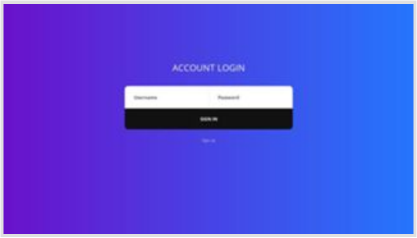
Executive Summary



Performance Report for:

http://tk-naidu.github.io/

Report generated:	Sun, Oct 6, 2019 12:58 PM -0700
Test Server Region:	Vancouver, Canada
Using:	Chrome (Desktop) 75.0.3770.100, PageSpeed 1.15-gt1.2, YSlow 3.1.8

PageSpeed Score	YSlow Score	Fully Loaded Time	Total Page Size	Requests
B (89%) ▲	C (77%) ▲	0.8s ▲	745KB ▲	28 ▲

Top 4 Priority Issues

Leverage browser caching	F (25)	✔ AVG SCORE: 61%	SERVER	HIGH
Minify JavaScript	B (86)	◆ AVG SCORE: 89%	JS	HIGH
Minify CSS	A (99)	◆ AVG SCORE: 96%	CSS	HIGH
Minify HTML	A (99)	◆ AVG SCORE: 98%	CONTENT	LOW

How does this affect me?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, **Google has announced that they are using page speed in their ranking algorithm.**

About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google PageSpeed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

About the Developer

GT.net Performance Hosting

GTmetrix is developed by the good folks at **GT.net**, a Vancouver-based performance hosting company with over 23 years experience in web technology.

https://gt.net/

What do these grades mean?

This report is an analysis of your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The **grades you see represent** how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

What's in this report?

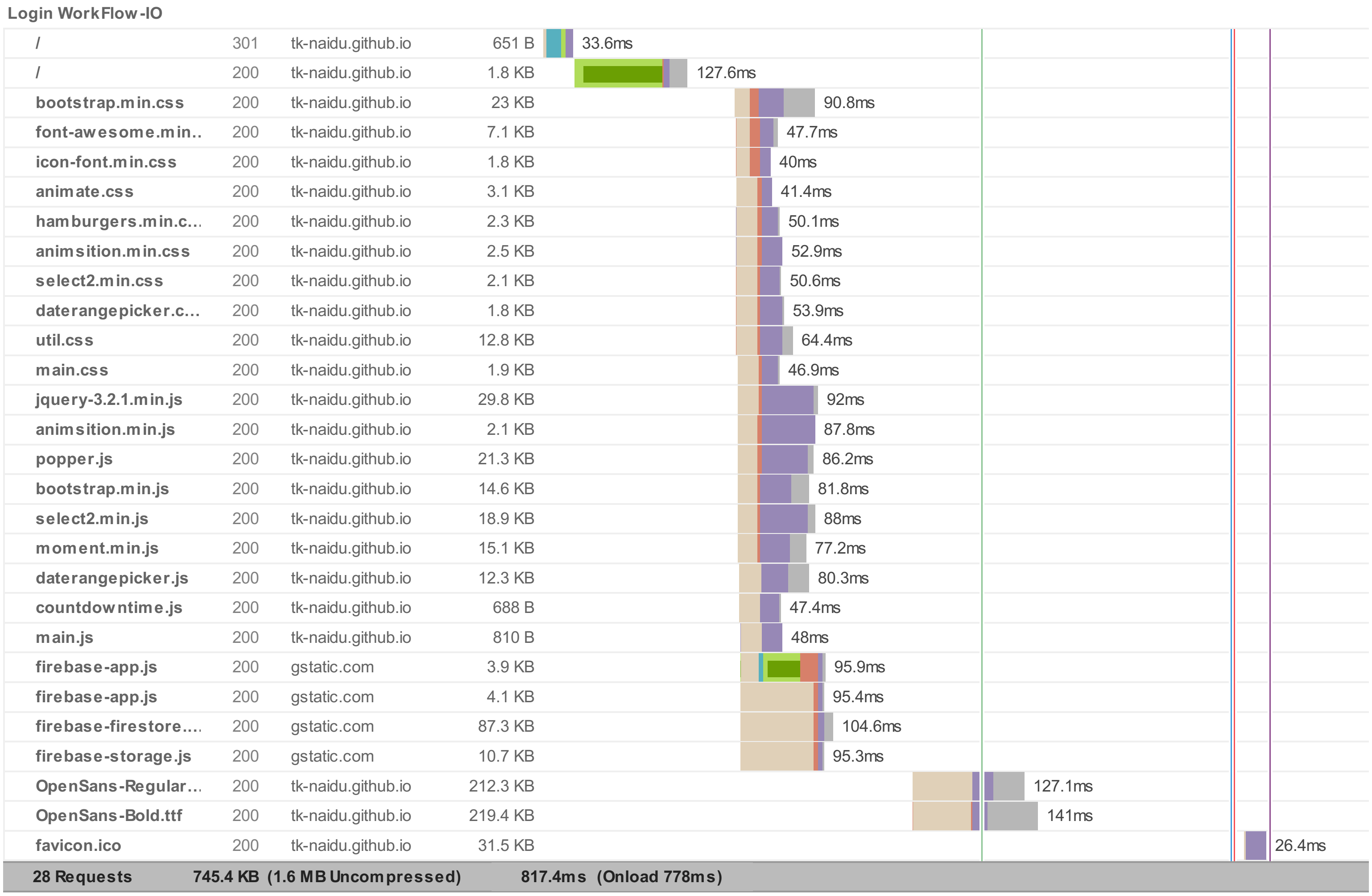
This report covers basic to technical analyses on your page. It is categorized under many headings:

