DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

EVALUATING AND IMPROVING PERFORMANCE OF WEB APPLICATIONS

by

Okan DAVUT

July, 2021

İZMİR

EVALUATING AND IMPROVING PERFORMANCE OF WEB APPLICATIONS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Master of
Science in Computer Engineering

by
Okan DAVUT

July, 2021
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"EVALUATING AND IMPROVING PERFORMANCE OF WEB APPLICATIONS"** completed by **OKAN DAVUT** under supervision of **ASST. PROF. DR**. **ÖZLEM AKTAŞ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

.....

Asst. Prof. Dr. Özlem AKTAŞ

Supervisor

.....

Asst.Prof.Dr.Kökten Ulaş BİRANT

(Jury Member)

.....

Assoc. Prof. Dr. Akın ÖZÇİFT

(Jury Member)

Prof. Dr. Özgür ÖZÇELİK

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First of all, I would like to thank to my academic advisor, Asst.Prof.Dr. Özlem AKTAŞ, for her guidance and useful advice that always show me the right direction in my research. During my research, she contributed to a rewarding graduate school experience by giving me intellectual freedom in my study, supporting my attendance at various conferences, and demanding high-quality work in all my efforts.

I would also like to thank my wife, *Şeyda,* and my mom *Şerife*, who supported me with love and understanding. Without you, I could never have reached this current level of success.

Okan DAVUT

EVALUATING AND IMPROVING PERFORMANCE OF WEB APPLICATIONS

ABSTRACT

When web applications started to be developed, performance evaluation and improvement in web applications was not considered very important in the first stage. However, performance has become one of the most important criteria with the increasing use of internet by human beings in the future. At the same time, the use of websites on mobile devices has caused performance to be one of the most important issues.

The performance of web applications has increased users' loyalty to and use of the website. That's why companies always have to pay attention to this issue.

In this case, performance measurements should be made and corrections should be made for web applications both during the development phase and after going live.

In the thesis study, together with the evaluation of the web performance of an existing web application, what should be done while the developments continue in a project to be started are discussed. In addition, the effect of performance on people will also be measured.

In this thesis, how to evaluate the web performance of an existing web application and what to do when developing a new project are discussed. In addition, the effect of performance on people was also measured.

Keywords: Web application performance, performance metrics, performance impact

WEB UYGULAMALARINDA PERFORMANS DEĞERLENDİRMESİ VE ARTTIRILMASI

ÖZ

Web uygulamalarında performans değerlendirmesi ve arttırılması Web uygulamaları geliştirilmeye başladığında ilk aşamada performans konusu çok önemsenmiyordu. Ancak ilerleyen zamanlarda insanoğlunun internet kullanımın artması ile beraber performans en önemli kriterlerden biri haline gelmiştir. Aynı zamanda mobil cihazlarda da web sitelerin kullanımı performans konusunun en önemli konulardan biri olmasına neden olmuştur.

Web uygulamalarının performansı kullanıcıların web sitesine olan bağlılığını ve kullanımını arttırmıştır. Bu yüzden şirketler bu konuyu her zaman önemsemek zorundadır.

Bu durumda web uygulamalarının hem geliştirme aşamasında hemde canlıya alım sonrasında performans ölçümleri yapılmalı ve düzeltmeler yapılmalıdır.

Tez çalışmasında, varolan bir web uygulamasının web performansını değerlendirme ile beraber yeni başlanacak bir projede geliştirmeler devam ederken yapılması gerekenler ele alınmıştır. Bununla beraber performansın insan üzerindeki etkisi de ölçülecektir.

Bu tezde mevcut bir web uygulamasının web performansının nasıl değerlendirileceği ve yeni bir proje geliştirilirken yapılması gerekenler tartışılmaktadır. Bununla beraber performansın insan üzerindeki etkisi de ölçülmüştür.

Anahtar kelimeler: Web uygulama performansı, performans metrikleri, performans etkisi

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 Background Information

When the World Wide Web was discovered in 1989 (Lumsden, 2012), websites were not very popular because there was no internet network and browser yet. However, in 1996 Microsoft released its first competitive browser, which was complete with its features and HTML tags. (McPeak, 2018).

When developers started developing websites on the developer's side, their first goal was to build a website and present it to people. However, the web development processes that are currently ongoing have started to occur. Of course, when the first browser came out, some websites could only be created with text. There were no content displays such as images and gifs.

By the announcement of CSS, the websites were made more appealing to the eye. Then, with ongoing additions such as flash, JavaScript, etc., it took its current form in 24 years (Lumsden, 2012).

Back today, websites have become an indispensable part of our lives. People browse and obtain information on millions of websites using both mobile devices and computers. With the proliferation of devices and websites, optimization studies have begun to make it easier to use and provide people with better experiences.

User experience has gained a lot of importance in developers' lives since 2015. The most important point of user experience is performance. Even if you develop an adequate website in terms of design and experience, a website with poor performance will make the user unhappy. Therefore, it is essential to increase the performance of the website and to continue the developments in this area.

1.2 Problem Definition

When we look at the problem in general, what we are trying to do is to make the customer, the user happy. But the first way to do this should be the performance for developers.

It is a very deep issue when you get into the performance detail. Interface design also affects performance. At the same time, databases, which are the back end of websites, also affect performance. The codes written by the developers are as important as these 2 issues.

Websites can be examined in many different categories. They are divided into two as static and dynamic according to programming types. Static websites are sites made without using any backend without taking any data. Measuring and improving the performance of these sites was easier than dynamic websites. Dynamic websites are websites that read or write data from a data source and bring them together with the user. Processing data and bringing it together with the user also creates a workload, so it is more difficult to evaluate and improve it in terms of performance.

Different types of websites are user and content-sided. For example, the website of a corporate company has a simple design and operates on fewer data. However, e-commerce sites deal with too many products and users. This leads to more performance problems and time wasted on solving these problems. However, if this is not seen as a waste of time, it will increase the user experience.

1.3 Aim of the thesis

With this study, all details of web performance will be explained. In this way, it will be easily seen how to increase web performance and its effect on the human. At the same time, a survey to be conducted will compare 2 web applications with high and low performance.

The first focus of the thesis understands what web performance is and when this topic is being discussed. However, detailed information will be given about the reason for the discussion and its effects.

The next focus will be on ways to improve the performance of an existing website or a new website. In this way, people who read and examine the thesis will be able to access practical information and implement it in the applications they have developed. One of the main aims of the thesis is to explain its effect on people comfortably.

1.4 Thesis Outline

This thesis consists of six chapters organized as follows:

In Chapter 2, we provide detailed background information and a literature review of the related studies and techniques.

In Chapter 3, we examine when the first steps are taken to review website performance. At the same time, we will explain the necessary methods for evaluating web performances in this chapter.

In Chapter 4, we show in detail how to fix the problems on our website, which we have explained and evaluated in chapter 2, and how we can improve web performance. However, we will provide you methods on how to progress in performance-oriented development while developing a new project.

In Chapter 5, we explain in detail the results of the survey we conducted on the web performance of people using different graphics. Its effects on people will be one of the most striking points of the study.

In Chapter 6, we explain in detail the results we achieved with our study and the comparison of these results. At the same time, we talk about what can be done in future studies.

CHAPTER TWO
LITERATURE REVIEW

In this chapter, previous studies on evaluating and increasing web performance are examined. At the same time, the differences between previous studies and this thesis are discussed.

2.1 Web Performance History

In the early 1990s, the first web applications started to be developed. However, at that time, only web applications consisted of static HTML pages. The first websites were completely text-oriented, and at that time the perception of performance was not yet established as the only goal was to create an application and all applications were slow. In 1995, web applications began to become more interactive with the use of the JavaScript language on websites. With the dynamic elements, users could interact on the site and it started to be a bidirectional data flow. A year later, with the announcement of Macromedia Flash, it became widespread not only for information purposes but also for entertainment and game purposes. Within 6 years, completely text-based web applications started to come to the level of game and dynamic websites. As can be understood from here, it was obvious that many people will use web applications in the future (Lumsden, 2012).

In 2005, with Ajax technology, end-to-end web applications started to work as a single system as front and backend (Garrett, 2012).

When we came to 2008 (10), the topic of performance started to be talked about and work on it. Different companies have begun to publish studies that reveal the impact of performance on user loss or gain, so it is now seen that performance has a significant impact (Souders, 2008).

In 2009, it was started to investigate mainly in which business areas web application performance is more important and effective, and improvements have been made in this regard. Technically speaking, the multi-business effect was greater (Souders, 2008).

In 2010 and 2011, the performance of web applications started to be examined technically. The opening times of the pages, the metrics that affect them were examined in detail and the importance of the performance of the websites was emphasized (Souders, 2010).

2012, web applications were now being examined technically, but at the same time, a great increase in user data required improvements to the back end as well as the front end (Lumsden, 2012).

Today, due to the rapid increase in data and the rapid operation of all applications, the performance issue puts a huge workload on both the backend and the front end.

Figure 2.1 History of world wide web and performance

2.2 Related Works

Garg et al. (2015) explained some metrics for web performance. However, metrics for evaluation are not included here. With this study, the author mentioned only the performance improvements that had a big impact. Researchers mentioned the importance of image files and their effect on performance enhancement in this study. In the study, some processes to increase the performance on the server are discussed. The Load Balancing process, which is the most important of these, is explained in detail with graphics. Along with load balancing, cache mechanisms are also discussed in detail. Proxy, server, web, and client cache types are mentioned. In this thesis, the importance of cache operations is also emphasized. As a result, the researchers explained some performance improvement metrics with their study and stated that network traffic and latency will decrease with their implementation.

Manhas (2013) has discussed the factors affecting web performance with the study it has done. In the study, the researcher mentioned the importance that large companies attach to web performance. The researcher also mentioned the technical infrastructure included in the web pages at the introductory level. The researcher mentioned the web pages on the World Wide Web and the file types in them. The data in the table in the study reveals that the performance of web applications should be considered. In the last parts of the study, the researcher mentioned the performance enhancement procedures that can be done at the introductory level. However, the researcher did not mention which values measurements should be made. As a result, this study has proven the importance of web performance with numbers and tables. But it makes very few recommendations in terms of performance improvement suggestions.

Gracas (2020) discussed performance metrics and the importance of performance on Single Page Applications. Single-page applications have started to take a big place in the web world recently. In the second part of the study, the metrics to be used for performance measurement for an existing single-page application are presented.

