttribute("src")) {
11754             var C = A.getAttribute("src");
11755             if (!e.support) {
11756                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11757             } else {
11758                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11759             }
11760         }
11761     }
11762     var X = t.querySelectorAll("img");
11763     for (var A = 0; A < X.length; A++) {
11764         var C = X[A];
11765         if (C.getAttribute("src")) {
11766             var B = C.getAttribute("src");
11767             if (!e.support) {
11768                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11769             } else {
11770                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11771             }
11772         }
11773     }
11774     var Y = t.querySelectorAll("img");
11775     for (var B = 0; B < Y.length; B++) {
11776         var A = Y[B];
11777         if (A.getAttribute("src")) {
11778             var C = A.getAttribute("src");
11779             if (!e.support) {
11780                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11781             } else {
11782                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11783             }
11784         }
11785     }
11786     var Z = t.querySelectorAll("img");
11787     for (var A = 0; A < Z.length; A++) {
11788         var C = Z[A];
11789         if (C.getAttribute("src")) {
11790             var B = C.getAttribute("src");
11791             if (!e.support) {
11792                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11793             } else {
11794                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11795             }
11796         }
11797     }
11798     var P = t.querySelectorAll("img");
11799     for (var C = 0; C < P.length; C++) {
11800         var B = P[C];
11801         if (B.getAttribute("src")) {
11802             var A = B.getAttribute("src");
11803             if (!e.support) {
11804                 B.setAttribute("src", A + "?ver=" + new Date().getTime());
11805             } else {
11806                 B.setAttribute("srcset", A + "?ver=" + new Date().getTime());
11807             }
11808         }
11809     }
11810     var Q = t.querySelectorAll("img");
11811     for (var A = 0; A < Q.length; A++) {
11812         var C = Q[A];
11813         if (C.getAttribute("src")) {
11814             var B = C.getAttribute("src");
11815             if (!e.support) {
11816                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11817             } else {
11818                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11819             }
11820         }
11821     }
11822     var R = t.querySelectorAll("img");
11823     for (var B = 0; B < R.length; B++) {
11824         var A = R[B];
11825         if (A.getAttribute("src")) {
11826             var C = A.getAttribute("src");
11827             if (!e.support) {
11828                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11829             } else {
11830                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11831             }
11832         }
11833     }
11834     var S = t.querySelectorAll("img");
11835     for (var A = 0; A < S.length; A++) {
11836         var C = S[A];
11837         if (C.getAttribute("src")) {
11838             var B = C.getAttribute("src");
11839             if (!e.support) {
11840                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11841             } else {
11842                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11843             }
11844         }
11845     }
11846     var T = t.querySelectorAll("img");
11847     for (var B = 0; B < T.length; B++) {
11848         var A = T[B];
11849         if (A.getAttribute("src")) {
11850             var C = A.getAttribute("src");
11851             if (!e.support) {
11852                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11853             } else {
11854                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11855             }
11856         }
11857     }
11858     var U = t.querySelectorAll("img");
11859     for (var A = 0; A < U.length; A++) {
11860         var C = U[A];
11861         if (C.getAttribute("src")) {
11862             var B = C.getAttribute("src");
11863             if (!e.support) {
11864                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11865             } else {
11866                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11867             }
11868         }
11869     }
11870     var V = t.querySelectorAll("img");
11871     for (var B = 0; B < V.length; B++) {
11872         var A = V[B];
11873         if (A.getAttribute("src")) {
11874             var C = A.getAttribute("src");
11875             if (!e.support) {
11876                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11877             } else {
11878                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11879             }
11880         }
11881     }
11882     var W = t.querySelectorAll("img");
11883     for (var A = 0; A < W.length; A++) {
11884         var C = W[A];
11885         if (C.getAttribute("src")) {
11886             var B = C.getAttribute("src");
11887             if (!e.support) {
11888                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11889             } else {
11890                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11891             }
11892         }
11893     }
11894     var X = t.querySelectorAll("img");
11895     for (var B = 0; B < X.length; B++) {
11896         var A = X[B];
11897         if (A.getAttribute("src")) {
11898             var C = A.getAttribute("src");
11899             if (!e.support) {
11900                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11901             } else {
11902                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11903             }
11904         }
11905     }
11906     var Y = t.querySelectorAll("img");
11907     for (var A = 0; A < Y.length; A++) {
11908         var C = Y[A];
11909         if (C.getAttribute("src")) {
11910             var B = C.getAttribute("src");
