

## Featured Posts

[How to do Effective Web Development](#)
[Common Web Application Threats](#)
[Common Hacking Techniques](#)
[View All](#)

SHARE  
ON

# Common Website Optimization Techniques

 March 22, 2019

[f](#) [p](#) [t](#) [in](#) [t](#)

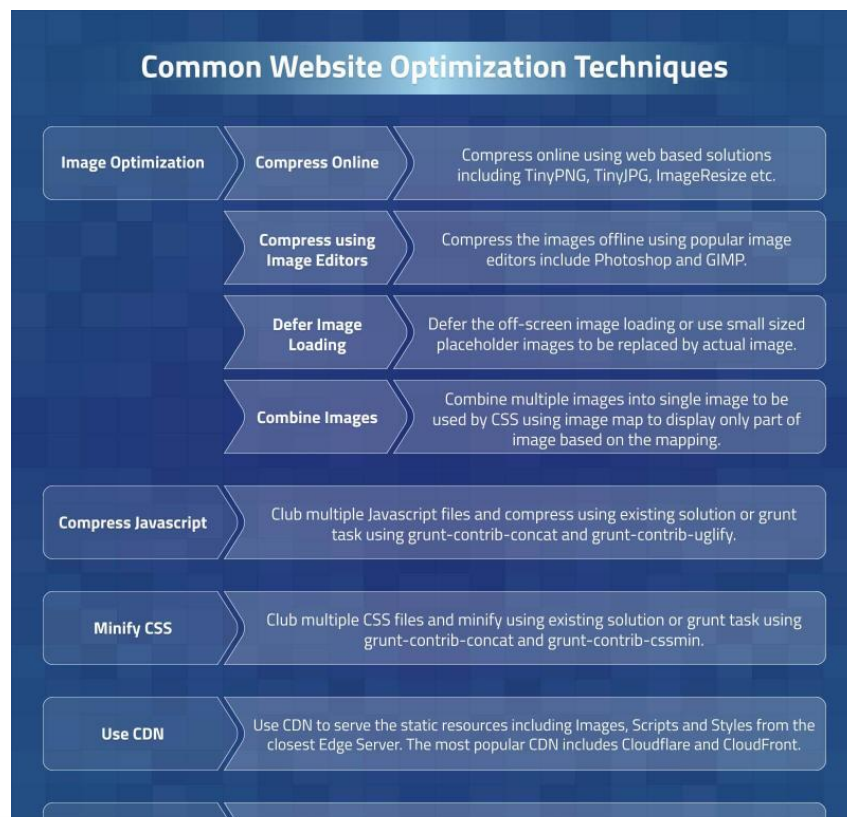
## Recent Posts



What's New in PHP 7.4



What's New in Laravel



SHARE  
ON

**Gzip Compression** Use Gzip Compression on server side to transfer the compressed resources including HTML, Javascript and CSS.

**Use HTTP Keep-Alive Headers** Use HTTP Keep-Alive Headers to reduce network overhead by keeping the same TCP connection open to serve multiple resources.

**Minimize Redirects** Minimize the HTTP redirects as much as possible to serve the page on first request itself.

**Hotlink Protection** Protect static resources embedded in web pages of other sites to reduce bandwidth consumption by other websites.

**Defer Request** Defer requests to 3rd-party resources where possible and call only when it's required.

**Avoid 404 Errors** Analyze your website pages and remove the 404 errors to speed up the page loading.

**Switch To Good Hosting Solution** Sometimes using a shared server under load might slow down the website. A good hosting provider or using dedicated server will be more ideal in such cases.

**Database Optimization** Optimize the database queries and use proper indexing to reduce the time taken by it to fetch the results.

**Use Prefetching Techniques** Use Prefetching techniques including DNS Prefetching, Link Prefetching and Prerendering.

**TECHNOLUSH**  
www.technolush.com

This post provides a list of most common optimization techniques that can be used to speed up the website, hence resulting in a fast loading web page. The faster loading pages have an impact on search engine ranking, hence must be considered as part of the web development process.

## Image Optimization Techniques

Images play a major role in almost all websites. The good looking images bring life to a web page if used properly. At the same time, the images contribute to the network overhead since these add size drastically to the overall page size impacting the page load time. Below listed are the ways to effectively use images on a web page.

**Compress using online solutions** - We can always use the available online solutions to compress the images. These include TinyPNG, Online Image Compressor, and Minify Image.