- **Executive:** Overall score information and Priority Issues
- **History:** Graphed history of past performance
- **Waterfall:** Graph of your site's loading timeline
- **Technical:** In-depth PageSpeed & YSlow information

These will provide you with a snapshot of your performance.

Waterfall Chart

The waterfall chart displays the loading behaviour of your site in your selected browser. It can be used to discover simple issues such as 404's or more complex issues such as external resources blocking page rendering.



Login WorkFlow-IO

/	301	tk-naidu.github.io	651 B	33.6ms
/	200	tk-naidu.github.io	1.8 KB	127.6ms
bootstrap.min.css	200	tk-naidu.github.io	23 KB	90.8ms
font-awesome.min..	200	tk-naidu.github.io	7.1 KB	47.7ms
icon-font.min.css	200	tk-naidu.github.io	1.8 KB	40ms
animate.css	200	tk-naidu.github.io	3.1 KB	41.4ms
hamburgers.min.c...	200	tk-naidu.github.io	2.3 KB	50.1ms
animsition.min.css	200	tk-naidu.github.io	2.5 KB	52.9ms
select2.min.css	200	tk-naidu.github.io	2.1 KB	50.6ms
daterangepicker.c...	200	tk-naidu.github.io	1.8 KB	53.9ms
util.css	200	tk-naidu.github.io	12.8 KB	64.4ms
main.css	200	tk-naidu.github.io	1.9 KB	46.9ms
jquery-3.2.1.min.js	200	tk-naidu.github.io	29.8 KB	92ms
animsition.min.js	200	tk-naidu.github.io	2.1 KB	87.8ms
popper.js	200	tk-naidu.github.io	21.3 KB	86.2ms
bootstrap.min.js	200	tk-naidu.github.io	14.6 KB	81.8ms
select2.min.js	200	tk-naidu.github.io	18.9 KB	88ms
moment.min.js	200	tk-naidu.github.io	15.1 KB	77.2ms
daterangepicker.js	200	tk-naidu.github.io	12.3 KB	80.3ms
countdowntime.js	200	tk-naidu.github.io	688 B	47.4ms
main.js	200	tk-naidu.github.io	810 B	48ms
firebase-app.js	200	gstatic.com	3.9 KB	95.9ms
firebase-app.js	200	gstatic.com	4.1 KB	95.4ms
firebase-firestore....	200	gstatic.com	87.3 KB	104.6ms
firebase-storage.js	200	gstatic.com	10.7 KB	95.3ms
OpenSans-Regular...	200	tk-naidu.github.io	212.3 KB	127.1ms
OpenSans-Bold.ttf	200	tk-naidu.github.io	219.4 KB	141ms
favicon.ico	200	tk-naidu.github.io	31.5 KB	26.4ms