The metrics described in this study are evaluated based on page loading time, resource loading times, user-centric metrics, loading times of JavaScript files, and bundle size. The tools and applications that will be used while making measurements are also mentioned. While explaining these metrics, Single Page Web Applications are taken as the basis. Therefore, it is mainly explained over new interface frameworks. In the solution part, the new framework created by the researcher is explained. The results of the solution presented in the study are also presented. The researcher also presented the evidence according to the performance metrics mentioned in the first sections with the framework created. As a result, with this study, some metrics have been discussed and solutions are presented for single-page applications.

With the study titled "Measuring Web Latency and Rendering Performance: Method, Tools & Longitudinal Dataset" (Asrese et. al. 2019), a different structure from previous studies was conducted. In this study, the Web latency parameter has been discussed and a brand-new solution has been presented to improve this parameter. The Time To First Byte (TTFB) value is the most important metric in the server response process. In this study, DNS lookup time and TTFB values were recorded with the software named "Webget". At the same time, with the "WebPerf" software, TCP and HTTP statistics were measured. In addition to this measurement, the researchers have expanded this software and reached the content on the Web pages. Using these contents and software, they passed the content they reached through parsing and rendering servers. They compared the results of this process with the large browsers currently used by users. While validating, they used the websites of large companies. In this way, they tested the result they created on more accurate data.

With the study titled "Improving Web Performance by a Differencing and Merging System" (Jevremovic et. al. 2012), the researchers presented a solution called the difference and merging system (DMS) on the HTTP protocol. In this study, one of the most important performance metrics, response time, was examined. It is aimed to find the differences between the downloaded documents and processed web documents while opening the web pages. These differences were found with DMS

and only the document belonging to the differences was downloaded. In this way, the loss of performance caused by re-downloading the entire document was prevented.

While working, the researchers also discussed the differences between 10 web documents. Considering the differences over 10 documents, they stated that there is an average of %50 data transfer gain. The researchers used live chat applications in the sampling section. They described the effect of gaining performance when previous messages are not downloaded again after a new incoming message. In this study, while addressing the performance problem, server response time was examined as a metric and a solution called DMS was presented.

2.2 Comparing Our Thesis with Related Works

In the previous studies, some of the metrics in the theses examined in our thesis study were also discussed. However, there is no study examining the web performance with all the details at the edge. Existing studies have generally progressed on one topic.

When we examine the differences between the existing studies and our work, the biggest difference is that it evaluates the performance of the web technologies currently used. At the same time, all web performance metrics are described by dividing them into backend and frontend. All criteria for analyzing the performance of an existing web application are explained and performance recommendations to be considered when starting a new web application are presented. The effects of web performance improvement processes on applications have also been demonstrated with the project. Finally, the web performance effect on different age groups was also evaluated with a questionnaire.

Along with the table below, the differences can be seen in terms of performance metrics with other works.

Table 2.1 Differences between our study and other studies

Thesis / Metrics	Hits per second, Transaction Per second	Page View & Page Load	Response Time & Request Time	Server Response Time	Evaluate Backend & Frontend Separated
Analysis of various techniques for improving Web performance (Garg et al., 2015)	No	No	No	No	No
A Study of Factors Affecting Websites Page Loading Speed for Efficient Web Performance (Manhas et al., 2013)	No	No	No	Yes	No
Measuring Performance in Network-Intensive Web Applications (Gracas et al., 2015)	No	Yes	Yes	Yes	No
Measuring Web Latency and Rendering Performance: Method, Tools & Longitudinal Dataset (Asrese et. al. 2019)	No	No	Yes	Yes	No
Improving Web Performance by a Differencing and Merging System (Jevremovic et. al. 2012)	No	No	Yes	Yes	No
Our thesis	Yes	Yes	Yes	Yes	Yes

CHAPTER THREE
EVALUATING WEB PERFORMANCE

3.1 Web Performance Metrics

Certain metrics must be used in performance measurements. The number of visitors alone is not a sufficient metric to measure website performance. In this section, measurement metrics will be examined in detail.

3.1.1 Transactions per second

The number of transactions per second is one of these metrics. It is the metric that represents the smallest unit of operations performed per second on the web application. All transactions on the web application can be added to this category. It includes all actions taken by the user, such as clicking, filling out forms, on the site. At the same time, financial applications reflect their transactions to the outside over this value. This value can be examined with graphs like the graphs below.

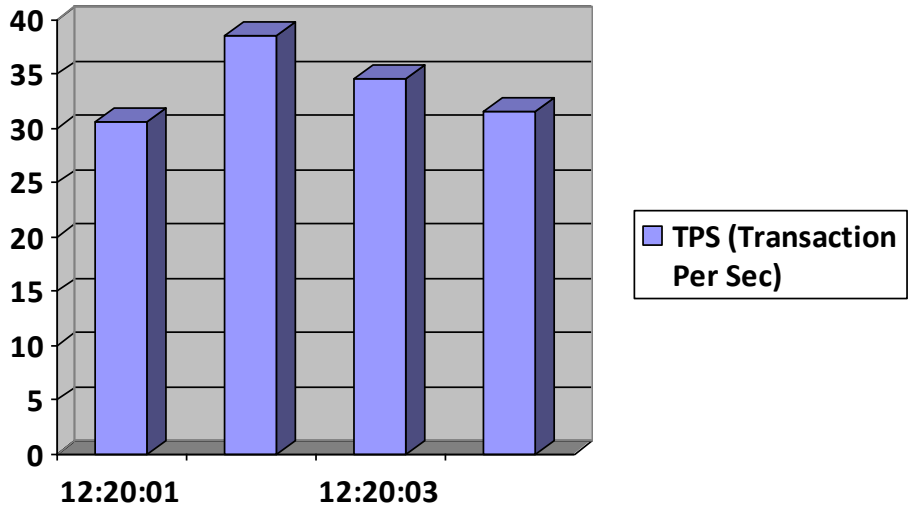

Figure 3.1 Transaction per second example data

3.1.2 Hits per second

Hits per second value is a different value than the transaction per second. It mainly covers all requests for the backend. However, transaction per second covers all transactions without separating them as frontend or back end. Since there are database operations, the periods here should also be evaluated in terms of performance.

It is mostly displayed as line charts and below is a sample hits per second chart.

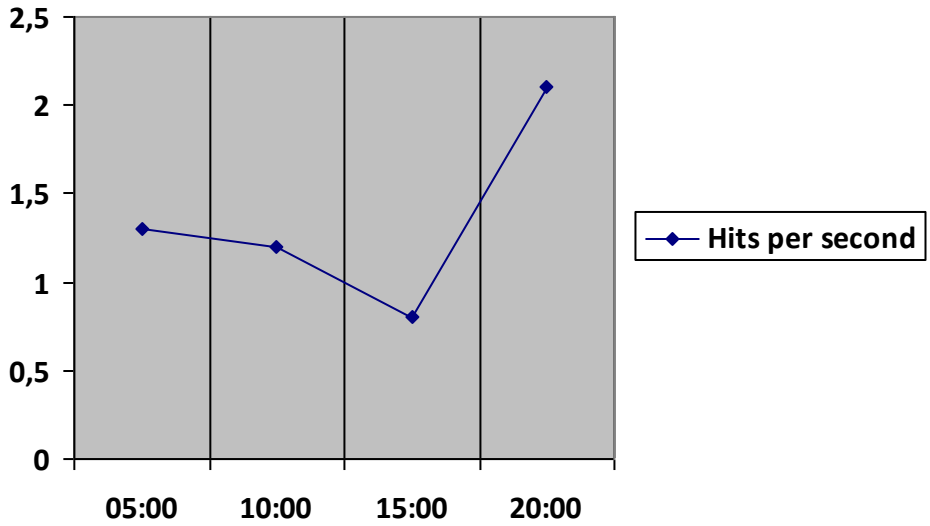

Figure 3.2 Hits per second example data

3.1.3 Throughput

Throughput data is a metric related to hits per second and shows the amount of data processed per second. The relationship with hits per second here is since each request and its result contains data. Sample graphics for this value can also be seen below.

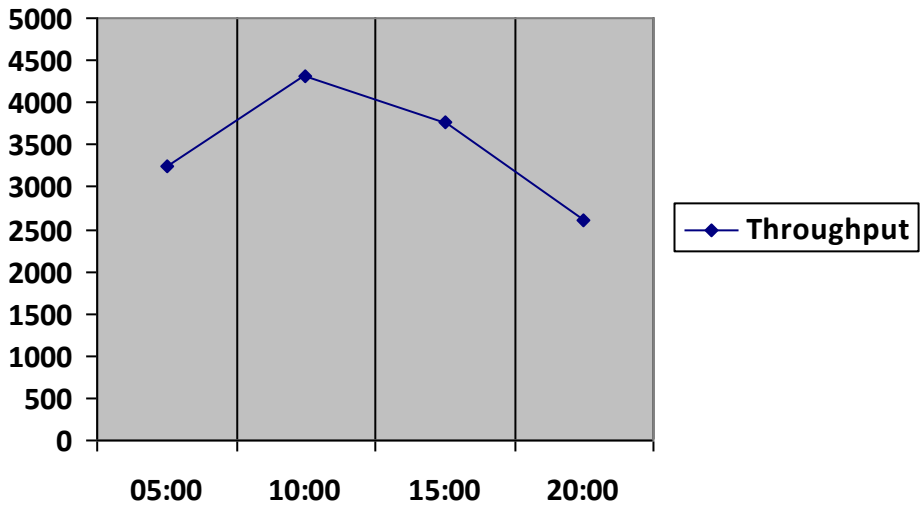

Figure 3.3 Throughput example data

3.1.4 Page View

The most basic and most important data in the metrics here are page visits. Because the entire infrastructure, servers, or pages on the front are shaped according to the number of visits to the site. The number of visits is important data in terms of providing a good user experience to ensure that the visiting users come back. A good user experience can be provided by a web application that is performance and meets the needs of the user.

Today, companies that provide this data provide very detailed information. All data such as the pages that the incoming visitors enter, the sections they interact with on the site, the duration of their stay on the site, and the sections they interact with the most can be accessed. To access such data, certain script codes provided by the companies providing this data should be used. Big companies like Google provide you with very detailed data with these scripts.

At the same time, live data flow is provided. The data on the number of users currently on the site allows it to be followed live.

