GiftOfSpeed

Website Speed Test | Tools | Techniques | Optimization Service

# Website speed test.

https://waleedos.github.io/ElwalidELKHABOU_4_21072021/beforeChanges/ Test Now

Test Location:

- [ ] **New York** (USA)
- [ ] **San Francisco** (USA)
- [ ] **Toronto** (Canada)
- [ ] **Singapore** (Singapore)
- [ ] **London** (UK)
- [ ] **Amsterdam** (Netherlands)
- [x] **Frankfurt** (Germany)
- [ ] **Bangalore** (India)

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/

| REQUESTS | TOTAL SIZE | CONTENT VISIBLE ? | FULLY LOADED | OPTIM. SCORE ? |
|---|---|---|---|---|

REQUESTS: **26**
TOTAL SIZE: **3869 kb**

CONTENT VISIBLE: **567 ms**
FULLY LOADED: **926 ms**

| 26 | 3869 kb | 567 ms | 926 ms | 65/100 |
|---|---|---|---|---|

request waterfall

| | | |
|---|---|---|
| beforeChanges | waleedos.github.io | 162 b |
| beforeChanges | waleedos.github.io | 3.0 kb |
| bootstrap.css | waleedos.github.io | 21.5 kb |
| style.css | waleedos.github.io | 3.9 kb |
| font-awesome.css | waleedos.github.io | 7.1 kb |
| et-line.css | waleedos.github.io | 1.4 kb |
| jquery-2.1.0.js | waleedos.github.io | 34.8 kb |
| bootstrap.js | waleedos.github.io | 10.9 kb |
| blocs.js | waleedos.github.io | 3.1 kb |
| jquery.touchSwipe.js | waleedos.github.io | 5.6 kb |
| gmaps.js | waleedos.github.io | 14.2 kb |
| la-chouette-agence.png | waleedos.github.io | 26.6 kb |
| logo.png | waleedos.github.io | 24.4 kb |
| title.png | waleedos.github.io | 9.4 kb |
| citation.png | waleedos.github.io | 11.3 kb |
| title2.png | waleedos.github.io | 7.8 kb |
| 1.jpg | waleedos.github.io | 266.5 kb |
| 2.jpg | waleedos.github.io | 106.7 kb |

| | | |
|---|---|---|
| 3.bmp | waleedos.github.io | 166.4 kb |
| 4.bmp | waleedos.github.io | 219.1 kb |
| la-chouette-agence-banniere.jpg | waleedos.github.io | 600.2 kb |
| texture-paper.png | waleedos.github.io | 93.9 kb |
| image-de-presentation.bmp | waleedos.github.io | 2101.2 kb |
| lines-h2-bg.png | waleedos.github.io | 1023 b |
| et-line.woff | waleedos.github.io | 53.9 kb |
| fontawesome-webfont.woff2?v=4.7.0 | waleedos.github.io | 75.4 kb |

26 requests 3869 kb 926 milliseconds

## size breakdown

| | | |
|---|---|---|
| image | 93.9% | 3634 kb |
| font | 3.3% | 129 kb |
| javascript | 1.8% | 69 kb |
| css | 0.9% | 34 kb |
| html | 0.1% | 3 kb |
| redirect | 0.0% | 162 b |
| | 100% | 3869 kb |

## request breakdown

| | | |
|---|---|---|
| image | 50.0% | 13 |
| javascript | 19.2% | 5 |
| css | 15.4% | 4 |
| font | 7.7% | 2 |
| html | 3.8% | 1 |
| redirect | 3.8% | 1 |
| | 100% | 26 |