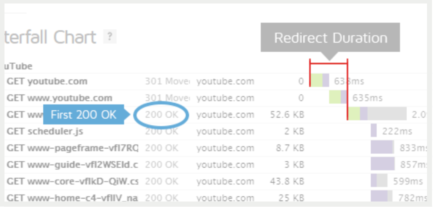
28 Requests **745.4 KB (1.6 MB Uncompressed)** **817.4ms (Onload 778ms)**

Page Load Timings

RUM Speed Index: 493

Redirect 36ms	Connect 101ms	Backend 7ms	TTFB 144ms
First paint 493ms	Contentful paint 493ms	DOM int. 0.8s	DOM loaded 0.8s (0ms)
Onload 0.8s (1ms)			

Redirect duration



uTube

GET youtube.com	301 Move	youtube.com	0	638ms
GET www.youtube.com	301 Move	youtube.com	0	635ms
GET ww	200 OK	youtube.com	52.6 KB	2.0
GET scheduler.js	200 OK	youtube.com	2 KB	222ms
GET www-pageframe-vfl7RQ	200 OK	youtube.com	8.7 KB	833ms
GET www-guide-vfl2WSEld.c	200 OK	youtube.com	3 KB	857ms
GET www-core-vflkD-QiW.cs	200 OK	youtube.com	43.8 KB	599ms
GET www-home-c4-vflIV_na	200 OK	youtube.com	25 KB	782ms

This is the time spent redirecting URLs before the final HTML page is loaded. Common redirects include:

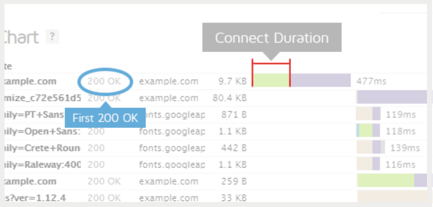
- Redirect from a non-www to www (eg. example.com to www.example.com)
- Redirect to a secure URL (eg. http:// to https://)
- Redirect to set cookies
- Redirect to a mobile version of the site

Some sites may even perform a chain of multiple redirects (eg. non-www to www, then to a secure URL). This timing is the total of all this time that's spent redirecting, or 0 if no redirects occurred.

In the Waterfall Chart, Redirect duration consists of the time from the beginning of the test until just before we start the request of the final HTML page (when we receive the first 200 OK response).

During this time, the browser screen is blank! Ensure that this duration is kept to short by minimizing your redirects.

Connection duration



te

ample.com	200 OK	example.com	9.7 KB	477ms
mize_c72e561d5	200 OK	example.com	80.4 KB	
ily=PT+Sans		nts.googleap	871 B	119ms
ily=Open+Sans:	200	fonts.googleap	1.1 KB	118ms
ily=Crete+Roun	200	fonts.googleap	442 B	139ms
ily=Raleway:400	200	fonts.googleap	1.1 KB	116ms
ample.com	200 OK	example.com	259 B	
s?ver=1.12.4	200 OK	example.com	33 KB	

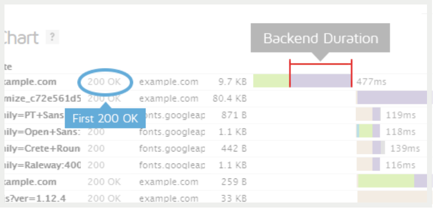
Once any redirects have completed, Connection duration is measured. This is the time spent connecting to the server to make the request to the page.

Technically speaking, this duration is a combination of the blocked time, DNS time, connect time and sending time of the request (rather than *just* connect time). We've combined those components into a single Connection duration to simplify things (as most of these times are usually small).

In the Waterfall Chart, Connection duration consists of everything up to and including the "Sending" time in the final HTML page request (the first 200 OK response).