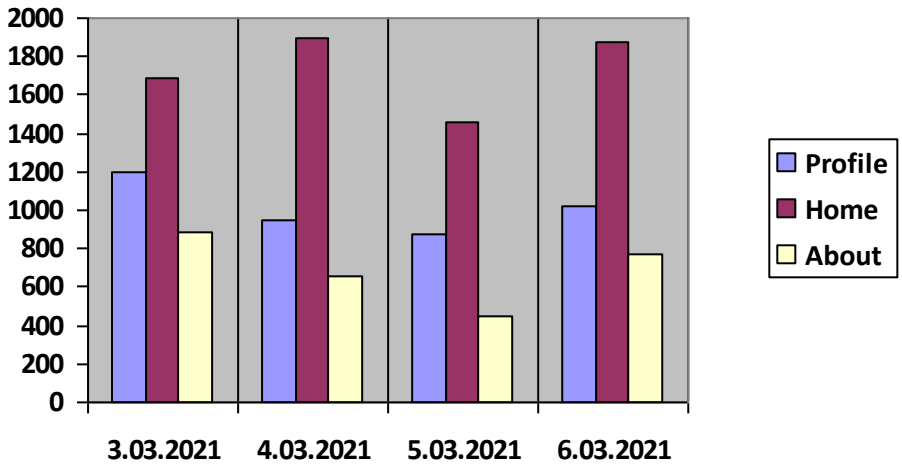

Figure 3.4 Page view counts example data

3.1.5 Request Per Second & Response Time

Request per second value is the list of transactions that are expected only as a data response on the site. The result of the data made based on seconds shows the desired transactions. Response time, on the other hand, shows the response time for these requests, that is, the data rotation time. Response time is the first and most important data to be evaluated in performance improvements to be made on the backend side. The graphics of the response time value are as follows.

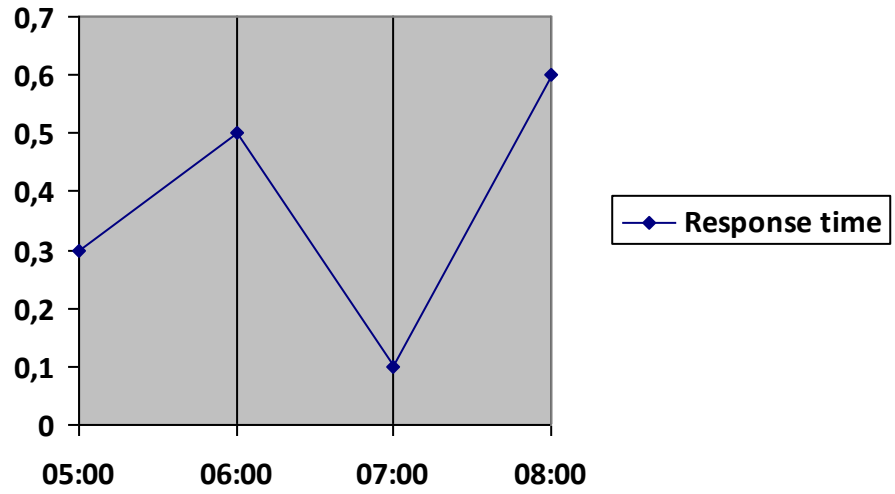

Figure 3.5 Page response time example data

3.1.6 Conclusion of Metrics

The metrics described above are the most basic metrics that can be addressed for a web application. With this data, the average number of users before the performance enhancement process, the load on the site can be obtained.

At the same time, some performance improvements can be made using this data, very basically only server settings or cloud system settings if cloud systems are used. However, if the values here are bad, the performance increase should also be done on the code or architecture side.

3.2 Evaluate Web Applications

3.2.1 Evaluate Backend Side

Since the front-end part of the applications is the part that the user sees and interacts with, it is always considered more important in terms of design, experience, and performance. However, the backend processing times and displaying this to the user on the front-end part have the most important effect.

Accordingly, back-end operations should be evaluated under certain headings. Some of these are as follows.

- Server response time
- HTTP Calls
- DNS Lookups
- Connection Times
- Resource Utilization

3.2.1.1 Server Response Time

Server response time specifies the time the web application is loaded while being displayed to the user. What is meant before loading is the HTML document displayed on the browser. The prolongation of this period has the greatest effect on the user. Because if the web application is loading slowly, the end-user will think that the whole site will run at this slow speed and will stop using the site.

The server response time is measured with the Time to first byte (TTFB) value. Time to first-byte is the time of the first byte of data on HTTP requests made by the user on the server of the web application. The more this value is optimized, the lower the server response time value will be, so it is very important.

Time to first-byte value is measured in 3 main sections.

- Time required for the HTTP request made.
- The time required to process the HTTP request.
- The time required for the response of the HTTP request to be returned after processing.

The Time to first-byte value can be viewed on the browser for the uploaded HTML file as seen in the figure.

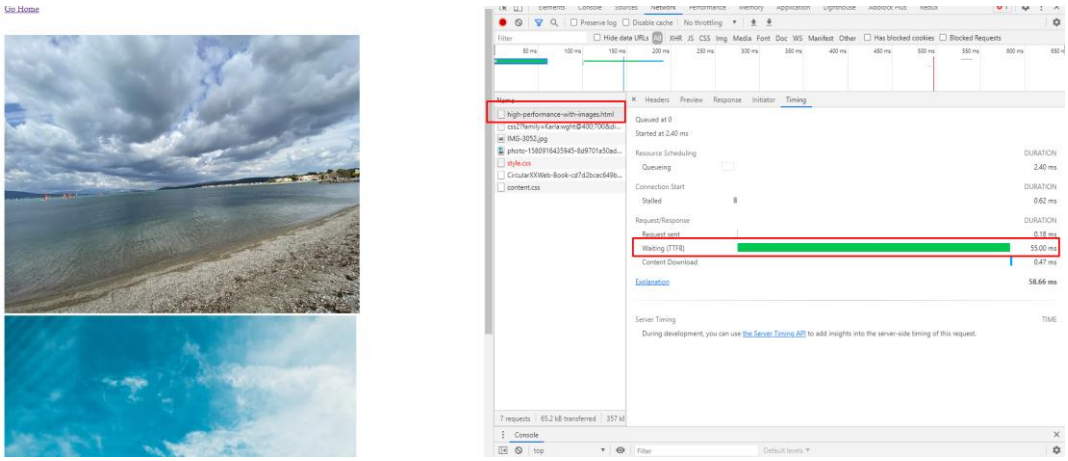

Figure 3.6 HTML file server response time example on browser

3.2.1.2 HTTP Calls

The more HTTP requests there are in the application, the more processing frequency there is. While this does not affect the performance of the application in some cases, it affects the crash or the slowdown of the application in different situations. This has a direct impact on performance.

There are 2 main topics. The first is that having many files causes many HTTP requests. The other issue is that there are large files. Likewise, large files cause long HTTP requests. All files contained in the web application should be inspected to avoid many files. Here, first, the loading time of the application and the HTTP requests it has made should be examined, which files and sizes should be determined. All HTTP requests made while loading a site through browsers are displayed. In addition, Type, Status, and Time information of HTTP requests are also displayed. It is seen in the figure below.

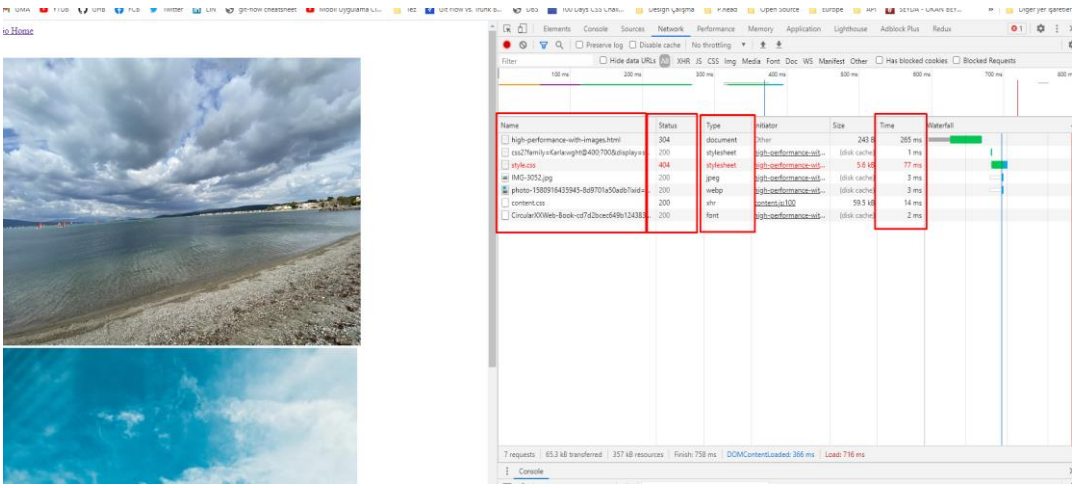

Figure 3.7 HTTP calls and files example on browser (Personal archive, 2021)

3.2.1.3 DNS Lookups

When users enter the web page, they match an IP with the domain name through the existing Domain name system. This pairing time increases according to different situations. Operations such as the distance of the server, the caching status, affect this time. This process cannot be skipped. Because for internet access, domain names need to be assigned an IP and accessed via this IP.

Domain names need to be resolved before they can match IP. Therefore, both domain name resolution and IP address matching also affect web application performance.

3.2.1.4 Connection Times

Wireless and mobile connections are important considerations for web applications. This is because it should have lower bandwidth than wired connections. As the bandwidth decreases, more packet loss will occur, so big problems may arise in terms of performance. Opening too many connections and extending their duration while the user is in the web application becomes a high risk in this case. If interactions initiated with high performance by the user are slowed down due to location change or receiver problems, it causes performance slowdown.

The prolongation of such processing times due to open connections causes the user to be informed that the existing connection exceeds the timeout period. This creates the most undesirable situation for an end-user.

3.2.1.5 Resource Utilization

One of the most important issues to be examined in a web application is resource usage metrics. Resource utilization required large hardware operations and costs when web applications first started to emerge. Accessing the necessary resources was difficult and time-consuming compared to today.

As of today, most web applications run on cloud systems. In addition, the number of companies that manage resource use and operations is quite high. With the increase in the use of cloud technologies, it has now become very easy to change the resources of applications.

The most measured and monitored values in monitor operations are the following values.

- CPU
- Disk
- Memory
- Network

All the values listed above are important in different ways. During the use of the web application, these values should be followed and when necessary, resource increase and optimization should be done. No matter how good the development and architecture parts of the web application are, if there is a lack of resources or server problems, the performance will always be low.

