**University of Waterloo**

**Faculty of Mathematics**

**Modern Approaches to Front-End Performance Enhancement**

**TD Securities**

**Equity Trading Team**

**Toronto, Ontario, Canada**

**Prepared By**

**Yizhi Cui**

**20602253**

**3B Computer Science**

**August 10th , 2018**

# Letter of Submittal

**To: Yan Jun An**

**From: Yizhi Cui**

**Date: 2018/08/15**

**Re: Work Report: Modern Approaches to Front-End Performance Enhancement**

Dear Yan Jun An,

I have prepared the enclosed report, “Modern Approaches to Front-End Performance Enhancement”, for my 3B work term report and for TD Securities. This report, the fourth of four work reports that the Co-operative Education Program requires that I successfully complete as part of my BCS Co-op degree requirements, has not received academic credit.

The TD Securities Equity Trading team that you lead provides Report Portal and various report templates for businesses and institutions to visualize data in the form of excel report. My job as Software Developer required that I complete implementations of and fix bugs in reports inside the Report Portal. I learned advanced SQL technologies, various Java design patterns. This report is an in-depth analysis of the importance and strategies of improving the response time, one important matrix of a website’s performance.

The Faculty of Mathematics requests that you evaluate this report for command of topic and technical content/analysis. Following your assessment, the report, together with your evaluation, will be submitted to the Math Undergrad Office for evaluation on campus by qualified work report markers. The combined marks determine whether the report will receive credit and whether it will be considered for an award.

Thank you for your assistance in preparing this report.

Sincerely,

Yizhi (Fred) Cui

Table of Contents

[Executive Summary 4](#_Toc522134540)

[1 Introduction 5](#_Toc522134541)

[1.1 History and Studies of Judging a Webpage’s Performance 5](#_Toc522134542)

[1.2 Modern Views on Response Time 6](#_Toc522134543)

[2.0 Analysis 7](#_Toc522134544)

[2.1 HTML/CSS Resources 7](#_Toc522134545)

[2.1.1 Resource Minification 7](#_Toc522134546)

[2.1.2 Resource Declaration Ordering 8](#_Toc522134547)

[2.1.3 Non-Blocking CSS Resources 9](#_Toc522134548)

[2.2 JavaScript 9](#_Toc522134549)

[2.2.1 Remove Render-Blocking Javascripts 9](#_Toc522134550)

[2.2.2 Async or Defer Loading 10](#_Toc522134551)

[3 Conclusion 11](#_Toc522134552)

[References 12](#_Toc522134553)

# Executive Summary

Nowadays with more and more people accessing the internet, web applications are becoming more and more popular among developers. However, performance of a website is closely related to the success of a web application because users will always prefer websites with better looks and shorter response times.

Moreover, Google, the most popular search engine, judges page load speed as a factor of the website’s pagerank. In other words, slower-loading pages are less likely to get visits from browsers than faster-loading pages with similar contents.

This report will introduce and analyze the development strategies of enhancing front-end performance of a web application by reducing waiting time, focusing on HTML, CSS and JavaScript’s resource management and declaration.

# 1 Introduction

## 1.1 History and Studies of Judging a Webpage’s Performance

As early as 1994, usability of information systems was defined to be equivalent to a set of design principles, and one of the five key elements is response time (download delay), which focuses on the speed with which the system provided a response to user activity (Nielsen, 1994).

From 1997 to 2000, Three studies were conducted to research the importances of the five elements and download delay were one of the most crucial factors of a website’s usability according to regression analysis, as the alpha-value is almost 1. Download delay was also found to be one key factor of a website’s success (Jonathan W.Palmer, 2002).

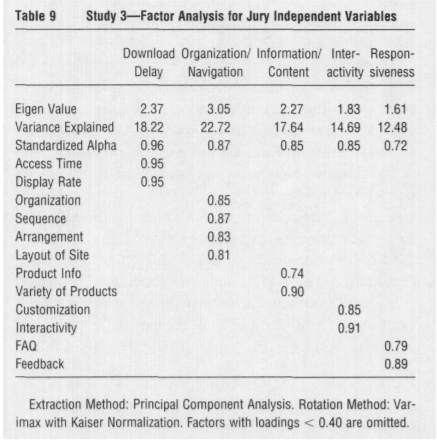


Image 1.1 Result of Study Showing Influence Factors of Website’s Usability

## 1.2 Modern Views on Response Time

As time went on, more and more researches and theories were done to seek for the way to build the most user-friendly website. According to Gardner, Figure A shows the time users are willing to spend waiting for a web page to load. According to surveys conducted by Akamai and Gomez.com, 40 percent of the users who go to a website will abandon it if page load time exceeds three seconds (Gardner, B.S. (2011)).

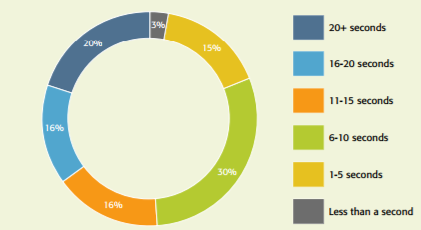


Image 1.2 Correlating page load time and user patience. Most users will wait only 6 to 10 seconds for a site to load. (Data from <http://blog.kissmetrics.com/loading-time>.)

# 2.0 Analysis

Typically, the process of a webpage being loaded consists of four parts. Load HTML formatted file, load required resources for rendering, construct rendering tree (DOM Tree and CSSOM Tree), render, load additional resources. The webpage’s visual content is loaded once the rendering process is done.

In this section we are going to focus on how developers can improve the performance of a website by reducing the time for the first three section.

## 2.1 HTML/CSS Resources

### 2.1.1 Resource Minification

When we develop our websites we often try to make it more human readable by writing a lot of comments, leaving white space and naming the variables with longer and better understandable names for human.