During this time, the browser screen is still blank! Various causes could contribute to this, including a slow/problematic connection between the test server and site or slow response times from the site.

Backend duration



te

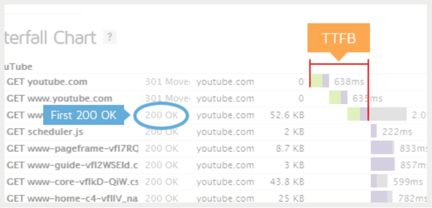
ample.com	200 OK	example.com	9.7 KB	477ms
mize_c72e561d5	200 OK	example.com	80.4 KB	
ily=PT+Sans		nts.googleap	871 B	119ms
ily=Open+Sans:	200	fonts.googleap	1.1 KB	118ms
ily=Crete+Roun	200	fonts.googleap	442 B	139ms
ily=Raleway:400	200	fonts.googleap	1.1 KB	116ms
ample.com	200 OK	example.com	259 B	
s?ver=1.12.4	200 OK	example.com	33 KB	

Once the connection is complete and the request is made, the server needs to generate a response for the page. The time it takes to generate the response is known as the Backend duration.

In the Waterfall Chart, Backend duration consists of purple waiting time in the page request.

There are a number of reasons why Backend duration could be slow. We cover this is our "Why is my page slow" article.

Time to First Byte (TTFB)



uTube

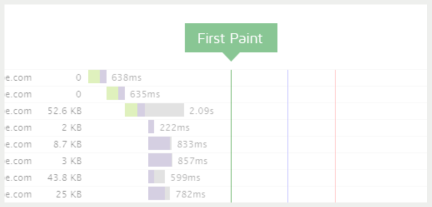
GET youtube.com	301 Move	youtube.com	0	638ms
GET www.youtube.com	301 Move	youtube.com	0	635ms
GET ww	200 OK	youtube.com	52.6 KB	2.0
GET scheduler.js	200 OK	youtube.com	2 KB	222ms
GET www-pageframe-vfl7RQ	200 OK	youtube.com	8.7 KB	833ms
GET www-guide-vfl2WSEId.c	200 OK	youtube.com	3 KB	857ms
GET www-core-vflkD-QiW.cs	200 OK	youtube.com	43.8 KB	599ms
GET www-home-c4-vflIV_na	200 OK	youtube.com	25 KB	782ms

Time to First Byte (TTFB) is the total amount of time spent to receive the first byte of the response once it has been requested. It is the sum of "Redirect duration" + "Connection duration" + "Backend duration". This metric is one of the key indicators of web performance.

In the Waterfall Chart, it is calculated at the start of the test until just before receiving on the page request and represented by the orange line.

Some ways to improve the TTFB include: optimizing application code, implementing caching, fine-tuning your web server configuration, or upgrading server hardware.

First paint time



e.com	0	638ms
e.com	0	635ms
e.com	52.6 KB	2.09s
e.com	2 KB	222ms
e.com	8.7 KB	833ms
e.com	3 KB	857ms
e.com	43.8 KB	599ms
e.com	25 KB	782ms

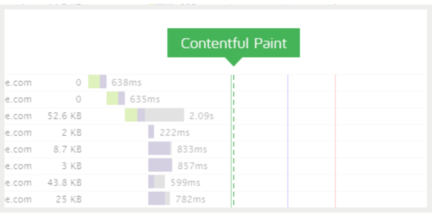
First paint time is the first point at which the browser does any sort of rendering on the page. Depending on the structure of the page, this first paint could just be displaying the background colour (including white), or it could be a majority of the page being rendered.

In the Waterfall Chart, it is represented by the green line.

This timing is of significance because until this point, the browser will have only shown a blank page and this change gives the user an indication that the page is loading. However, we don't know how much of the page was rendered with this paint, so having a early first paint doesn't necessarily indicate a fast loading page.

If the browser does not perform a paint (ie. the html results in an blank page), then the paint timings may be missing.