3.2.2 Evaluate Frontend Side

Although it is very important that the back end is fast in web applications, it is equally important on the front end because the user always interacts with the front end. Developers mostly focus on the problems here, since what needs to be done on the front end is simpler and faster to solve than the backend.

The main problems that may occur on the front-end side of web applications are listed below.

- CDN Usage
- JavaScript File Problems
- CSS File Problems
- HTML Problems
- Image Problems

3.2.2.1 CDN Usage

Web applications can have users in multiple locations around the world. In this case, CDN (Content Delivery Network) plays a very active role. The reason for this is that the files used on the CDN are stored on the servers closest to the user, so accessing the files that are close in location is faster. With this speed, the web application performance is maximized while the files tried to be accessed from a single point decrease the performance.

The CDN usage of files should be checked for new or existing projects. The general logic is understood with the diagram below. The closest server is in communication with the original server, so the user can access the files from the closest server.

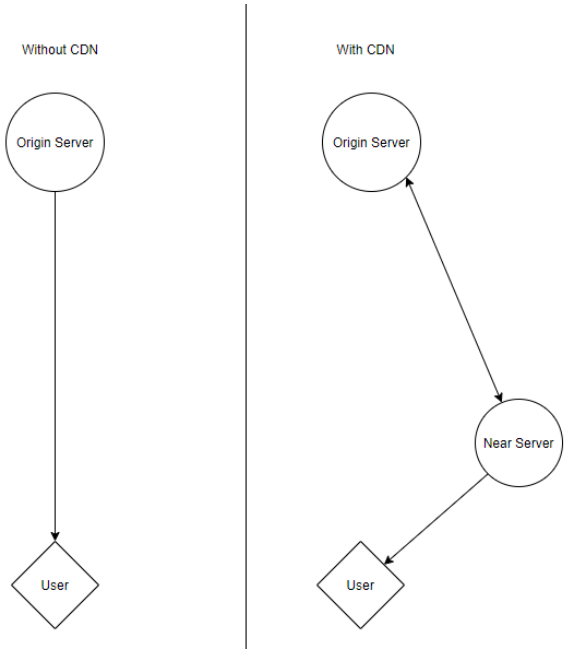

Figure 3.8 CDN server and normal server example scheme

3.2.2.2 JavaScript File Problems

The most used script language in the front-end part of web applications is the JavaScript language. JavaScript controls communication with the backend, front-end transactions, validations, and all other operations. All web applications contain excessive JavaScript code. Optimization and optimization of JavaScript files are very important. The most basic and first thing to do is to check the position of JavaScript files in web application codes. This is because loading JavaScript files before the web application is loaded slows down the application.

Reading the JavaScript files in the website as they are written brings an additional load for the web application. It should be checked whether the minify process is implemented on all JavaScript files. The methods written during JavaScript coding or the required code mustn't be repetitive. Attention should be paid to the generic coding procedures for the codes that do the same operations. With this clean coding, JavaScript files are automatically reduced in size and easier to manage.

In the first of the figures below, the jQuery library had to be loaded with a period of 13ms with its unminified form.

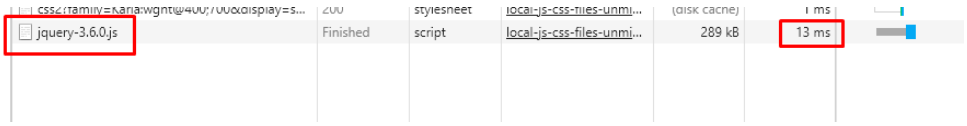

Figure 3.9 JavaScript file load time on browser

However, the same minified library in the second figure was loaded in 8ms. Although the difference here seems not to be very important in terms of the project used on the thesis, it shows how many files there can be for a web application used by millions of people and the importance of these times.

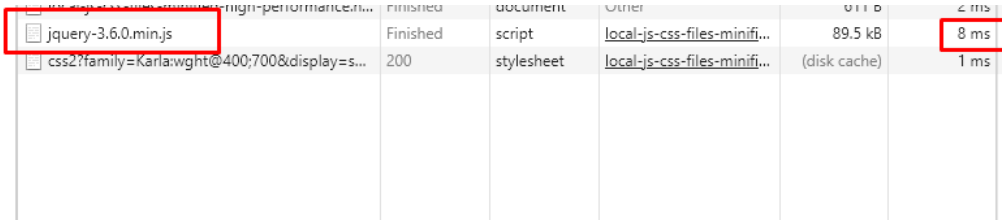

Figure 3.10 JavaScript minified file load time on browser

3.2.2.3 CSS File Problems

CSS files are an indispensable part of all applications found in the web world. Independent of all functional operations seen in web applications, all of the designs, animations, and colors on the screen are made with CSS files. With the change of web applications and discoveries, different approaches have also emerged on the CSS side and the contribution of the improvements to the CSS files on the web performance has been proven.

The minify process applied to JavaScript files can also be applied to CSS files. Because even if the CSS codes written on a site are in different files, they can be in the form of long lines of code. Accordingly, all of the CSS files on the project should be checked.

With the minify process, necessary and unnecessary CSS files should also be detected.

Most of the animations in web applications are provided with CSS codes. Some websites seem to have low performance due to animation processes, even if the entire backend and front end are very performance. Therefore, animation processes in web applications should also be controlled. In addition to CSS files and sizes of web applications, extra coloring and styling processes in CSS codes should be reviewed. The presence of unnecessary CSS codes creates an extra load on the system during installation.

3.2.2.4 HTML File Problems

When the history of web applications is examined, the cornerstone of websites is determined as HTML files. The entire structure is created over HTML files and browsers interpret the HTML files and show them to the user. When the approach of new web applications is examined, HTML files are created with JavaScript code. However, most applications in the web world work with old trends.

There are different versions of the HTML language. The latest version, HTML5, contains huge improvements and different tags compared to the older versions. Accordingly, HTML versions of existing codes should be checked. The minify process, which can be applied to all file types, should also be applied to some HTML files. If performance improvement processes are started in a web application, it is necessary to examine the codes and identify the problems. Any improvement to be made in the codes will increase the performance of the application.

HTML files are predominantly the first downloaded files for normal working websites. However, the server-side loading process has gained popularity in today's web application approaches. Accordingly, the uploading of HTML files through the server, if possible, should be checked and planning should be made in this regard.

Finally, paying attention to SEO practices, which is the most important process for all websites, will contribute to the correct formation of HTML files during development.

3.2.2.4 Image Problems

Image optimization is the most important issue for websites. Because image size, file size, quality are factors that directly affect the performance of the website. Therefore, using the correct format and choosing the lowest possible dimensions is of great importance in terms of speed. Web application users often leave the product or the site when faced with image loading times of more than 3 seconds. Even 3 seconds is too high to talk about a performance web application. Caching is one of the easiest operations for image files in web applications. Therefore, it should be checked whether all images in the application are cached or not.

Web sites are used from many different points with developing technology and devices. Different image files should be used for devices of different sizes. Images used for a mobile device and a laptop should be different. Image file extensions are generally JPG, PNG types.

However, different image extensions should be used in web applications. All pages should be checked again, file-based web performance should be measured. In this way, it should be determined which image files are involved in long file uploads.

When the download times of all image files are found, the table below should be used to compare them with other sites on the web. By making a comparison in this way, the study should be done until average values or lower values are obtained.

Table 3.1 Website image sizes in world wide web (Rodman, 2013)

Metric	**Unit**	**Top 10%**	**Median**	**Bottom 10%**
Image Count	#	5	25	68
Image Size	Bytes(kb)[MB]	30,295(30.2)	332,420(324.6)	1,476,038(1441)

CHAPTER FOUR
IMPROVING WEB PERFORMANCE

4.1 Web Performance Improving Methods Technics

In this chapter, performance improvement processes that have emerged as a result of performance evaluations will be explained. Accordingly, the improvements to be made for all front-end and back-end sections will be explained. These operations will generally be for an existing web application. However, in the last part, the stages of developing web applications will be discussed in a performance way.

4.1.1 Backend Side Improvements

As explained in the previous chapter, back end is the most important part of the applications. No matter how fast and performance the frontend processes are, the slow backend operation will always have a huge impact on the user.

The operations that can be done at the rear end will be explained with the following main headings.

- Database & SQL Operations
- Caching
- Web Application Hosting
- Improving Backend Code

4.1.1.1 Database & SQL Operations

The database is the most important part of our application. Without data, the application has no meaning and the interaction with the user is minimized. Accordingly, application performance is also directly proportional to database performance. Different procedures can be done to increase the performance of the queries that enable us to operate on the data we call database and SQL. The first action that can be done on behalf of database performance is to optimize the queries that run on the application and manage database operations.

Queries are the most basic database operations. Therefore, the following improvements in queries will have a great impact on the application.

- One of the simplest steps you need to do when starting to optimize the queries is to convert all the data queries you are trying to get into only the fields that are needed, contributing to the performance.
    - Inefficient
        - SELECT ***** FROM Users
    - Efficient
        - SELECT **Name, Surname** FROM Users

- At the same time, using the "DISTINCT" keyword while performing data retrieval in your existing queries will negatively affect performance. Instead of checking it over all the data, it is more useful to take all of it quickly and then filter it.
    - Inefficient
        - SELECT **DISTINCT** Name, Surname FROM Users
    - Efficient
        - SELECT Name, Surname FROM Users

- While working on data, data reading operations are performed over multiple tables rather than a single table. Linking tables using "INNER JOIN" instead of using "WHERE" in queries in these operations yields more efficient results.
    - Inefficient
        - SELECT Users.UserId, Users.Name, Store.StoreId FROM Users, Store **WHERE** Store. UserId = Store. UserId
    - Efficient
        - SELECT Users.UserId, Users.Name, Store.StoreId FROM Users **INNER JOIN** Store ON Users.UserId = Store.UserId

- Limiting the query results and handling the paging process in your queries is the most important step. It is also referred to as making querying server-side rather than frontend.

    - Inefficient
        - SELECT * FROM Users
    - Efficient
        - SELECT TOP 10 * FROM Users

- Planning and acting on large query operations on the database with the least impact on the user will always ensure that performance problems are not seen by the user. The times that will have the least impact on the user are usually midnight. For this reason, banks or institutions with many users usually schedule both database and other transactions at midnight.

The indexing process allows you to search within the data in the fastest way possible. However, indexing is a process that provides speed if done correctly. If the indexing process is applied to each field, this can also create a performance problem. Since indexing works as a data structure, it must be applied correctly and to the correct fields. In this way, an increase in performance can be achieved.