11911             if (!e.support) {
11912                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11913             } else {
11914                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11915             }
11916         }
11917     }
11918     var Z = t.querySelectorAll("img");
11919     for (var B = 0; B < Z.length; B++) {
11920         var A = Z[B];
11921         if (A.getAttribute("src")) {
11922             var C = A.getAttribute("src");
11923             if (!e.support) {
11924                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11925             } else {
11926                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11927             }
11928         }
11929     }
11930     var P = t.querySelectorAll("img");
11931     for (var C = 0; C < P.length; C++) {
11932         var B = P[C];
11933         if (B.getAttribute("src")) {
11934             var A = B.getAttribute("src");
11935             if (!e.support) {
11936                 B.setAttribute("src", A + "?ver=" + new Date().getTime());
11937             } else {
11938                 B.setAttribute("srcset", A + "?ver=" + new Date().getTime());
11939             }
11940         }
11941     }
11942     var Q = t.querySelectorAll("img");
11943     for (var A = 0; A < Q.length; A++) {
11944         var C = Q[A];
11945         if (C.getAttribute("src")) {
11946             var B = C.getAttribute("src");
11947             if (!e.support) {
11948                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11949             } else {
11950                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11951             }
11952         }
11953     }
11954     var R = t.querySelectorAll("img");
11955     for (var B = 0; B < R.length; B++) {
11956         var A = R[B];
11957         if (A.getAttribute("src")) {
11958             var C = A.getAttribute("src");
11959             if (!e.support) {
11960                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11961             } else {
11962                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11963             }
11964         }
11965     }
11966     var S = t.querySelectorAll("img");
11967     for (var A = 0; A < S.length; A++) {
11968         var C = S[A];
11969         if (C.getAttribute("src")) {
11970             var B = C.getAttribute("src");
11971             if (!e.support) {
11972                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11973             } else {
11974                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11975             }
11976         }
11977     }
11978     var T = t.querySelectorAll("img");
11979     for (var B = 0; B < T.length; B++) {
11980         var A = T[B];
11981         if (A.getAttribute("src")) {
11982             var C = A.getAttribute("src");
11983             if (!e.support) {
11984                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
11985             } else {
11986                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
11987             }
11988         }
11989     }
11990     var U = t.querySelectorAll("img");
11991     for (var A = 0; A < U.length; A++) {
11992         var C = U[A];
11993         if (C.getAttribute("src")) {
11994             var B = C.getAttribute("src");
11995             if (!e.support) {
11996                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
11997             } else {
11998                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
11999             }
12000         }
12001     }
12002     var V = t.querySelectorAll("img");
12003     for (var B = 0; B < V.length; B++) {
12004         var A = V[B];
12005         if (A.getAttribute("src")) {
12006             var C = A.getAttribute("src");
12007             if (!e.support) {
12008                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
12009             } else {
12010                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
12011             }
12012         }
12013     }
12014     var W = t.querySelectorAll("img");
12015     for (var A = 0; A < W.length; A++) {
12016         var C = W[A];
12017         if (C.getAttribute("src")) {
12018             var B = C.getAttribute("src");
12019             if (!e.support) {
12020                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
12021             } else {
12022                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
12023             }
12024         }
12025     }
12026     var X = t.querySelectorAll("img");
12027     for (var B = 0; B < X.length; B++) {
12028         var A = X[B];
12029         if (A.getAttribute("src")) {
12030             var C = A.getAttribute("src");
12031             if (!e.support) {
12032                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
12033             } else {
12034                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
12035             }
12036         }
12037     }
12038     var Y = t.querySelectorAll("img");
12039     for (var A = 0; A < Y.length; A++) {
12040         var C = Y[A];
12041         if (C.getAttribute("src")) {
12042             var B = C.getAttribute("src");
12043             if (!e.support) {
12044                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
12045             } else {
12046                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
12047             }
12048         }
12049     }
12050     var Z = t.querySelectorAll("img");
12051     for (var B = 0; B < Z.length; B++) {
12052         var A = Z[B];
12053         if (A.getAttribute("src")) {
12054             var C = A.getAttribute("src");
12055             if (!e.support) {
12056                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
12057             } else {
12058                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
12059             }
12060         }
12061     }
12062     var P = t.querySelectorAll("img");
12063     for (var C = 0; C < P.length; C++) {
12064         var B = P[C];
12065         if (B.getAttribute("src")) {
12066             var A = B.getAttribute("src");
12067             if (!e.support) {