First contentful paint time



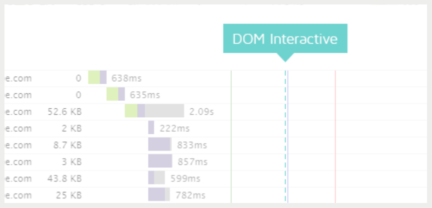
First Contentful Paint is triggered when any *content* is painted - i.e. something defined in the DOM (Document Object Model). This could be text, an image or canvas render.

This timing aims to be more representative of your user's experience, as it flags when actual content has been loaded in the page, and not just any change - but it may often be the same time as First Paint.

Because the focus is on content, the idea is that this metric gives you an idea of when your user receives consumable information (text, visuals, etc) - much more useful for performance assessment than when a background has changed or a style has been applied.

If the browser does not perform a paint (ie. the html results in an blank page), then the paint timings may be missing.

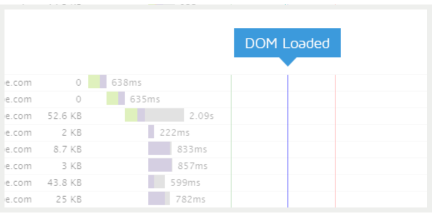
DOM interactive time



DOM interactive time is the point at which the browser has finished loading and parsing HTML, and the DOM (Document Object Model) has been built. The DOM is how the browser internally structures the HTML so that it can render it.

DOM interactive time isn't marked in the Waterfall Chart as it's usually very close in timing to DOM content loaded.

DOM content loaded time



DOM content loaded time (DOM loaded or DOM ready for short) is the point at which the DOM is ready (ie. DOM interactive) and there are no stylesheets blocking JavaScript execution.

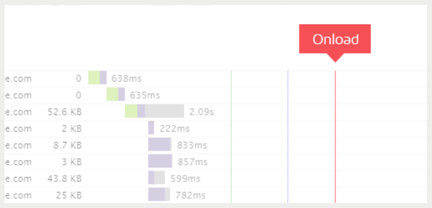
If there are no stylesheets blocking JavaScript execution and there is no parser blocking JavaScript, then this will be the same as DOM interactive time.

In the Waterfall Chart, it is represented by the blue line.

The time in brackets is the time spent executing JavaScript triggered by the DOM content loaded event. Many JavaScript frameworks use this event as a starting point to begin execution of their code.

Since this event is often used by JavaScript as the starting point and delays in this event mean delays in rendering, it's important to make sure that style and script order is optimized and that parsing of JavaScript is deferred.

Onload time



Onload time occurs when the processing of the page is complete and all the resources on the page (images, CSS, etc.) have finished downloading. This is also the same time that DOM complete occurs and the JavaScript window.onload event fires.

Note that there may be JavaScript that initiates subsequent requests for more resources, hence the reason why Fully loaded timing is preferred.

In the Waterfall Chart, it is represented by the red line.

The time in brackets is the time spent executing JavaScript triggered by the Onload event.

Note that Onload time was the previous default for when to stop the test prior to Feburary 8th, 2017.

PageSpeed Recommendations

RECOMMENDATION	GRADE	RELATIVE	TYPE	PRIORITY
Leverage browser caching	F (25)	▼ AVG SCORE: 61%	SERVER	HIGH

Leverage browser caching for the following cacheable resources:

- https://tk-naidu.github.io/css/main.css (10 minutes)
- https://tk-naidu.github.io/css/util.css (10 minutes)
- https://tk-naidu.github.io/fonts/Linearicons-Free-v1.0.0/icon-font.min.css (10 minutes)
- https://tk-naidu.github.io/fonts/font-awesome-4.7.0/css/font-awesome.min.css (10 minutes)
- https://tk-naidu.github.io/images/icons/favicon.ico (10 minutes)
- https://tk-naidu.github.io/js/main.js (10 minutes)
- https://tk-naidu.github.io/vendor/animate/animate.css (10 minutes)
- https://tk-naidu.github.io/vendor/animsition/css/animsition.min.css (10 minutes)
- https://tk-naidu.github.io/vendor/animsition/js/animsition.min.js (10 minutes)
- https://tk-naidu.github.io/vendor/bootstrap/css/bootstrap.min.css (10 minutes)
- https://tk-naidu.github.io/vendor/bootstrap/js/bootstrap.min.js (10 minutes)
- https://tk-naidu.github.io/vendor/bootstrap/js/popper.js (10 minutes)
- https://tk-naidu.github.io/vendor/countdowntime/countdowntime.js (10 minutes)
- https://tk-naidu.github.io/vendor/css-hamburgers/hamburgers.min.css (10 minutes)
- https://tk-naidu.github.io/vendor/daterangepicker/daterangepicker.css (10 minutes)
- https://tk-naidu.github.io/vendor/daterangepicker/daterangepicker.js (10 minutes)
- https://tk-naidu.github.io/vendor/daterangepicker/moment.min.js (10 minutes)
- https://tk-naidu.github.io/vendor/jquery/jquery-3.2.1.min.js (10 minutes)
- https://tk-naidu.github.io/vendor/select2/select2.min.css (10 minutes)
- https://tk-naidu.github.io/vendor/select2/select2.min.js (10 minutes)

Minify JavaScript	B (86)	◆ AVG SCORE: 89%	JS	HIGH

Minify JavaScript for the following resources to reduce their size by 16.7KiB (8% reduction).

- Minifying https://tk-naidu.github.io/vendor/bootstrap/js/popper.js could save 12.0KiB (57% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/daterangepicker/daterangepicker.js could save 4.0KiB (34% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/bootstrap/js/bootstrap.min.js could save 163B (2% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/js/main.js could save 125B (22% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/animsition/js/animsition.min.js could save 114B (6% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/daterangepicker/moment.min.js could save 90B (1% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/jquery/jquery-3.2.1.min.js could save 64B (1% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/select2/select2.min.js could save 63B (1% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/countdowntime/countdowntime.js could save 62B (13% reduction) after compression. See optimized version.
- Minifying https://www.gstatic.com/firebasejs/6.0.2/firebase-app.js could save 30B (1% reduction) after compression. See optimized version.
- Minifying https://www.gstatic.com/firebasejs/6.2.4/firebase-app.js could save 30B (1% reduction) after compression. See optimized version.
- Minifying https://www.gstatic.com/firebasejs/6.0.2/firebase-firestore.js could save 27B (1% reduction) after compression. See optimized version.
- Minifying https://www.gstatic.com/firebasejs/6.0.2/firebase-storage.js could save 25B (1% reduction) after compression. See optimized version.

Minify CSS	A (99)	◆ AVG SCORE: 96%	CSS	HIGH

Minify CSS for the following resources to reduce their size by 2.1KiB (4% reduction).

- Minifying https://tk-naidu.github.io/css/util.css could save 702B (6% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/animate/animate.css could save 323B (12% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/css/main.css could save 301B (18% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/daterangepicker/daterangepicker.css could save 204B (13% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/bootstrap/css/bootstrap.min.css could save 192B (1% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/animsition/css/animsition.min.css could save 135B (7% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/vendor/css-hamburgers/hamburgers.min.css could save 115B (6% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/fonts/font-awesome-4.7.0/css/font-awesome.min.css could save 101B (2% reduction) after compression. See optimized version.
- Minifying https://tk-naidu.github.io/fonts/Linearicons-Free-v1.0.0/icon-font.min.css could save 69B (5% reduction) after compression. See optimized version.

- Minifying https://tk-naidu.github.io/vendor/select2/select2.min.css could save 2B (1% reduction) after compression. See optimized version.

Minify HTML	A (99)	◆ AVG SCORE: 98%	CONTENT	LOW

Minify HTML for the following resources to reduce their size by 219B (16% reduction).

- Minifying https://tk-naidu.github.io/ could save 219B (16% reduction) after compression. See optimized version.

Avoid bad requests	A (100)	◆ AVG SCORE: 98%	CONTENT	HIGH

You scored 100% on this recommendation - nothing to do here!

Avoid landing page redirects	A (100)	◆ AVG SCORE: 98%	SERVER	HIGH

You scored 100% on this recommendation - nothing to do here!