The database and its structure are also as important as the queries, the more important is the power of the **hardware and the system**. Here, the improvements below are the processes that will always be a plus in the way of being more performance.

Using a powerful CPU

- The CPU determines the number of operations per second. Therefore, it is one of the most important factors that directly affect performance. Database systems running on the machine with a powerful processor give faster results.

- More Memory
    - Memory means the sections where data is stored. Therefore, a limited memory always causes performance problems. In case of excess, the performance will always be as expected.

The last section to be mentioned in database and SQL operations is the database selection and version used. In the web application, a database infrastructure suitable for the optimum application and its dynamics should be selected. With each new version of the database used, companies that offer this infrastructure make improvements. However, the older the infrastructure your web application uses, the more problematic it will be. Therefore, using up-to-date infrastructures as much as possible will always provide performance.

4.1.1.2 Caching

Cache is a data storage layer that increases data processing speed to incredibly high levels. You can think of the speeds here close to the processing speeds on RAM. There are cache systems for the data that is not wanted to be read over the database repeatedly. In this way, the results are shown directly to the user, bypassing the database process time again. Cache data is usually kept and accessed on the hardware through RAM. By removing the slow storage layer, data can be accessed in a minimum time. However, although it seems logical when the cache is transferred in this way, it does not include processes such as data security, storage, or backup in case of possible problems. Therefore, it may cause massive data loss. This issue is as important as performance.

Caching can be achieved in many ways. Some cache volumes can be seen below.

- **Database caching**: The importance of database speed and efficiency was mentioned in the previous section. Therefore, the caching of some parts of the database eliminates or reduces server-related delays. This provides an increase in performance.
- **CDN Usage:** The topic of using CDNs will be discussed later. However, using CDN allows the resources on the site to be read from the cache after the first installation. In this way, an increase in performance is achieved.
- **DNS Caching:** The DNS - IP mapping process, which we mentioned as a problem in the previous sections, is solved by the DNS cache servers. After the first installation, the IP - DNS matching process is done through the DNS cache. In this way, delay times are minimized.

- **Session Management:** Session is the information that is kept when users log into the web application. In addition, it is ensured that the information requested within the site is kept within the entire application.

  With Session, data is read over the session instead of reading data from the database again. However, the session is timely and keeps the data for the specified period. When the session ends, the data is retrieved from the database again. In most web applications, a session is created on the application with the login process.
- **Web Application Caching:** Web application caching is a caching process used for documents (image, sound, documents, etc.) used in the web application. It can be configured as browser or server-based.
- **API Caching:** The API caching operation is done by adding a different layer between the database and the API. One of the most preferred databases for this is Redis. Redis is a NoSQL database that works with ram logic and stores data at certain times. However, unlike normal databases, it works very fast, so Redis is overwritten in the first data retrieval process, and in this way, it is read through Redis in the next operations.

4.1.1.3 Web Application Hosting

Hosting service is a service for all websites. Regardless of how fast the front end or backend of your website is, if the company or service you receive the hosting service from is slow, your web application will run slowly. The structures we call hosting are the entire system that includes RAM, Hard Disk, and other hardware. In this case, if a web performance evaluation will be made through web hosting, the following structures should be checked.

- **Ram:** Ram is the center of all work in your web application. Because it is the storage area that maintains all scripts (JavaScript Files, etc.) of your application. The part that manages the cache operations we mentioned in the previous chapters and is used as a cache is the Ram. Gigabyte value should be measured according to the loads on the server and the processing density.

- **Hard Drive:** Hard drives are the partitions where all your data is kept. With the developing technology, Solid State Drive (SSD) type disks have been produced and they offer a lot of speed when compared to hard disks.
- **Bandwidth:** Bandwidth is a value that indicates how much data can be transferred between your web application and hosting. The lower this value is the lower the web application performance.

4.1.1.4 Improving Backend Code

After this much parameter improvement, performance-oriented development is also required on the code side of the web application. Today, there are many programming languages and different development styles. However, it would be wrong to direct it in one way or another. However, no matter how much you write the code you wrote as "Clean Code", this will be good for your application. At the same time, when you write code, care should be taken to prioritize quality and speed and to do research on how to write better.

Below are a few simple but effective methods that can be done in the code. It may contain different arrangements in different programming languages.

- **Pay attention to variable naming in the code.**
    - Most programming types have variable definitions or naming. And in many ways correct naming is important. Maybe it has no direct effect on the performance of your website. However, to be able to read the code and call it clean code, correct naming should be done.
- **Try to follow solid principles.**
    - Although solid principles seem to be created for you to write code more clearly, following the rules here will also affect the speed of your code. Dividing the methods into parts that perform a single task and minimizing dependencies will also provide an increase in terms of performance.

- **Don't put off refactor jobs.**
    - Refactoring must be done before sending the code to the headquarters. Because the delayed code editing or TODO operations will not be done in the future due to the time problem. Therefore, all the corrections and improvements you will make on the code should be corrected before the job is completed, whether they influence performance.
- **Avoid comment lines.**
    - If a code needs a comment line it isn't good enough. There should not be any comment lines that we write between the lines of code. The code should be able to explain its work through itself. If it is necessary to explain something about a subject, it should be included in the documentation of the project.
- **Learn the tools used.**
    - The tools it uses, the best way to learn the programming languages, and the most efficient of the methods used while developing, should be preferred. The best way to do this is to control the vehicles used. However, the more projects you develop or work, the more you can improve.

4.1.2 Frontend Side Improvements

Front end is the most important part of the application. Users interact with your web application through the front end. Web applications with poor performance are not always liked by the user.

Many small but effective operations can be done on the front end. The performance effect of the operations that will be described in the following titles together with the application I made will be explained in the Results Chapter.

- Cleaning Code
- JavaScript Placement
- Minify Operation

- Use Content Delivery Network
- Compressing Files
- Optimize Images

4.1.2.1 Cleaning Code

The codes of the web application are a big factor affecting the performance. Therefore, writing the code more carefully and effectively at the beginning of the project will reduce the workload in the future. However, if the project is a project developed without paying attention to performance criteria, then the actions to be made on the code later will improve the performance.

An increase in performance can be achieved with recommendations for HTML, CSS, and JavaScript in detail below.

Give version with DOCTYPE

The HTML programming language includes different versions. The latest version with new features is HTML5. When using HTML, a definition as "DOCTYPE" is made at the top of the file. With the version defined here, the features of that version of the HTML language are also included in the code. Therefore, defining the version to be used will provide an increase in terms of performance since it will only include the features of that version.

- **Doctype for HTML 4.01 version:**
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
- **Doctype for HTML5 or latest version:**
  - <!DOCTYPE html>

Do not use JavaScript and CSS code in HTML

The entire website can be encoded via just one HTML file. The entire website can be encoded via just one HTML file. However, writing all JavaScript and CSS code in separate files increases maintenance and manageability. There is a performance factor that we have specified as loading JavaScript and CSS files in the correct order. Loading files in the correct order affects performance. Therefore, codes should be written as separate files. Each time the page is reopened, a shorter HTML file is loaded.

Cleaning All the Codes

The first and most important thing to do is to go through all HTML CSS and JavaScript file types and clean up useless and unused code. Clearing code means reducing the size of the files. With the decrease in file sizes, the uploading on the web server is also accelerated. This contributes to the performance of the web application. At the same time, the development processes will improve in the later stages of the project, as the code quality also increases.

4.1.2.2 JavaScript Placement

When the first coding is learned, it is taught to add JavaScript and CSS codes into the "head" tag in the HTML file. The reason for this is to load all of the JavaScript files before the entire screen and components appear on the screen, thus avoiding any problems that may occur. However, having CSS files in the "head" tag is sufficient for the page to load correctly. If the JavaScript files are loaded in the head, the HTML and CSS files wait for the JavaScript files to be loaded. The user sees a blank screen while doing this and thinks that they are using a low-performance application.

In this case, both JavaScript and CSS files that are not required to be loaded at the first opening of the page should be located at the bottom of the HTML files.

The two methods that can be done here are listed below.

- Mandatory files must be uploaded by partitioning them in the "head" tag.
- Files that do not have to be uploaded should be loaded after the "body" tag.

Asynchronous JavaScript files can also be uploaded on the page. For this, it can be done with the "async" parameter as in the example below.

- Synchronous and slow way:
    - <script async src="test.js"></script>
- Asynchronous and fast way:
    - <script async src="test.js"></script>

In addition to all these, the "defer" parameter can also be used. In this way, a deferred file upload is performed using a separate sequence of operations.

- Defer parameter usage:
    - <script defer src="test.js"></script>

Step by step defers, async, and normal file upload sequences can be seen below.

1. HTML Load Started
2. HTML Parsing
3. Parsing paused.
    a. CSS Loading Started
    b. CSS Object Build
4. HTML Parsing Continues
    a. “defer” JavaScript loads have started.
    b. “async” JavaScript loads have started.
5. Parsing Paused
    a. JavaScript executing
    b. Synchronous JavaScript loads started.
    c. JavaScript executes.
6. Parsing Finished
7. JavaScript executes.

4.1.2.3 Minify Operation

The minification process, as the name suggests, is the process of reducing the number and size of lines of code in a file. However, this process is done without breaking the functionality of the code. In other words, unused parts such as

punctuation marks and spaces are discarded in the minification process and a smaller version of the file is created. As we mentioned in the previous sections, file shrinkage contributes to the performance of our web application in many ways. Smaller files are files that load faster on the web server side and increase the speed of the website. The difference can be understood from the example below. The code block below consists of 7 lines before minification.

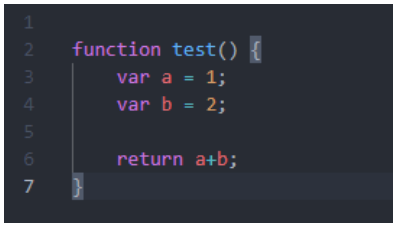

Figure 4.1 JavaScipt simple function example

For the same line of code, after the minification process, the code turns into a single line.

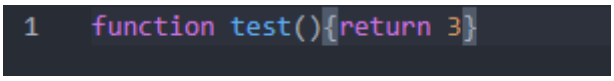

Figure 4.2 JavaScipt minified simple function example

A slight change may also be noticed in the above process. The 2 variables that were defined were removed with the minification process, and the operation that could be done by 7 lines of code was reduced to one line. With the removal of the variables, a workload was also deleted from the code. Thus, performance can be increased by the minification process for all JavaScript and CSS files in the application.