## slowest local resources

| | load time |
|---|---|
| https://waleedos.github.io/ElwalidELKHABOU_4_21072021/beforeChanges/img/image-de-presentation.bmp | 0.3s |
| https://waleedos.github.io/ElwalidELKHABOU_4_21072021/beforeChanges/img/la-chouette-agence-banniere.jpg | 0.2s |

| slowest local resources | load time |
|---|---|
| https://waleedos.github.io/ElwalidELKHABOU_4_21072021/beforeChanges/img/texture-paper.png | 0.1s |
| https://waleedos.github.io/ElwalidELKHABOU_4_21072021/beforeChanges/img/lines-h2-bg.png | 0.1s |
| https://waleedos.github.io/ElwalidELKHABOU_4_21072021/beforeChanges/fonts/et-line.woff | 0.1s |

Show More

| slowest external resources | load time |
|---|---|

## performance metrics

### combine js files

**offending resources:**

waleedos.github.io
- https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/jquery-2.1.0.js
- https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/bootstrap.js
- https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/blocs.js
- https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/jquery.touchswipe.js
- https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/gmaps.js

if possible, js files under 100kb in size should always be combined

### combine css files

**offending resources:**

waleedos.github.io
- https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/css/bootstrap.css
- https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/style.css
- https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/css/font-awesome.css
- https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/css/et-line.css

if possible, css files under 100kb in size should always be combined

### minimize js files

#### offending resources:

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/jquery-2.1.0.js
∟over 30.52% can be saved

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/bootstrap.js
∟over 19.34% can be saved

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/blocs.js
∟over 12.60% can be saved

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/jquery.touchswipe.js
∟over 21.57% can be saved

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/js/gmaps.js
∟over 17.64% can be saved

### avoid query strings in urls

#### offending resources:

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/fonts/fontawesome-webfont.woff2?v=4.7.0

### minify html

#### offending resources:

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/
∟5.87 can be saved%

### minimize css files

### offending resources:

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/css/bootstrap.css
∟over 18.89% can be saved

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/style.css
∟over 29.14% can be saved

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/css/font-awesome.css
∟over 19.94% can be saved

https://waleedos.github.io/elwalidelkhabou_4_21072021/beforechanges/css/et-line.css
∟over 21.60% can be saved

## keep total size of css small

### perfect score!

total combined size is 212 kb

## number of total http requests should not exceed 500

### perfect score!

## character set should be specified

### perfect score!

## avoid bad requests

### perfect score!

## keep server response time low

perfect score!

avoid too many paralell downloads from the same domain

perfect score!

serve resources from a consistent url

perfect score!

keep total size of inlined css and javascripts small

perfect score!

do not load more than 10 prefetched requests

perfect score!

enable compression (gzip/brotli)

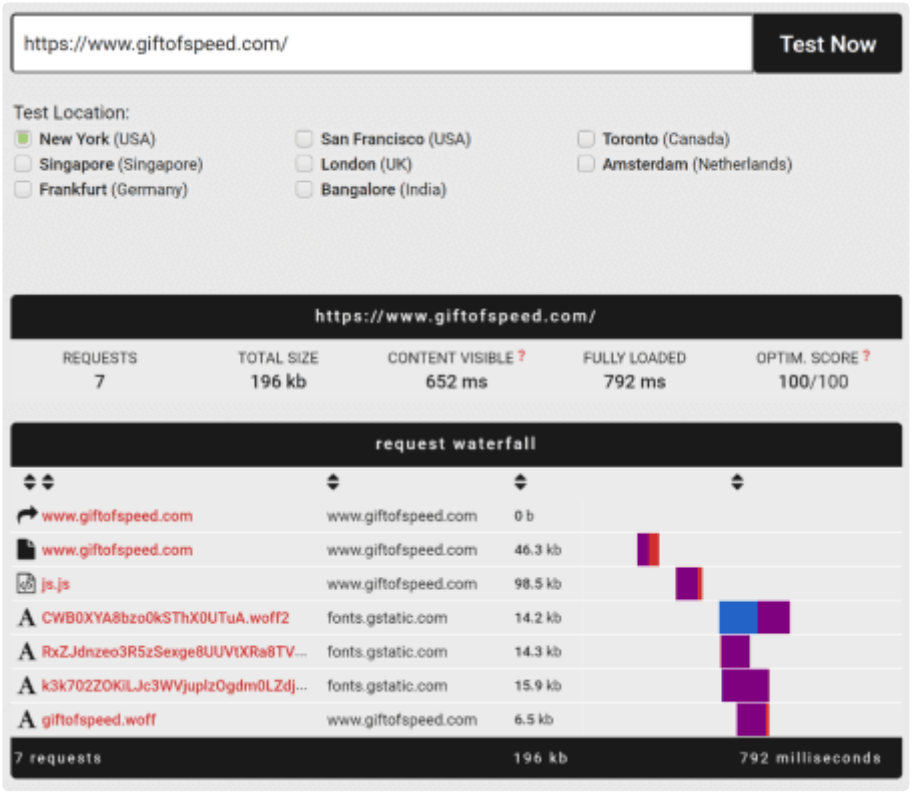
perfect score!