Removing all unnecessary spaces, breaks and including comments that are used to help other developers understand will significantly cut down your HTML file’s size and speed up your site's page load times and furthermore lighten the download for your user. The word ‘Minifying’ stands for this process.

Using minification we can also reduce the bandwidth consumption, what results in smaller energy consumptions for the servers and in the long-term scale it will reflect in smaller running costs (MINÁRIK, Daniel, 2017).

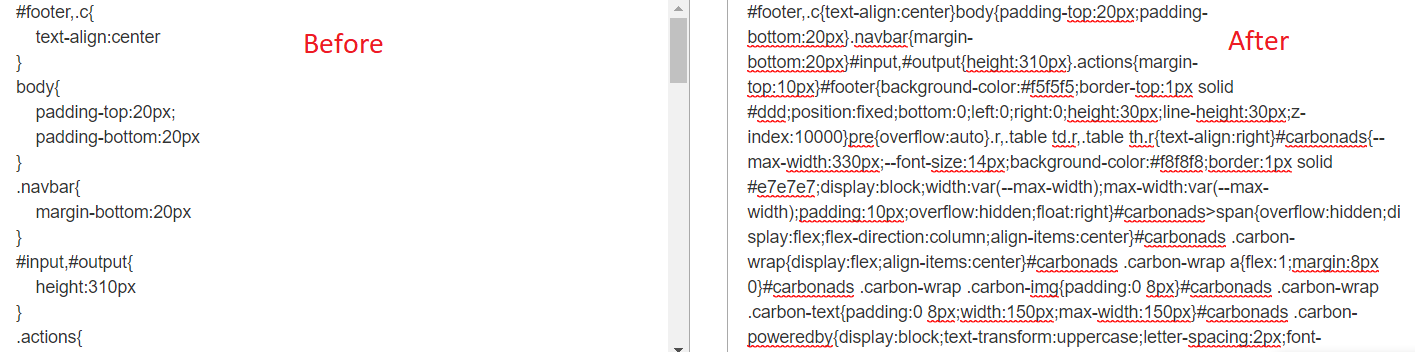


Image 2.1 CSS Minification: Before and After

### 2.1.2 Resource Declaration Ordering

Summary: Parallel downloading can be used by the browser by placing CSS tags before any JavaScript tags, which speeds up the browser’s rendering process.

Webpage Loading: When the browser goes to a website (url), it fetches the HTML formatted file first, then it starts asking for all the resources the HTML needs from the server (css, images, scripts, etc.) before rendering. After all css files are loaded, the browser builds the CSSOM (CSS Object Model) which helps rendering project.

Parallel downloading: Resources can be downloaded synchronously. Image 2.2 demonstrates an example of parallel downloading. Five requests, each takes around 77 ms, can be fetched within 100 ms. Whereas, if they were downloaded asynchronously, will take up around 400 ms to load.



Image 2.2 Example of Parallel Downloading CSS Style Sheets

Thus we can see that parallel downloading can increase a website’s loading time significantly, and this not only apply to initial load, but also affects page redirection.

According to Introduction to the CSS Object Model from Sexton, while stylesheets are parallel downloaded, script loading often blocks the loading of CSS, so placing all script resource declarations before CSS declarations can make most use of parallel downloading, therefore speeds up the CSSOM construction, the browser can start the rendering process earlier.

### 2.1.3 Non-Blocking CSS Resources

CSS files can block the page load and delay the rendering of your page.

Preloading allows developers to write fetch requests in the HTML header, specifies the requests that are needed very soon after load. Those resources if specified as ‘preload’, will start loading before the web page is being rendered.

According to *mozilla.org*, using preload can load the CSS files before the browser starts showing the content of the page. Preloading can also help developers prioritize resource loading.

## 2.2 JavaScript

### 2.2.1 Remove Render-Blocking Javascripts

During the rendering process of browser, whenever a script block is encountered, the browser will stop everything and start trying to execute the script. It will wait for the script to be downloaded if the script is external and may incur one or more network round trips and delay the time to first render the webpage.

So, developers should include external script declarations together in the header instead of in the body and only have lightweight inline script blocks that are critical or enhances performance, as inline scripts are render-blocking.

### Async or Defer Loading

Async attribute of a script element in HTML allows the script to be downloaded asynchronously while rendering. According to Google, by default javascript blocks DOM construction, which is an important part of the rendering process, and delays the time to first render. However, async scripts are not guaranteed to be executed before rendering or to be executed in the order they are declared.

This can help reduce loading speed significantly if the scripts referenced are not required to run before DOM construction.

Defer loading a script element in HTML means only start loading the script resource after the page content is loaded.

Comparing to async loading, this method does not necessarily reduce more time on a web page’s content loading, but can significantly reduce the waiting time on the completion of rendering.

# 3 Conclusion

Studies and reports indicate that a webpage’s loading time is crucial to the performance of it and is an important factor of user experience. The websites with better user experience usually attract more users than the others. Thus improving the loading speed should be a prioritized job for front-end web developers and if the strategies mentioned in this report are followed and used in web development, there will be significant performance improvement in the webpage’s loading speed, overall performance and popularity.

# References

Gardner, B. S. (2011). Responsive web design: Enriching the user experience. *Sigma Journal: Inside the Digital Ecosystem*, *11*(1), 13-19.

MINÁRIK, Daniel. *Accelerated Mobile Pages*. Diss. Masarykova univerzita, Fakulta informatiky, 2017.

Nielsen, Jakob. *Usability engineering*. Elsevier, 1994.

Palmer, Jonathan W. "Web site usability, design, and performance metrics." *Information systems research* 13.2 (2002): 151-167.