4.1.2.4 Use Content Delivery Network

Content delivery network can be thought of as a web server spread all over the world. When we upload files on a web hosting, a request to our application goes through this server and reads the codes or other necessary files, and the result is shown to the user. However, if the web servers belonging to the hosting service used are far from the location where the users use your web application, the response

times to the requests made also increase. As we mentioned in the web performance evaluation sections, the higher the server response times, the lower the web application performance will be.

With CDN servers, static contents (JavaScript CSS files, pictures, music, etc.) used on the web application are received from the server closest to the user. Here, thanks to the CDN servers located at different points all over the world, wherever the main server is, the CDN server that I use communicates with the main server and sends the necessary file for the application to the user. A standard hosting service without CDN servers:

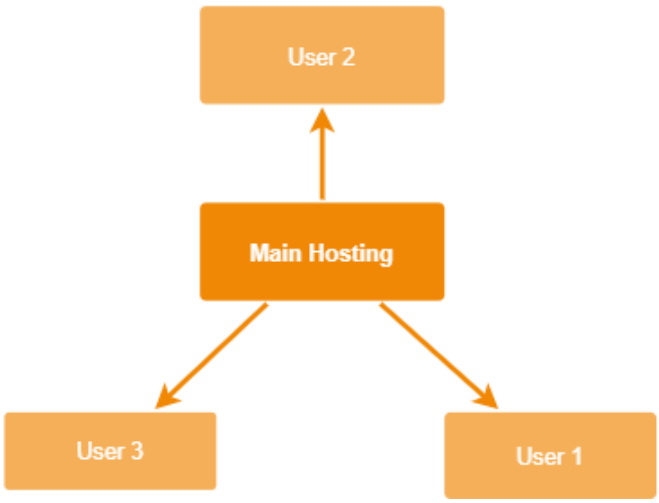

Figure 4.3 Without CDN server scheme example

The web application that contains CDN servers and accesses data via a nearby server:

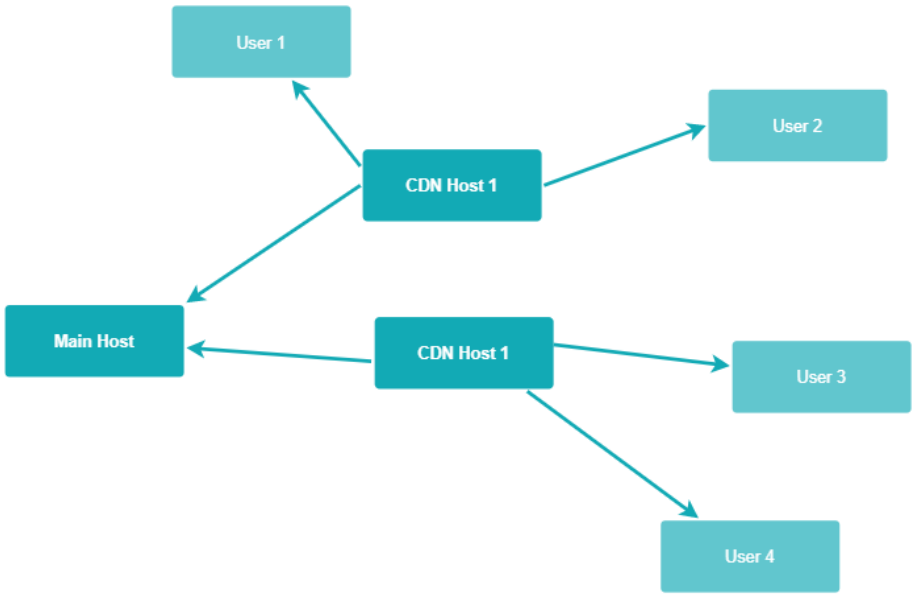

Figure 4.4 With CDN server scheme example

The information the CDN uses when detecting the nearby server is the DNS information. CDN not only increases the speed but also increases the security of your server and web application. What is meant by security here is that applications taken to CDN servers are more secure systems against DDOS attacks.

One of the factors that increase the performance of CDN is to distribute and manage the high traffic generated on the web application. CDN is one of the structures that should be considered and used for your web applications from the very beginning.

4.1.2.5 Compressing Files

From the first topics, small files mean fast websites, as we have explained in the thesis document. The main reason for this is that when you want to open a web application, all files are downloaded at that time. The most basic file compression algorithm used in web applications is "**gzip**". The Gzip compression algorithm can compress almost any file type, from plain text to audio files.

Gzip information can be seen in the content-encoding section of the page in the network section of the browsers.

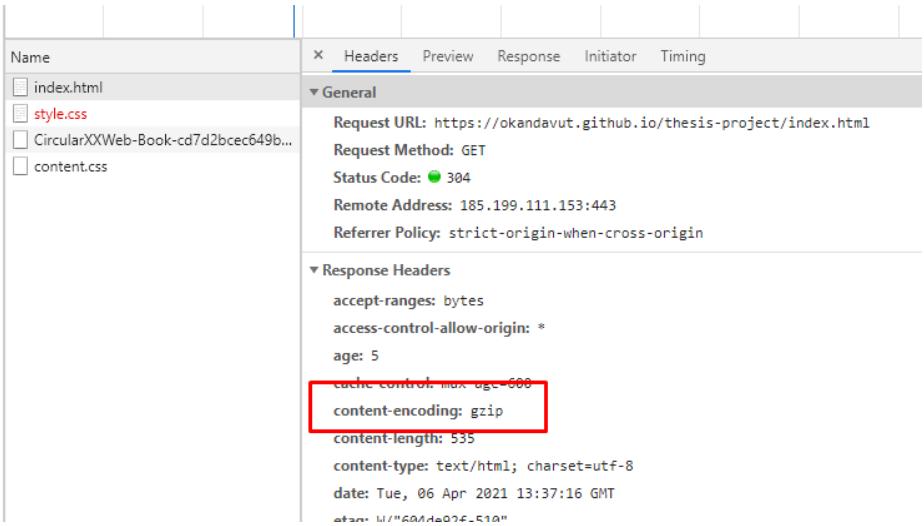

Figure 4.5 Gzip information on browser

To add this compression to a web application that does not use the gzip compression algorithm, the "gzip" value must be entered in the "Accept-Encoding" field in the Header values. After the web server generates the response for the relevant request, it looks for the "Accept-Encoding" value for the compression algorithm and returns the response by compressing it according to the value it reads. For these operations to occur, the server must support gzip compression. After compression, the web server adds "Content-Encoding": "gzip" to the Header fields on the response. The data that comes to the scanner as compressed is converted to its original form by the scanner and displayed to the user.

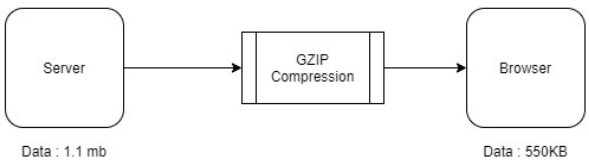

Figure 4.6 Gzip compression size effect

4.1.2.6 Optimize Images

Image files constitute the biggest performance problems of web applications. All the items described above actually provide great advantages in image files. For

example, image files are uploaded in the fastest way with the use of CDN. At the same time, size reduction operations are performed on the image files sent by the server with Gzip compression. In addition to these procedures, web application performance can be increased with faster loading image files by making the following corrections.

- **Using the correct file format**
    - It is necessary to use the correct extensions for the correct purposes while determining the file extensions of the images to be used on the pages of the web application.
    - JPEG, PNG, and GIF file extensions are used in the majority of web applications.
    - The JPEG file format is generally lighter and ideal for smaller files. It gives choice on the quality side and should therefore be considered for performance.
    - Image files with PNG extensions are generally used for situations with transparent backgrounds.
    - GIF format is preferred when image files containing predominant animation are required.

- **Using image files in the correct size (width x height)**
    - Web applications usually have more than one image file. It is not correct to use them as they come directly from the camera. When you take it in its original form and resize it on the code side, the size of the original image will affect the performance.
    - It is necessary to bring the image files to the size to be used on the web page by using different programs. By adding it to the project with new and desired dimensions, the correct file is loaded.
    - As a result, image files should be configured and resized as needed.

4.2 Performance Oriented Website Development Methods

All the methods described in the previous subsections are used for existing web applications that want to increase their performance. However, progressing by

making careful and performance-oriented development for a new application will save an additional workload in the future.

Performance application development, of course, will cost training for existing developers, even if it reduces the workload that will occur in the future. However, the training provided will both provide a plus for the developers within the company and will ensure that the application progresses with the right development processes from the beginning. The information to be transferred in this section will be in the form of steps for the implementation of the processes applied later in the previous sections, starting from the beginning.

Steps for performance-oriented application development are as follows.

- Continuously checking of application performance data
- Partitioning applications as backend and frontend
- Making improvements regularly

4.2.1 Continuously checking of application performance data

One of the most important planning to be done at the beginning of a web application is the plan to regularly check performance data. Because activities to be performed to measure the performance of the web application should be added to the regular works such as deployment operations to be done, work planning that includes weekly or certain periods. The regular actions to be taken may vary according to each company, team, and project. However, if some of the following operations are generally the same, at least continuous control of the web performance is provided.

- Application of load and stress tests on each living creature after intake
- Monitoring system for constantly monitoring the web application.
- Automation tests should be added for the whole application.
- Performance criteria should be checked one by one as stated in the previous chapter.

The operations mentioned above require additional time, training, and cost. However, if these operations are carried out at the very beginning of the road, larger costs will be avoided in the future.

4.2.2 Partitioning applications as backend and frontend

While performance has been mentioned from the beginning of this thesis, both the evaluation and enhancement sections are mentioned separately for the front end and the back end. In the old systems, projects were developed from end to end as a single piece. At the same time, the application was considered in one piece rather than parts such as the back end or front end. With the rapid development of technology and devices, the number of users using web applications has started to increase very rapidly. With this increase, the back end and front-end systems began to be divided into different systems. On the software side, new frameworks for the front end and microservices architectures for the back end have started to emerge. However, software developers, known as full stack, are now separated as frontend developers and backend developers.

As described in the previous sections, different evaluations and procedures are carried out on both sides of the projects in performance measurements. Performance enhancement processes for Frontend and Backend projects are completely different. The division into backend and frontend will be a huge plus to easily apply them to projects and increase application performance. Splitting the application into 2 parts means acting faster and with less dependency for the whole project.