12068                 B.setAttribute("src", A + "?ver=" + new Date().getTime());
12069             } else {
12070                 B.setAttribute("srcset", A + "?ver=" + new Date().getTime());
12071             }
12072         }
12073     }
12074     var Q = t.querySelectorAll("img");
12075     for (var A = 0; A < Q.length; A++) {
12076         var C = Q[A];
12077         if (C.getAttribute("src")) {
12078             var B = C.getAttribute("src");
12079             if (!e.support) {
12080                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
12081             } else {
12082                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
12083             }
12084         }
12085     }
12086     var R = t.querySelectorAll("img");
12087     for (var B = 0; B < R.length; B++) {
12088         var A = R[B];
12089         if (A.getAttribute("src")) {
12090             var C = A.getAttribute("src");
12091             if (!e.support) {
12092                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
12093             } else {
12094                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
12095             }
12096         }
12097     }
12098     var S = t.querySelectorAll("img");
12099     for (var A = 0; A < S.length; A++) {
12100         var C = S[A];
12101         if (C.getAttribute("src")) {
12102             var B = C.getAttribute("src");
12103             if (!e.support) {
12104                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
12105             } else {
12106                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
12107             }
12108         }
12109     }
12110     var T = t.querySelectorAll("img");
12111     for (var B = 0; B < T.length; B++) {
12112         var A = T[B];
12113         if (A.getAttribute("src")) {
12114             var C = A.getAttribute("src");
12115             if (!e.support) {
12116                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
12117             } else {
12118                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
12119             }
12120         }
12121     }
12122     var U = t.querySelectorAll("img");
12123     for (var A = 0; A < U.length; A++) {
12124         var C = U[A];
12125         if (C.getAttribute("src")) {
12126             var B = C.getAttribute("src");
12127             if (!e.support) {
12128                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
12129             } else {
12130                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
12131             }
12132         }
12133     }
12134     var V = t.querySelectorAll("img");
12135     for (var B = 0; B < V.length; B++) {
12136         var A = V[B];
12137         if (A.getAttribute("src")) {
12138             var C = A.getAttribute("src");
12139             if (!e.support) {
12140                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
12141             } else {
12142                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
12143             }
12144         }
12145     }
12146     var W = t.querySelectorAll("img");
12147     for (var A = 0; A < W.length; A++) {
12148         var C = W[A];
12149         if (C.getAttribute("src")) {
12150             var B = C.getAttribute("src");
12151             if (!e.support) {
12152                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
12153             } else {
12154                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
12155             }
12156         }
12157     }
12158     var X = t.querySelectorAll("img");
12159     for (var B = 0; B < X.length; B++) {
12160         var A = X[B];
12161         if (A.getAttribute("src")) {
12162             var C = A.getAttribute("src");
12163             if (!e.support) {
12164                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
12165             } else {
12166                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
12167             }
12168         }
12169     }
12170     var Y = t.querySelectorAll("img");
12171     for (var A = 0; A < Y.length; A++) {
12172         var C = Y[A];
12173         if (C.getAttribute("src")) {
12174             var B = C.getAttribute("src");
12175             if (!e.support) {
12176                 C.setAttribute("src", B + "?ver=" + new Date().getTime());
12177             } else {
12178                 C.setAttribute("srcset", B + "?ver=" + new Date().getTime());
12179             }
12180         }
12181     }
12182     var Z = t.querySelectorAll("img");
12183     for (var B = 0; B < Z.length; B++) {
12184         var A = Z[B];
12185         if (A.getAttribute("src")) {
12186             var C = A.getAttribute("src");
12187             if (!e.support) {
12188                 A.setAttribute("src", C + "?ver=" + new Date().getTime());
12189             } else {
12190                 A.setAttribute("srcset", C + "?ver=" + new Date().getTime());
12191             }
12192         }
12193     }
12194     var P = t.querySelectorAll("img");
12195     for (var C = 0; C < P.length; C++) {
12196         var B = P[C];
12197         if (B.getAttribute("src")) {
12198             var A = B.getAttribute("src");
12199             if (!e.support) {
12200                 B.setAttribute("src", A + "?ver=" + new Date().getTime());
12201             } else {
12202                 B.setAttribute("srcset", A + "?ver=" + new Date().getTime());
12203             }
12204         }
12205     }
12206     var Q = t.querySelectorAll("img");
12207     for (var A = 0; A < Q.length; A++) {
12208         var C = Q[A];
12209         if (C.getAttribute("src")) {
12210             var B = C.getAttribute("src");
12211             if (!e.support) {
12212                 C.setAttribute("src", B + "?ver=" + new Date().getTime());

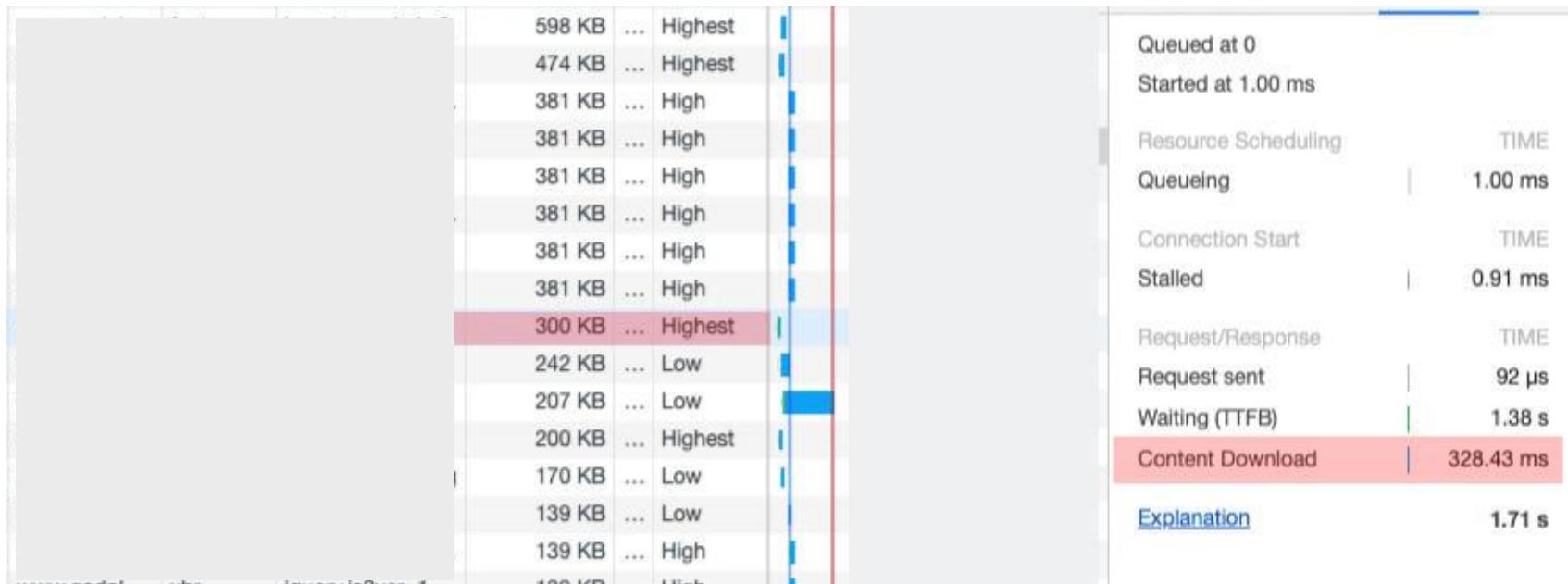
```

3.4. Theme issue – Excessive HTML size + Nesting

▲ Avoid an excessive DOM size – 4,079 elements

Browser engineers recommend pages contain fewer than ~1,500 DOM elements. The sweet spot is a tree depth < 32 elements and fewer than 60 children/parent element. A large DOM can increase memory usage, cause longer style calculations, and produce costly layout reflows.