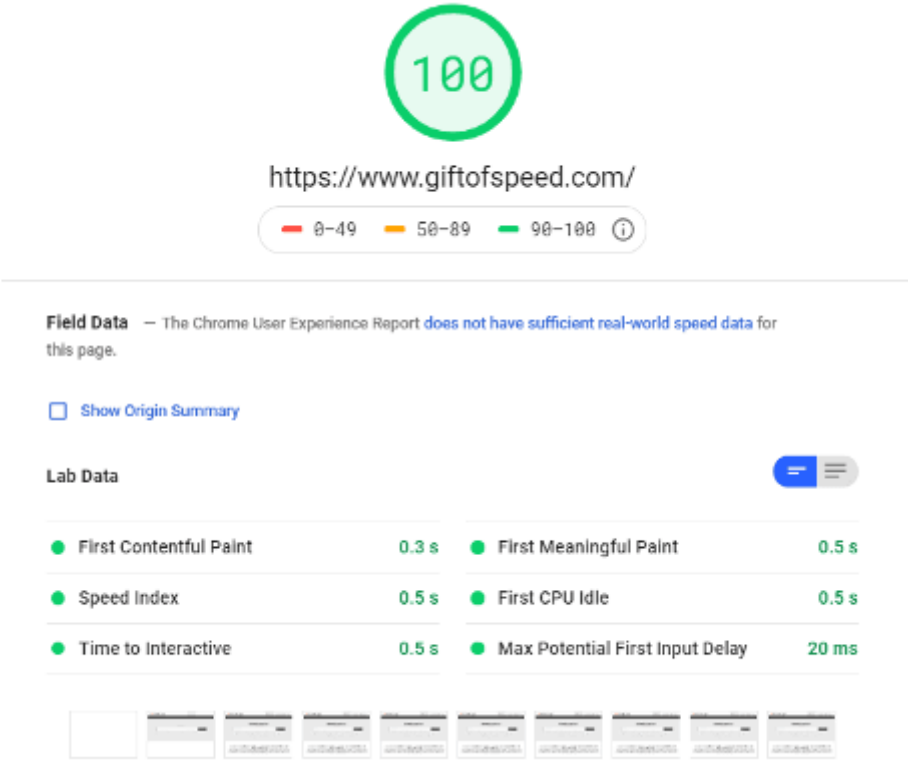
replace or remove slow loading resources