With the diagram below, each of the projects can be evaluated within itself. Thus, dependency can be minimized, and the fastest actions can be taken, and performance enhancement operations can be applied.

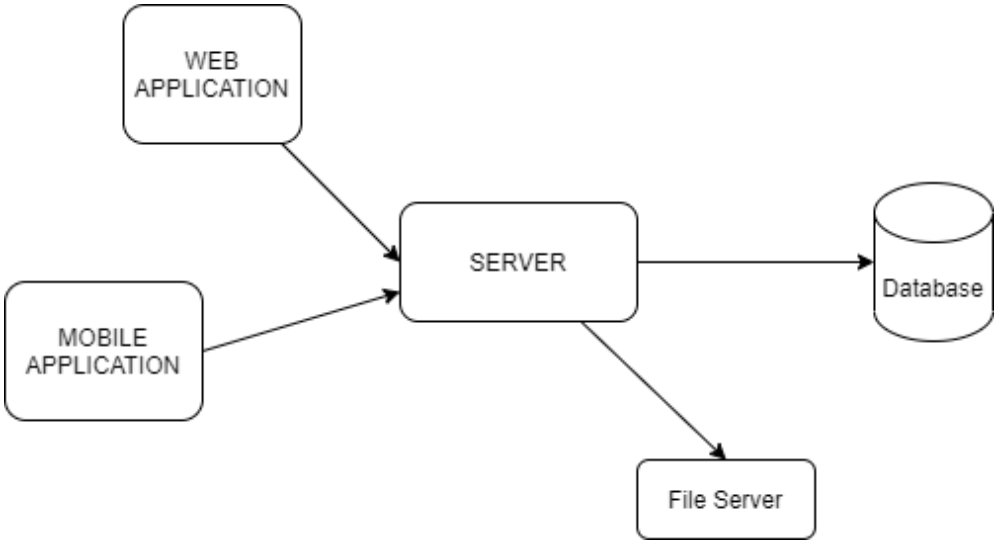

Figure 4.7 Server architecture in web and mobile application

4.2.3 Making improvements continuously

Continuous control provides a process to improve both performance and all other aspects while developing a web application or any application. Regular development means providing solutions for regular checks of all parts of the application. At the same time, with the development of technology, a transition should be made to new technologies in line with the need. In this way, the teams that develop the application regularly improve themselves.

CHAPTER FIVE
RESULTS

5.1 Project and Evaluation

Within the scope of the thesis, we created a web application to see the effect of performance enhancement processes on web applications. The purpose of this application is to show that the actions that are made or not done influence the performance even for a very simple application.

Some comparisons can be seen in the figure below.

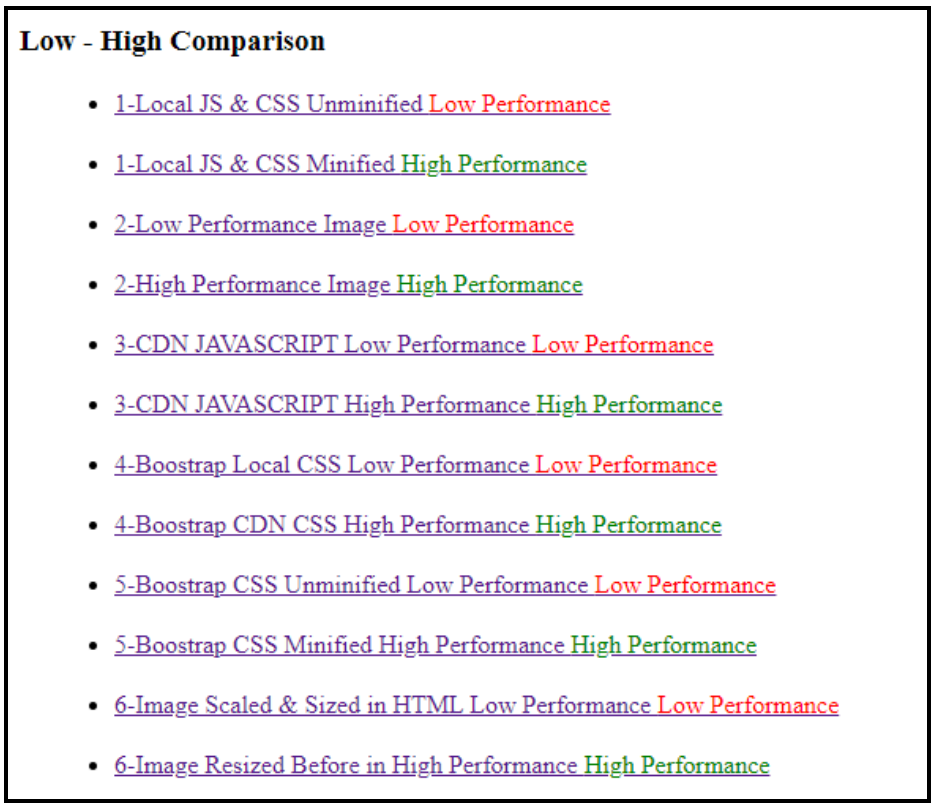

Figure 5.1 Comparison project structure screenshot

Since the examples here are entered through a simple application, frontend improvements are mainly discussed. Performance improvements to be made on Frontend will have a big impact. This is because the user is constantly interacting with the front end. When examining the results of the above comparisons, it should

not be overlooked that the number of users of large applications and such small changes will have a great performance impact.

5.2 Results of Comparisons

In this section, the results obtained from the Network section on the Browser will be discussed under the heading of all the comparison examples made in the project. Instead of examining all the results under a single heading, we evaluated the performance-enhancing processes described above one by one, and the examples of some of them under separate headings. The results under the headings below are the loading times taken from the browser console section of the application.

5.2.1 Local JavaScript Files Minified & Unminified Low Performance

The process of minification of JavaScript files was mentioned in detail in the previous sections. In the first part of the application made for the thesis, the effect of the minify operation on even the simplest JavaScript file has been examined.

As can be seen in the first figure, the unminified version of the jQuery library takes 118ms to load the file.

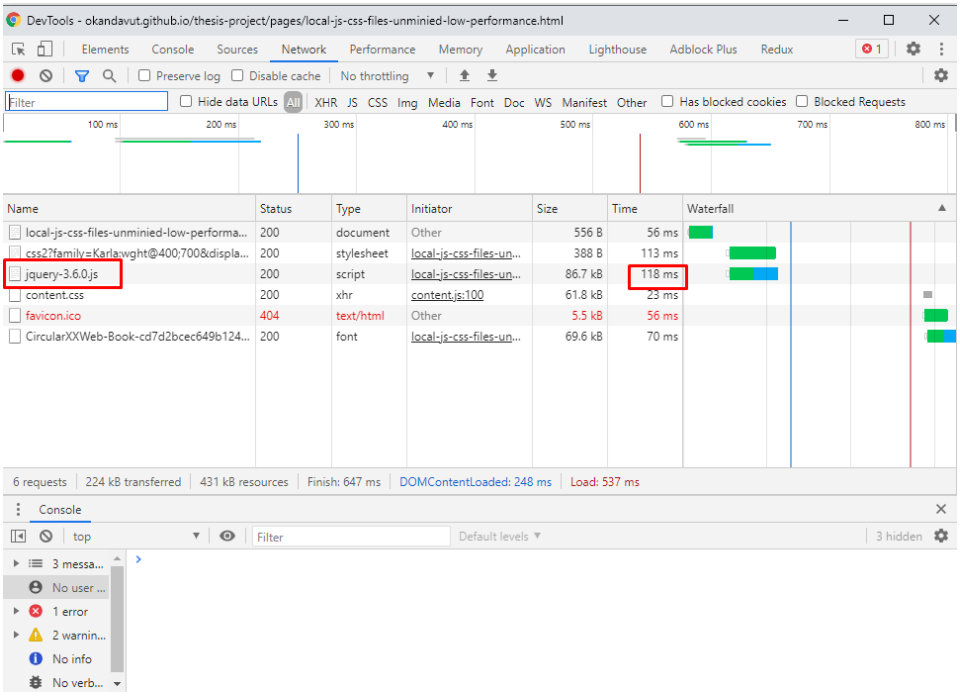

Figure 5.2 JQuery unminified file load time on browser

In the figure below, the loading time for the same version of the minified Jquery library is 62ms.

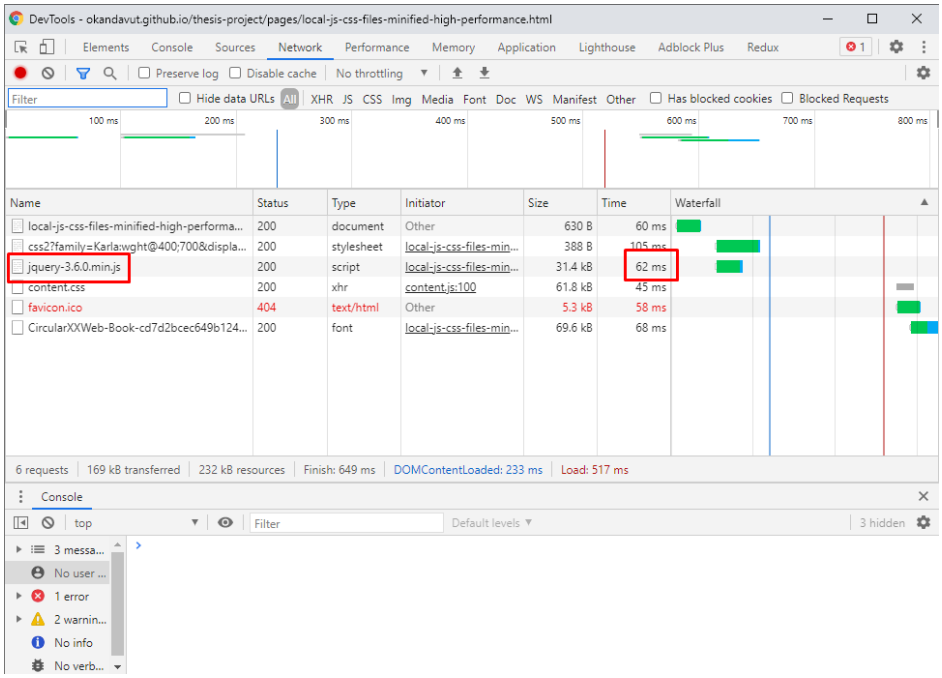

Figure 5.3 JQuery minified file load time on browser

It can be seen how effective the minify process is, with a difference of 50ms in the example here. When we consider this for web applications that contain too many libraries or JavaScript files, the times matter.