3.5. Page document too large (300kb incl. code)

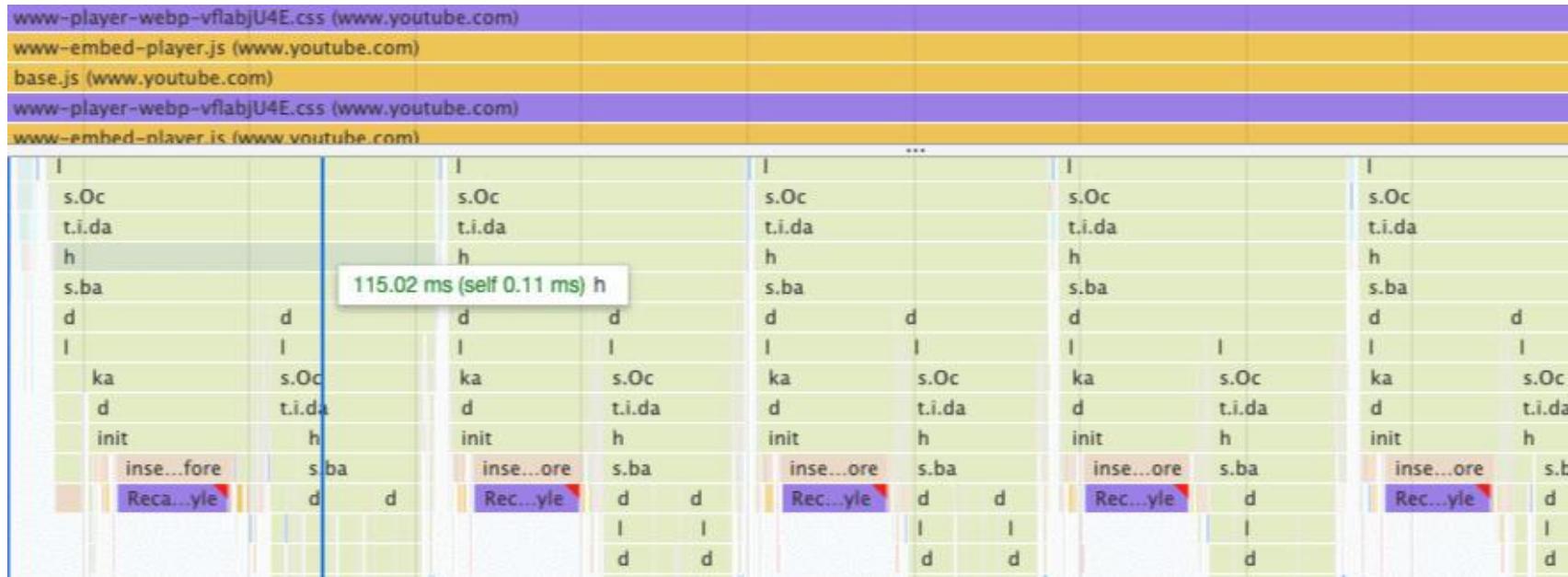


4.1. YouTubeplugins – Slow load +Rendering

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#)

URL	Total CPU Time	Script Evaluation	Script Parse
Other	6,022 ms	368 ms	18 ms
...www-embed.../www-embed-player.js (www.youtube.com)	5,539 ms	4,904 ms	100 ms
...jquery/jquery.js?ver=1.12.4-wp (www.gadplan.com)	4,067 ms	2,529 ms	23 ms
...en_US/base.js (www.youtube.com)	1,274 ms	576 ms	698 ms

4.2. YouTube- Loaded multiple times +Slow

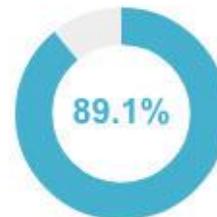


5. Enable GZIP – Small text files (HTML, CSS, JS, ...)

Check GZIP compression

Oh No! GZIP is not enabled :(

Domain Name	Gadplan.com
Compressed size	33346 bytes (~33 KB)
Uncompressed size	305784 bytes (~299 KB)



Could be saved by
compressing this page with
GZIP.

6.1. Fix 301/302 redirects from CDN

The screenshot shows a browser's developer tools Network tab. On the left, there is a list of network requests, each with a status code of 302 and a URL starting with "https://". A red box highlights the status code 302 in the first row. To the right, a detailed view of the first request is shown. The "General" section contains the following information:

- Request Method: GET
- Status Code: 302
- Remote Address: 151.139.128.10:443
- Referrer Policy: no-referrer-when-downgrade

Below this, there are sections for "Response Headers" (11) and "Request Headers". A note indicates that provisional headers are shown. The "Request Headers" section includes DNT: 1, Referer: (redacted), and User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36.

6.2. Ineffective CDN use – Redirects, Few files

GET	302	https	cdn.shortpixel.ai	text/html	lazysizes.min.js:2	126 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	155 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	155 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	164 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	179 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	179 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	188 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	203 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	204 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	204 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	209 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	210 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	217 B	...
GET	302	https	cdn.shortpixel.ai	text/html	(index)	220 B	...
GET	302	https	cdn.shortpixel.ai	text/html	jquery.js?ver=1.1...	228 B	...

6.3. Fix http to https redirects – Speed +Security

Group similar

⚠ Mixed Content: The page at '[https://www.example.com/](#)' was loaded over HTTPS, but requested an insecure image '[http://www.example.com/dplan_300px.jpg](#)'. This content should also be served over HTTPS.