Defer parsing of JavaScript	A (100)	▲ AVG SCORE: 71%	JS	HIGH

You scored 100% on this recommendation - nothing to do here!

Enable gzip compression	A (100)	▲ AVG SCORE: 87%	SERVER	HIGH

You scored 100% on this recommendation - nothing to do here!

Enable Keep-Alive	A (100)	◆ AVG SCORE: 97%	SERVER	HIGH

You scored 100% on this recommendation - nothing to do here!

Inline small CSS	A (100)	◆ AVG SCORE: 96%	CSS	HIGH

You scored 100% on this recommendation - nothing to do here!

Inline small JavaScript	A (100)	◆ AVG SCORE: 95%	JS	HIGH

You scored 100% on this recommendation - nothing to do here!

Minimize redirects	A (100)	▲ AVG SCORE: 89%	CONTENT	HIGH

You scored 100% on this recommendation - nothing to do here!

Minimize request size	A (100)	◆ AVG SCORE: 95%	CONTENT	HIGH

You scored 100% on this recommendation - nothing to do here!

Optimize images	A (100)	▲ AVG SCORE: 71%	IMAGES	HIGH

You scored 100% on this recommendation - nothing to do here!

Optimize the order of styles and scripts	A (100)	◆ AVG SCORE: 95%	CSS/JS	HIGH

You scored 100% on this recommendation - nothing to do here!

Put CSS in the document head	A (100)	◆ AVG SCORE: 100%	CSS	HIGH

You scored 100% on this recommendation - nothing to do here!

Serve resources from a consistent URL	A (100)	▲ AVG SCORE: 90%	CONTENT	HIGH

You scored 100% on this recommendation - nothing to do here!

Serve scaled images	A (100)	▲ AVG SCORE: 70%	IMAGES	HIGH

You scored 100% on this recommendation - nothing to do here!

Specify a cache validator	A (100)	◆ AVG SCORE: 95%	SERVER	HIGH

You scored 100% on this recommendation - nothing to do here!

Combine images using CSS sprites	A (100)	▲ AVG SCORE: 93%	IMAGES	HIGH

You scored 100% on this recommendation - nothing to do here!

Avoid CSS @import	A (100)	◆ AVG SCORE: 98%	CSS	MEDIUM

You scored 100% on this recommendation - nothing to do here!

Prefer asynchronous resources	A (100)	◆ AVG SCORE: 100%	JS	MEDIUM

You scored 100% on this recommendation - nothing to do here!

Specify a character set early	A (100)	◆ AVG SCORE: 100%	CONTENT	MEDIUM

You scored 100% on this recommendation - nothing to do here!

Specify image dimensions	A (100)	◆ AVG SCORE: 98%	IMAGES	MEDIUM

You scored 100% on this recommendation - nothing to do here!

Avoid a character set in the meta tag	A (100)	◆ AVG SCORE: 100%	CONTENT	LOW

You scored 100% on this recommendation - nothing to do here!

Specify a Vary: Accept-Encoding header	A (100)	◆ AVG SCORE: 96%	SERVER	LOW

You scored 100% on this recommendation - nothing to do here!

YSlow Recommendations

RECOMMENDATION	GRADE	RELATIVE	TYPE	PRIORITY
Add Expires headers	F (0)	▼ AVG SCORE: 27%	SERVER	HIGH

There are 19 static components without a far-future expiration date.

- https://tk-naidu.github.io/vendor/bootstrap/css/bootstrap.min.css
- https://tk-naidu.github.io/fonts/font-awesome-4.7.0/css/font-awesome.min.css
- https://tk-naidu.github.io/fonts/Linearicons-Free-v1.0.0/icon-font.min.css
- https://tk-naidu.github.io/vendor/animate/animate.css
- https://tk-naidu.github.io/vendor/css-hamburgers/hamburgers.min.css
- https://tk-naidu.github.io/vendor/animsition/css/animsition.min.css
- https://tk-naidu.github.io/vendor/select2/select2.min.css
- https://tk-naidu.github.io/vendor/daterangepicker/daterangepicker.css
- https://tk-naidu.github.io/css/util.css
- https://tk-naidu.github.io/css/main.css
- https://tk-naidu.github.io/vendor/jquery/jquery-3.2.1.min.js
- https://tk-naidu.github.io/vendor/animsition/js/animsition.min.js
- https://tk-naidu.github.io/vendor/bootstrap/js/popper.js
- https://tk-naidu.github.io/vendor/bootstrap/js/bootstrap.min.js
- https://tk-naidu.github.io/vendor/select2/select2.min.js
- https://tk-naidu.github.io/vendor/daterangepicker/moment.min.js
- https://tk-naidu.github.io/vendor/daterangepicker/daterangepicker.js
- https://tk-naidu.github.io/vendor/countdowntime/countdowntime.js
- https://tk-naidu.github.io/js/main.js