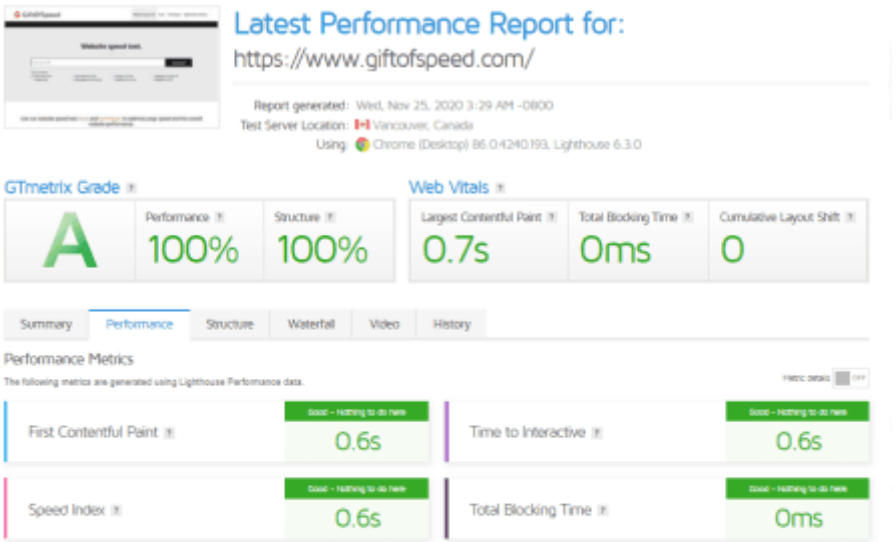
perfect score!

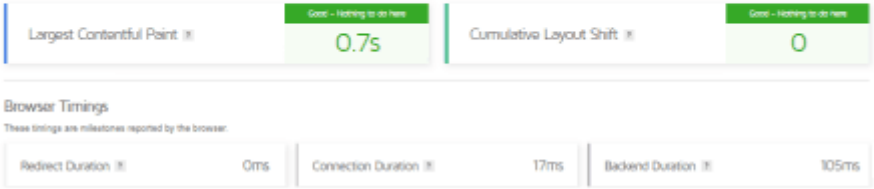
# Let Us Optimize Your Website's Speed

Do you want to achieve the fastest page load times possible for your website? We can help you with that! With years of experience we know all the ins and outs of how to get the maximum performance out of any website. Get a free page speed audit of your website and a price quote for our website speed optimization services now. Look below to see what we've done for our own website:


https://www.giftofspeed.com/
Test Now
Test Location:
New York (USA)
San Francisco (USA)
Toronto (Canada)
Singapore (Singapore)
London (UK)
Amsterdam (Netherlands)
Frankfurt (Germany)
Bangalore (India)
https://www.giftofspeed.com/
REQUESTS 7
TOTAL SIZE 196 kb
CONTENT VISIBLE 652 ms
FULLY LOADED 792 ms
OPTIM. SCORE 100/100
request waterfall
www.giftofspeed.com www.giftofspeed.com 0 b
www.giftofspeed.com www.giftofspeed.com 46.3 kb
js.js www.giftofspeed.com 98.5 kb
CWB0XYA8bzo0kSThX0UTuA.woff2 fonts.gstatic.com 14.2 kb
RxZJdnzeo3R5zSexge8UUVtXRa8TV... fonts.gstatic.com 14.3 kb
k3k702ZOKiLJc3WVjuplzOgdm0LZdj... fonts.gstatic.com 15.9 kb
giftofspeed.woff www.giftofspeed.com 6.5 kb
7 requests 196 kb 792 milliseconds


100
https://www.giftofspeed.com/
0-49 50-89 90-100
Field Data — The Chrome User Experience Report does not have sufficient real-world speed data for this page.
Show Origin Summary
Lab Data
First Contentful Paint 0.3 s
First Meaningful Paint 0.5 s
Speed Index 0.5 s
First CPU Idle 0.5 s
Time to Interactive 0.5 s
Max Potential First Input Delay 20 ms


Latest Performance Report for:
https://www.giftofspeed.com/
Report generated: Wed, Nov 25, 2020 3:29 AM -0800
Test Server Location: Vancouver, Canada
Using: Chrome (Desktop) 86.0.4240.193, Lighthouse 6.3.0
GTmetrix Grade
A
Performance 100%
Structure 100%
Web Vitals
Largest Contentful Paint 0.7s
Total Blocking Time 0ms
Cumulative Layout Shift 0
Summary Performance Structure Waterfall Video History
Performance Metrics
First Contentful Paint 0.6s
Time to Interactive 0.6s
Speed Index 0.6s
Total Blocking Time 0ms




# How To Optimize A Website's Performance.

Learn more about how to fully optimize the speed of a website by reading about the free page speed techniques below.