✖ **Uncaught ReferenceError: responsive_functions is not defined**
at [\(index\):1768](#)

JQMIGRATE: Migrate is installed, version 1.4.1

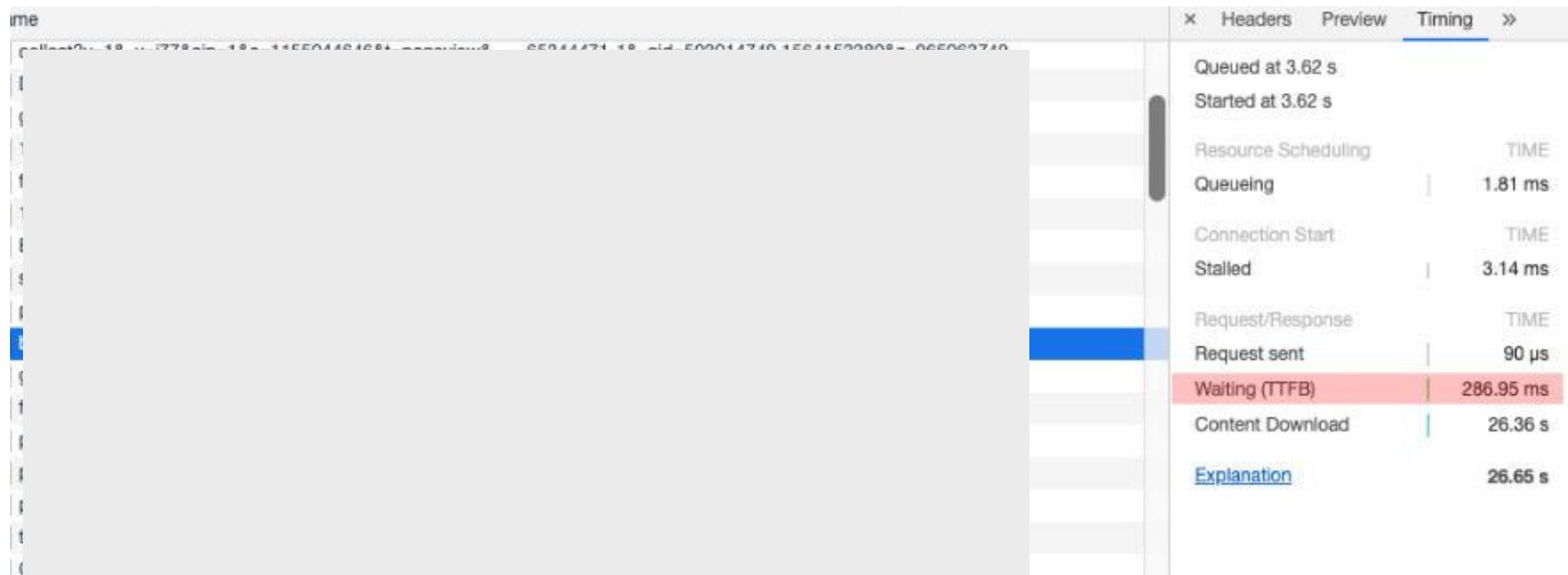
3 !
3 !
3 !
3 !
3 !
3 !

⚠ Mixed Content: The page at '[https://www.example.com/](#)' was loaded over HTTPS, but requested an insecure image '[http://www.example.com/dplan_300px.jpg](#)'. This content should also be served over HTTPS.

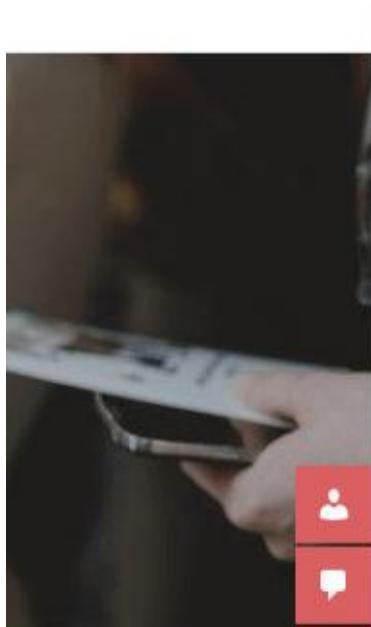
✖ **Failed to execute 'postMessage' on 'DOMWindow': The target origin provided ('<https://www.youtube.com>') does not match the recipient's ('[https://www.example.com](#)')**

>

7.1.Slow server- Long delay before page load



7.2.Server – Utilise HTTP/2 protocol for assets



Best Practices

57

- | | | | |
|---|---|-----------------------------------|-----|
| 1 | Does not use HTTPS | 2 insecure requests found | ▲ ▾ |
| 2 | Does not use HTTP/2 for all of its resources | 67 requests not served via HTTP/2 | ▲ ▾ |
| 3 | Does not use passive listeners to improve scrolling performance | | ▲ ▾ |
| 4 | Links to cross-origin destinations are unsafe | | ▲ ▾ |
| 5 | Includes front-end JavaScript libraries with known security vulnerabilities | 2 vulnerabilities detected | ▲ ▾ |
| 6 | Browser errors were logged to the console | | ▲ ▾ |

Passed audits

9 audits ▾

Directions settings

8. Too many font variants (unused alphabets)

```
font-weight: 400;
src: local('Poppins Regular'), local('Poppins-Regular'), url(https://fonts.gstatic.com/s/poppins/v8/pxiEyp8kv8J3gFVrJJbecnFHGPezSQ.woff2) format('woff2');
unicode-range: U+0900-097F, U+1CD0-1CF6, U+1CF8-1CF9, U+200C-200D, U+20A8, U+20B9, U+25CC, U+A830-A839, U+A8E0-A8FB;
}
/* latin-ext */
@font-face {
  font-family: 'Poppins';
  font-style: normal;
  font-weight: 400;
  src: local('Poppins Regular'), local('Poppins-Regular'), url(https://fonts.gstatic.com/s/poppins/v8/pxiEyp8kv8J3gFVrJJnecnFHGPezSQ.woff2) format('woff2');
  unicode-range: U+0100-024F, U+0259, U+1E00-1EFF, U+2020, U+20A0-20AB, U+20AD=20CF, U+2113, U+2C60-2C7F, U+A720-A7FF;
}
/* latin */
@font-face {
  font-family: 'Poppins';
  font-style: normal;
  font-weight: 400;
  src: local('Poppins Regular'), local('Poppins-Regular'), url(https://fonts.gstatic.com/s/poppins/v8/pxiEyp8kv8J3gFVrJJfecnFHGPc.woff2) format('woff2');
  unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02BB-02BC, U+02C6, U+02DA, U+2000-206F, U+2074, U+20AC, U+2122, U+2191, U+2193, U+2212, U+2215, U+FEFF, U+FFFF;
}
/* cyrillic-ext */
@font-face {
  font-family: 'Roboto';
  font-style: italic;
  font-weight: 100;
  src: local('Roboto Thin Italic'), local('Roboto-ThinItalic'), url(https://fonts.gstatic.com/s/roboto/v20/KPoICnqEu92Fr1Mu51QrRz0dL-vwmYh2eg.woff2) format('woff2');
  unicode-range: U+0460-052F, U+1C80-1C88, U+20B4, U+2DE0-2DFF, U+A640-A69F, U+FE2E-FE2F;
}
/* cyrillic */
@font-face {
  font-family: 'Roboto';
  font-style: italic;
  font-weight: 100;
  src: local('Roboto Thin Italic'), local('Roboto-ThinItalic'), url(https://fonts.gstatic.com/s/roboto/v20/KPoICnqEu92Fr1Mu51QrRzQdL-vwmYh2eg.woff2) format('woff2');
  unicode-range: U+0400-045F, U+0490-0491, U+04B0-04B1, U+2116;
```

9.1.WP Autoptimize – Emojis, Preconnect, YouTube

Portfolio
Waldara Formulare
Design
Plugins 7
Benutzer
Werkzeuge
PBakery Page Builder
PBackItUp
Einstellungen
Nein ben

Emojis entfernen

Entfernt WordPress' internes Emoji-Inline-CSS, Inline JavaScript und eine weitere nicht optimierte JavaScript.