What's New In Java 13



Differences Between AR, VR, and MR



Types of Digital Marketing

**Defer the Image** - Defer the image loading following appropriate technique suitable and possible in your scenario. We can defer the loading of image and download and show it only when the part of the webpage is visible for the first time. This will definitely improve the web page performance since we do lazy loading of images only when required.

```
<!-- SEO Friendly - Lazy Loading -->  
  
<noscript></noscript>
```

SHARE  
ON

**Combine Images** - We can also combine images accessed by the CSS and use the image maps to show the part of the image. This will reduce the total requests count, resulting in low network overhead.

## Compress Javascript

We must prioritize and club together the Javascript files according to the web page requirements. The clubbed files must be minified using either online solutions or develop own solution as explained on [Tutorials24x7](#). In case the website is developed using the framework, the framework might provide a solution out of the box. We can also use node and write the grunt tasks using the gruntfile in order to club and compress using available npm packages **grunt-contrib-concat** and **grunt-contrib-uglify**.

## Compress CSS

Similar to Javascript, the CSS files can also be clubbed together and compressed in order to reduce the number of requests and overall size of the CSS. The node packages to be used to club and minify the CSS files include **grunt-contrib-concat** and **grunt-contrib-cssmin**.

## Use CDN

Using standard CDN services like Cloudflare and CloudFront drastically improve the website performance by serving the static content from the closest Edge server, hence reducing Network overhead and also enhance server performance. Once the CDN service cache the static assets based on

using the Web Server. On top of it, Cloudflare is 100% free service for basic usage. The only issue with Cloudflare is that it needs control of DNS record, but also prevent common network attacks like Denial Of Service(DDoS).

## Minimize HTTP Request

We have already discussed the clubbing of CSS and Javascript in previous sections to reduce the request count. We can also reduce other HTTP requests by further analyzing the total requests triggered by the web page. The most common way to reduce the HTTP requests is by using effective cache strategy which can reduce the subsequent requests of the resource if it is cached by the Browser.

SHARE  
ON

## Cache Optimization

Caching can take place at both Server side and Client side to optimize the web page. The server-side caching can generate the web page and cache it for further requests. The server-side cache can improve the overall performance of the web page by reducing the CPU usage and database queries.

The client-side cache works according to the caching rules specified by the web server. Accordingly, the browsers can maintain a local cache and use the cached resources instead of getting it multiple times from the server.

Use **Expires Header** where possible for effective client-side caching.

## Gzip Compression

Add gzip Compression on the server side where possible to reduce the overall size of static resource transferred over the network. We can compress the HTML and CSS files with Gzip Compression to save around approx 50% of the size transferred over the network.

## HTTP Keep-Alive Headers

Enable the HTTP Keep-Alive Headers with limitations to serve multiple requests over the same connection. Though it's an effective way to keep network overhead low, use with precautions in a limited way.

## Minimize Redirects

speed of the website.

## Hotlink Protection

Protect your website static resources especially images to be hot-linked by other websites. The hotlinks embed the images from your server to other websites which consume the server bandwidth, hence brings additional network bandwidth consumption by other websites.

## Defer Request

SHARE  
ON

We might be using several 3rd-party services or integrations with the main website. We can defer loading the resources from these solutions and load them only when required instead of loading them all at the start.

## Remove 404 Errors

The 404 errors keep the browser blocking for some time in order to get the resources which are actually not there resulting in 404 error.

## Good Hosting Provider

Sometimes, a bad server provided by the hosting provider results in a slow website. We might consider using a better hosting provider or use a dedicated server from well-known providers where possible to generate the web pages faster.

## Database Optimization

It is a must-have activity for websites heavily dependent on the database to generate web pages. Inefficient queries keep database busy in finding out the results, hence adds additional time to serve the requests. The most common approach to avoid it to use proper indexes so that the database find the required data in less time.

## Use Prefetching Techniques

Use prefetching techniques to pre-load the resources before the user triggers it by analyzing resource utilization. The common prefetching techniques are as listed below.

Prerendering - Keep the page ready in advance.

## Write a Comment

SHARE  
ON



[Click on the captcha image to get new code.](#)

**Submit**

## Discussion Forum by DISQUS

LOG IN WITH

OR SIGN UP WITH DISQUS 

SHARE  
ON

Be the first to comment.

## Links

[Testimonial](#)[Feedback](#)[Contact Us](#)[Blog Search](#)[Terms & Conditions](#)[Privacy Policy](#)

## Featured Posts



## Newsletter

Sign Up to receive updates from us.

[Join Us](#)

Copyright © 2017 - 2019 TechnoLush. All Rights Reserved.