## Enable gzip Compression

Reduce the size of web files served from a server by an average of 50-70%.

## Optimize CSS delivery

Optimize CSS delivery for faster page rendering by inlining, defer loading,

## Defer Load CSS

Defer load CSS scripts to render web pages quicker.

## Leverage Browser Caching

Leverage browser caching to speed up your website. Learn about the

compressing and learning what, and what not to do.

methods that allow you to enable caching server side and client side.

## Defer Load JavaScript

Defer load JavaScript files to improve page load times.

## Inline CSS Scripts

Instantly render the critical CSS by calling it from the HTML head. Avoid render-blocking CSS files.

## Make Fewer HTTP Requests

Make fewer HTTP requests to minimize parallel downloads by reducing the number of files a web page needs to render a page.

## Use Less JavaScript

Learn how to detect and remove JavaScript that a web page doesn't necessarily need to function correctly.

## Lazy Load Images

## Optimize Images

## Reduce The Amount Of Functional Images

Reduce the amount of image files a web page is loading by combining or replacing them.

## Optimize The Critical Rendering Path

Optimize the critical rendering path to speed up the initial above-the-fold view visitors see when loading a web page.

Lazy load images by only loading them when the visitor is about to view them. This speeds up the loading of the above-the-fold content.

Optimize images by reducing their file size to a bare minimum without losing image quality.

## How To Speed Up WordPress

Optimize a WordPress website by using various plugins, tricks and methods.

## Fix Broken Requests

Detect and fix all broken links, images and other files to improve performance. Broken requests can slow your website down.

## Choose The Right Type Of Hosting

Which type of hosting is best for performance? Shared, VPS, dedicated or another type?

## How To Speed Up Apache

Learn how to speed up an Apache server by tweaking its settings and using free applications.

## Inline JavaScript

Inline (smaller) JavaScript to improve page load times.

## Avoid Use CSS @import

Avoid using CSS @import to load external CSS files to avoid slowing a web page down.

## Load Scripts Asynchronously

Load scripts asynchronously to improve page load times.

## Avoid JavaScript Libraries

Avoid loading big JavaScript libraries like Jquery for website functionalities when possible.

## Make Use Of a CDN

Use a Content Delivery Network to achieve the fastest response and download times.

## Enable keep-alive

Make sure keep-alive is enabled to allow multiple browser connections without using multiple TCP connections.

## Avoid Redirects

Avoid using unnecessary redirects, stop them from slowing your website down.

# Improve Page Speed. Use One of Our Free Tools.

Use one of the below free tools to improve the performance of your website.

## CSS Optimization Test

Analyze a website's CSS for performance.

## CSS Compressor

Minimize CSS scripts to improve page speed.

## JavaScript Optimization Test

Analyze if JavaScript is being optimally delivered on a website.

## JavaScript Compressor

Minify JavaScript to maximize performance.


## Gzip / Brotli Compression Test

Test whether Gzip or Brotli compression is enabled on your website.

## Image Optimization Test

Test if images being loaded on your website can be optimized.

## PNG Compressor

Reduce the file size of PNG images while keeping the image quality.

## JPEG Compressor

Adjust the quality and/or size of JPEG images to reduce their file size.

## CSS Sprites Generator

Save multiple images to a single image, resulting in fewer HTTP requests.

## Caching Test

Check if and how all the files loaded on your website are being properly cached.

## Broken Links/Requests Test

Test a web page for broken links and requests.

## HTTP Requests Checker

How many HTTP requests does a web page make?

## Base64 Encoder

## Keep-Alive Checker

## Line Breaks Remover

## HTTP Header Checker

Encode web files to a Base64 string to reduce the number of HTTP requests.

Check whether a website has keep-alive enabled.

Remove line breaks from scripts to reduce their size.

Check the HTTP server header of a web page.