Abfragezeichenfolgen von statischen Ressourcen entfernen

Das Entfernen von Abfrage-Zeichenfolgen (oder genauer gesagt, des `ver`-Parameters) wird die Ladegeschwindigkeit der Leistungsmessungen verbessern.

Zu Dritt-Domains vorverbinden *(Fortgeschrittene Benutzer)*

Füge Drittanbieter-Domains hinzu, zu denen sich der Browser [vorverbinden](#) soll, durch Komma getrennt. Stelle anzugeben (HTTP oder HTTPS).

Asynchrone JavaScript-Dateien *(Fortgeschrittene Benutzer)*

Kommagetrennte Liste lokaler oder Drittanbieter-JS-Dateien, die per `async` geladen werden sollen. JS-Dateien automatisch ausgeschlossen, falls hier aufgeführt. Die Konfiguration von asynchronem JavaScript ist einfacher als [JavaScript](#). [Hier klicken, um es zu installieren und zu aktivieren.](#)

Optimiere YouTube-Videos

[WP YouTube Lyte](#) ermöglicht dir, deine Videos durch das Einfügen von responsiven „Lite YouTube Embeds“ zu installieren und zu aktivieren.

9.2. WordPress Autoptimize – Optimise CSS?

CSS-Optionen	
CSS-Code optimieren?	<input type="checkbox"/>
CSS-Dateien zusammenfügen?	<input checked="" type="checkbox"/> Alle verlinkten CSS-Dateien zusammenfügen? Falls diese Option nicht gewählt wird, werden individuelle CSS-Dateien an ihrem Ort verbleiben, aber minimiert werden.
Auch Inline-CSS zusammenfügen?	<input checked="" type="checkbox"/> Wähle diese Option, wenn Autoptimize CSS auch innerhalb des HTML hinzufügt.

10. If possible: Remove 'admin-ajax' calls

The screenshot shows a browser developer tools Network tab with a list of requests. One specific request is highlighted in blue. The details panel on the right is open for this request.

General

- Request URL: ajax.php
- Request Method: POST
- Status Code: 200 OK
- Remote Address: 87.230.94.39:443
- Referrer Policy: no-referrer-when-dc

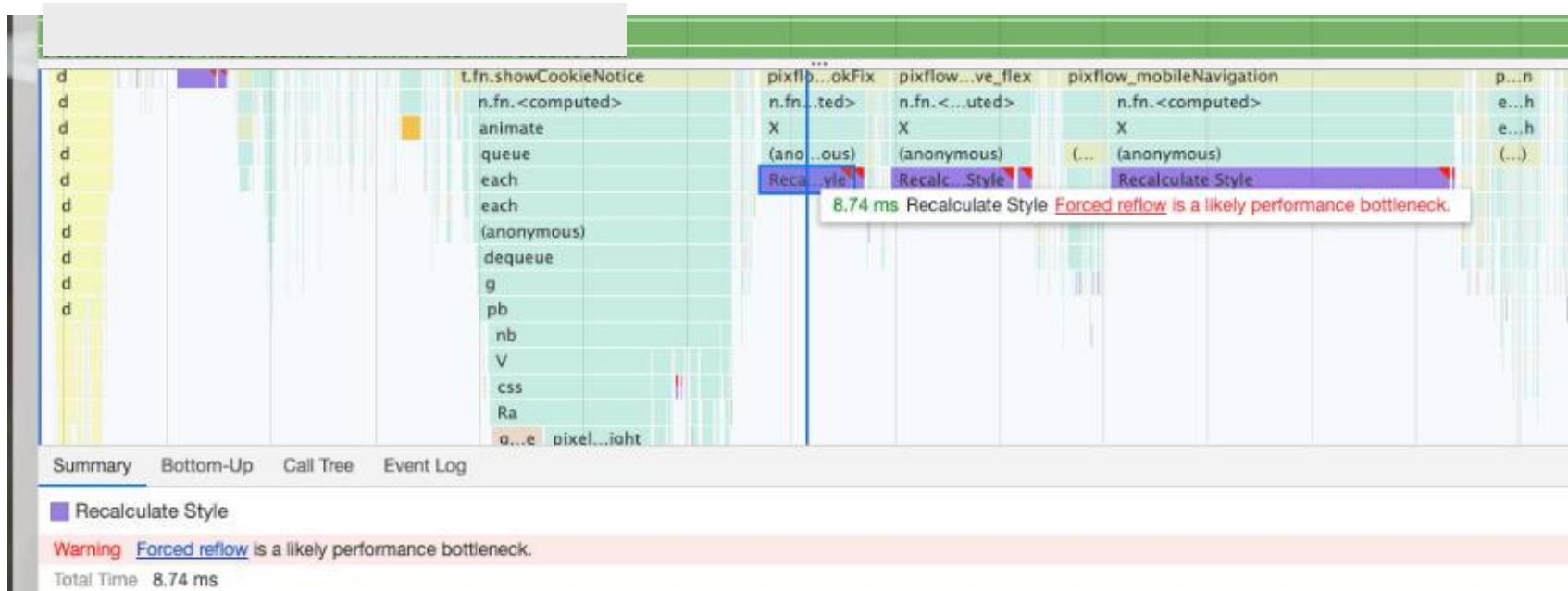
Response Headers (17)

Request Headers

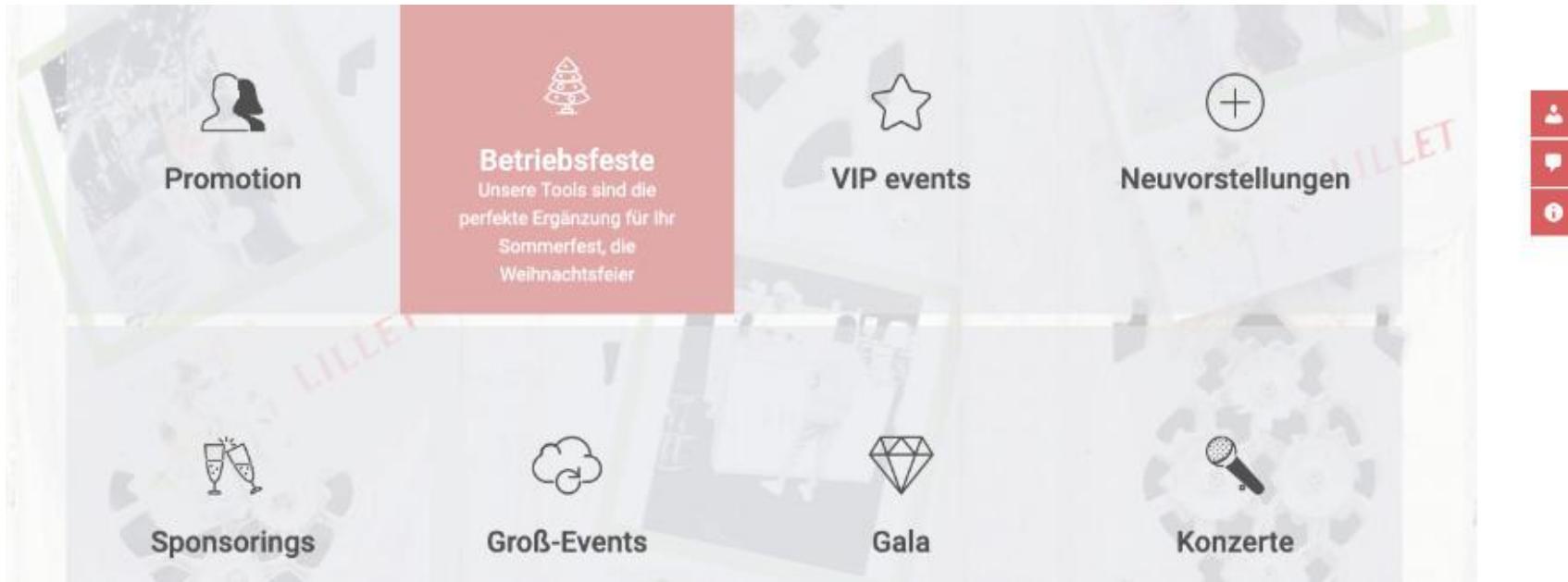
- Accept: */*
- Accept-Encoding: gzip, deflate, br
- Accept-Language: en-US,en;q=0.9,d

0.7

11. Style reflows – jQuery (pixflow navigation)



12. UI animation feels slow/laggy (speed up)



4. Recommendations



Solutions – 1/3

Possible solutions

1st priority

High impact and benefits

Varying implementation cost

Problems 1 – 3

- Reduce image files and amount
(might result in needed layout changes)
 - Replace some images by text
- Use dedicated WordPress plugin(s) to minify/bundle assets (CSS, JS)