Make fewer HTTP requests	F (28)	◆ AVG SCORE: 31%	CONTENT	HIGH

This page has 13 external Javascript scripts. Try combining them into one.

This page has 10 external stylesheets. Try combining them into one.

Minify JavaScript and CSS	C (70)	◆ AVG SCORE: 72%	CSS/JS	MEDIUM

There are 3 components that can be minified

- https://tk-naidu.github.io/vendor/daterangepicker/daterangepicker.js
- https://tk-naidu.github.io/vendor/countdowntime/countdowntime.js
- https://tk-naidu.github.io/js/main.js

Avoid URL redirects	A (90)	◆ AVG SCORE: 89%	CONTENT	MEDIUM

There is 1 redirect

- http://tk-naidu.github.io/ redirects to https://tk-naidu.github.io/

Compress components with gzip	A (100)	▲ AVG SCORE: 89%	SERVER	HIGH

You scored 100% on this recommendation - nothing to do here!

Use a Content Delivery Network (CDN)	A (100)	▲ AVG SCORE: 27%	SERVER	MEDIUM

Using a CDN YSlow doesn't recognize? Specify your CDNs in your User Settings.

Make AJAX cacheable	A (100)	◆	AVG SCORE: 100%	JS	MEDIUM

You scored 100% on this recommendation - nothing to do here!

Remove duplicate JavaScript and CSS	A (100)	◆	AVG SCORE: 100%	CSS/JS	MEDIUM

You scored 100% on this recommendation - nothing to do here!

Avoid AlphaImageLoader filter	A (100)	◆	AVG SCORE: 99%	CSS	MEDIUM

You scored 100% on this recommendation - nothing to do here!

Avoid HTTP 404 (Not Found) error	A (100)	◆	AVG SCORE: 98%	CONTENT	MEDIUM

You scored 100% on this recommendation - nothing to do here!

Reduce the number of DOM elements	A (100)	▲	AVG SCORE: 91%	CONTENT	LOW

You scored 100% on this recommendation - nothing to do here!

Use cookie-free domains	A (100)	▲	AVG SCORE: 54%	COOKIE	LOW

You scored 100% on this recommendation - nothing to do here!

Use GET for AJAX requests	A (100)	◆	AVG SCORE: 100%	JS	LOW

You scored 100% on this recommendation - nothing to do here!

Avoid CSS expressions	A (100)	◆	AVG SCORE: 99%	CSS	LOW

You scored 100% on this recommendation - nothing to do here!

Reduce DNS lookups	A (100)	▲	AVG SCORE: 70%	CONTENT	LOW

- tk-naidu.github.io: 21 components, 802.9K (175.9K GZip)
- www.gstatic.com: 4 components, 411.4K (107.7K GZip)

Reduce cookie size	A (100)	◆	AVG SCORE: 100%	COOKIE	LOW

You scored 100% on this recommendation - nothing to do here!

Make favicon small and cacheable	A (100)	◆	AVG SCORE: 100%	IMAGES	LOW

You scored 100% on this recommendation - nothing to do here!

Configure entity tags (ETags)	A (100)	▲ AVG SCORE: 94%	SERVER	LOW

You scored 100% on this recommendation - nothing to do here!

Make JavaScript and CSS external	(n/a)		CSS/JS	MEDIUM

Only consider this if your property is a common user home page.

- There is a total of 1 inline script