 - e.g. W3 Total Cache
 - Requires extensive browser + functionality testing afterwards
- Implement image lazy load solution
 - Might not work with all themes
 - Requires extensive layout + functionality testing afterwards
 - Don't lazy load first/header image that users will see on page entry



Solutions – 2/3

Possible solutions

2nd priority

Less impact, still important

Problems 4 – 6

- Remove or replace animated GIF by Video or static image (<1.9 MB)
- Change all URLs from 'http://' to 'https://' to remove 301 redirects
 - The http-URLs seem to come from a theme /custom CSS editor (plugin?).
 - Fix cdn.shortpixel.ai usage
- Hack: Remove admin_ajax calls from frontend (if possible)^{*10}
 - It can reduce requests and speed up the page a little bit (low impact)
 - It might be needed though, depending on functionality or theme



Solutions – 3/3

Possible solutions

3rd priority

High impact and benefits; but also
high effort & realisation cost

Problems 4 – 6

Important / High effort & realisation cost

- Change theme to a more performant
 - Drawback: Content +Features will have to change,refactoring needed (plugins,3rd party integrations)
 - The code is slow and cluttered
 - See chapter [Resources & Links](#)
- Upgrade or move server hosting
 - Slows connection(TTFB^{*8}) delays each file download (CSS, JS, images)
 - High effort & higher maintenance costs; Developer might be needed

5. Further Considerations

Advanced techniques

Additional tips / No in-depth description / Developer needed



WordPress setup

Themes & Optimisations

- Disable unneeded plugins. Check and revise app plugin settings, e.g. ‘Heartbeat’ or DB/Object caches.^{*9}
- If choosing a new theme; consider focus on performance first, then features (and prefer to use built in).
 - Choose a theme without jQuery (these are usually slower during load and runtime). Note: It might not be compatible with other widely used plugins!
 - Fight spam: Disable Pingbacks / Trackbacks / Reblog feature (if enabled).^{*4}



Database config

Architecture & Optimisation

- Use a plugin to regularly cleanup and optimise the DB: Remove orphaned data, stored revisions, reduce used disk space, improve memory usage.^{*6}
- Change tables to InnoDB instead of MyISAM to benefit of different memory and cache settings.^{*6}
 - Meaning: Older tables or those created by plugins, if compatible.
 - These settings need to be individually defined for your server environment!



PHP +Apache

Server settings

- Check server +WordPress config: Use PHP 7+ instead of PHP 5.
- Set PHP memory_limit to a higher value if possible (e.g. 128/256MB).
 - Might be possible in .htaccess (likely not on small/shared host packages)
- Depending on editors needs:
 - Increase upload file limits
 - Increase execution times



Image hotlinking

Prevent & Fight back

- **htaccess:** Prevent hotlinking of images by server config.^{*5}
 - Other websites that embed your content steal your bandwidth and reduce server speed.
 - It helps to prevent content stealing or duplicate content.
 - Black magic trick: You might use a fallback image that is displayed on these sites to link back to your own website or simply display that the content has been stolen. BUT it also costs bandwidth and might not always be displayed in proper context, depending on the website that steals your content.

Layout + Content

Reduce complexity

- Utilise a ‘display more’ link or page numbers to limit and load content only if requested by users.
- Don’t implement image sliders, videos or image-based animations on the most important landing pages.
 - They are ‘heavy’ in load and runtime performance.
 - Keep it minimal: Prefer smaller static images, text or SVG vector graphics.
- Progressive content enhancement:
Start with static elements and exchange later with interactive alternatives (e.g. on second load when cache is ‘warm’ or out of initial viewport).

6. Terms & Definitions

Terms A–M

- **Asset:** Any files: CSS, JS, Images, Audio/Video, Documents (like PDF) ...
- **Bundle/Bundling:** The technique of combining many files into one to reduce file sizes and speed up load times. Allows for better minification.
 - Bundling might lead to code errors if not done or tested properly (esp. in JS)
- **CSS:** Styling descriptions – Layout, Colors, Fonts, Animations, ...
- **JS:** JavaScript – The ‘interactive logic’ part. Animations, Tracking, Forms, Pop-ups, ...
- **Lazy loading:** The technique of loading only what is in the currently visible user viewport.
- **Minify/Minification:** The technique of reducing file sizes – text, code, images – by stripping irrelevant information: White spaces, comments, unused colors, ... Allows for great size reduction.
 - Some tools can remove unused code (especially when bundling many files).

Terms M-W

- **MPEG4:** Video file format that can offer reduced file sizes. Can be used as replacement for GIF images.
 - Needs a converter tool and coding.
 - Might not work in all themes.
- **SEO:** ‘Search Engine Optimisation’ – nowadays highly related with page speed.
- **SSD:** Solid-state drive (Flash drive) – Faster than a hard drive (HDD).
- **TTFB:** ‘Time to first Byte’: The time spent waiting for the first server response (for each file).^{*8}
- **Viewport:** The part of the website users currently see on their browser screen.
- **VPS:** ‘Virtual Private Server’ – Virtual server that offers installation and configuration of any desired software.
- **WebP:** Modern imageformat that can offer reduced file sizescomparedto JPG/PNG.
 - Needs a plugin to use within WordPress.
 - Might not work with all themes.

Definitions

- **Backend:** The source code that runs server-side ‘behind the scenes’. Sometimes ambiguously used for the admin panel.
- **Frontend:** The publicly ‘visible’ part of the website that anyone can see.
- **Performance:** How fast the page content in the currentview is visible and responds to user interaction (e.g. by clicking, touch-and swipe gestures, scrolling, filling forms, ...)

7.Resources & Links



WordPress – Performance plugins

- [Autoptimize \(CSS, JS, Images\)](#)
- [WP Optimize / Advanced Database Cleaner](#)
- [W3 TotalCache^{*7} / WP Super Cache](#)
- [WP YouTubeLyte –YouTube performance](#)
- [Lazy Load for Comments](#)
- [WP Smushit\(Images\)](#)



WordPress – Performance focused themes 1/2

Review & Comparison of performance focused themes:

- helpiewp.com -5
- wp-rocket.me -10
- hostinger.com -15
- codeinwp.com -20+
- athemes.com -24
- colorlib.com -34
- [wplibuffs.com](http://wpbuffs.com) -53



WordPress – Performance focused themes 2/2

Selection of a few themes:

- [WP Astra](#) – VanillaJS (nojQuery)
- [Generate Press](#) – VanillaJS
- [Neve -Themelsle](#) – VanillaJS
- [The Gem -Codex Themes](#) – Uses jQuery / Offers many tempting ‘slow’ features

Current theme:

- [Massive Dynamic](#)

Testing tools

Speed tests

- [Google Page Speed Insights](#)
- [Google Lighthouse –Chrome DevTool extension](#)

Monitoring

- [Freshping](#)
- [StatusCake](#)
- [ServiceUptime](#)
- [Pingdom](#)

Annotations / Footnotes

1. [Think with Google –Industry benchmarks](#)
2. [MachMetrics –Why Page Speed Scores can be dangerous](#)
3. [WPMUDev Blog –Don't try to get 95 on Google PageSpeed Insights](#)
4. [Trackbacks –Pingbacks –Reblogs](#)
5. [Prevent imagehotlinkingin WordPress / Speed up WordPress –Hotlinking](#)
6. [WordPress Database Optimisation Guide /DB Optimisation –Best Practices](#)
7. [W3 Total Cache –Settings /W3 Total Cache –Ultimate Guide](#)
8. [TTFB –Time to First Byte –SEO impact](#)
9. [Slow WordPress dashboard–13 remedies](#)
10. [Disable admin ajax calls –Hack](#)

8. Disclaimer



Warranty & Liability disclaimer

All of the preceding tips and recommendations are understood as informational.

Implementation of any of these is at your own risk.