5.2.2 Low & High-Performance Image

Images are indispensable elements of web applications. Most web applications have many images. Images in web applications are one of the most important issues in performance processes. If an image file opens slowly and feels like there is a problem, the user can give up using the application.

In the figure below, there are 2 image files uploaded in the web application. These both open slowly and cause the image to be seen in parts instead of waiting by the user. The slow opening times we are talking about here are a huge number of 5.45 seconds for the first image. For the second image, it is 3.51 seconds. These are very high numbers for a user.

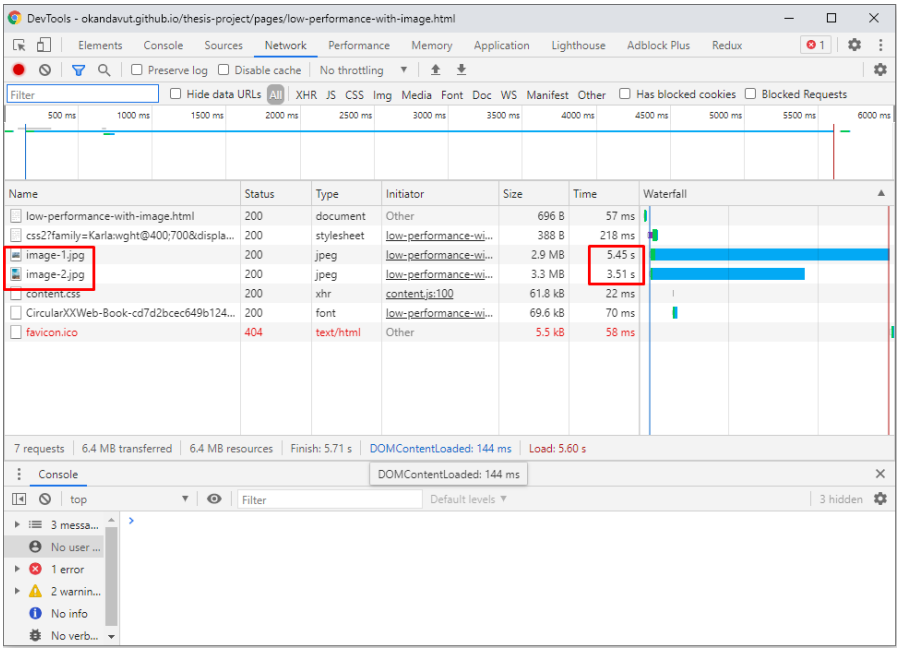

Figure 5.4 Images low-performance load times on the browser

In the picture below, the image files were loaded at much faster times. This is because the files are used over the content delivery network. Here, the first image with a loading time of over 5 seconds was downloaded in 601ms. The download time for the second image decreased from 4 seconds to 389ms. These times are acceptable for the user. In this way, a better experience is provided to the user. This will be a huge problem for sites with too many images.

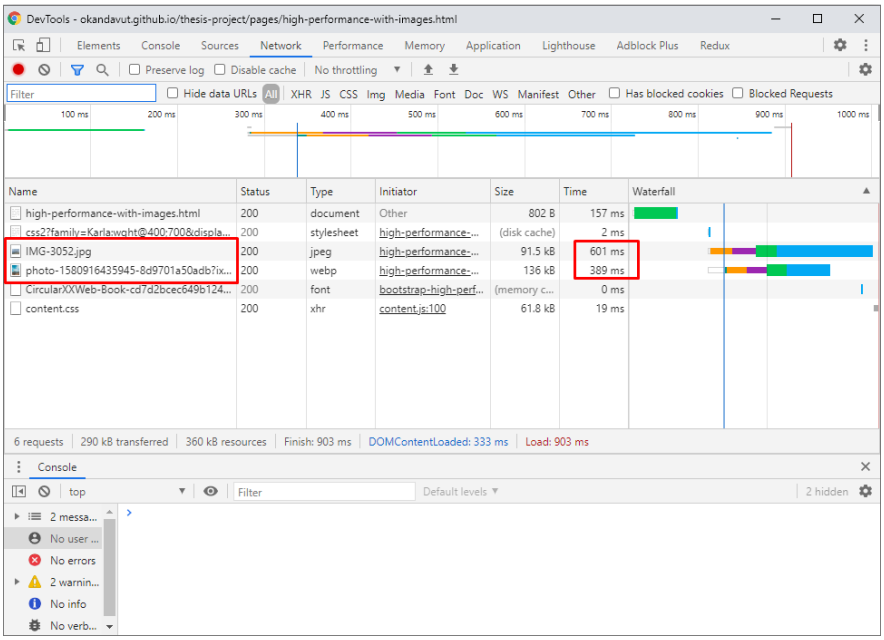

Figure 5.5 Images low-performance load times on the browser

5.2.3 JavaScript CDN or Local Low & High-Performance Low Performance

In some cases, it may not be possible to provide more performance with the content delivery network mentioned in the previous sections. In this part of our application, we have shown the performance drawbacks of using a wrong content delivery network server.

As can be seen in the picture below, the jQuery library file in the local files was loaded faster. The loading time took a period of 61ms.

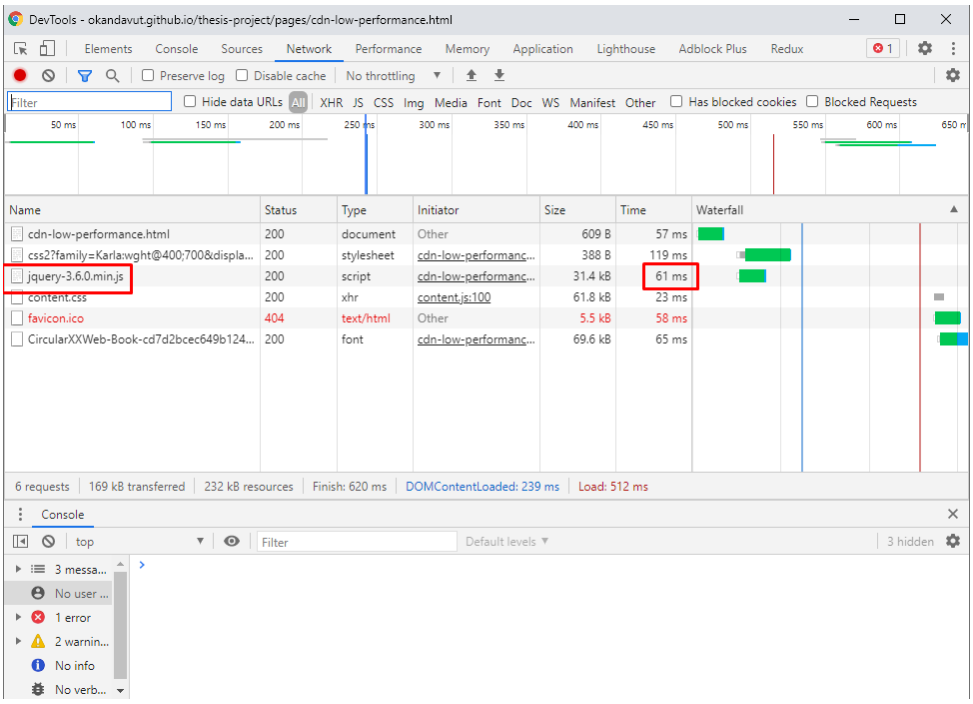

Figure 5.6 JQuery minified load time on browser

Looking at the figure below, the jQuery library used over the content delivery network took 122ms. The main reason for this is the difference between the country where the application is used and the country where the content delivery network server is used. Such situations may arise if the content delivery network servers are far from the country where the application is located or used.

Therefore, while increasing the performance (server selection etc.) the effects of this process should be measured, and the correct actions should be taken after each process.

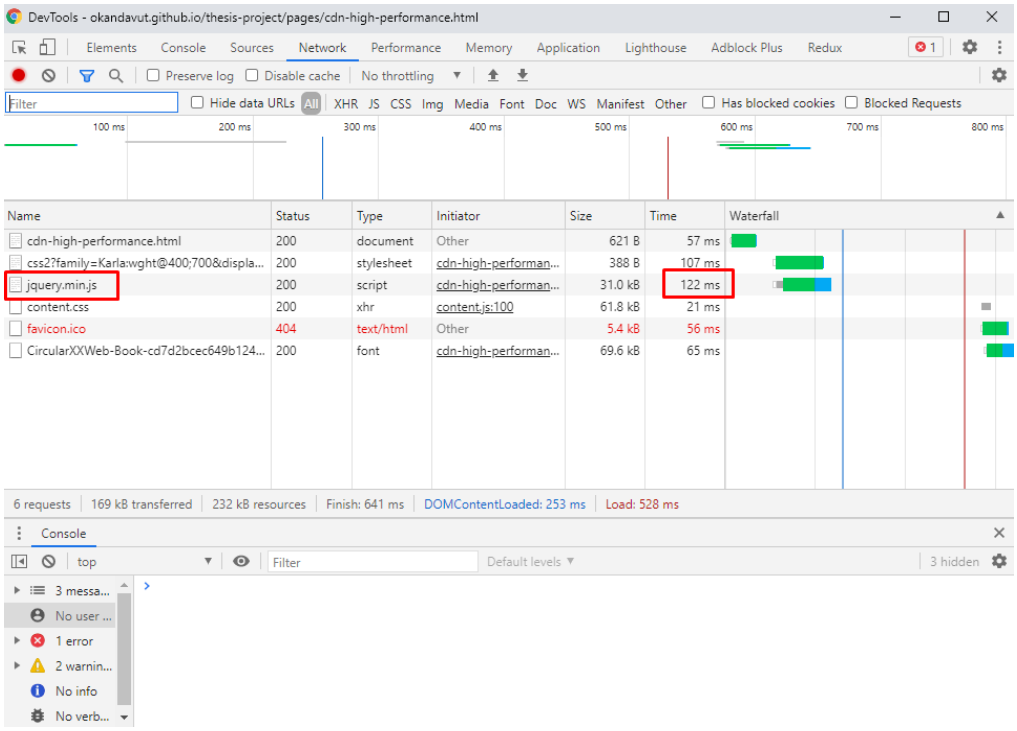

Figure 5.7 JQuery minified low-performance load time on browser

5.2.4 Bootstrap CSS Low & High-Performance Low Performance

Although JavaScript files are important for web applications, style files (CSS files) are equally important. The fastest loading of the style files of our application is the most important factor for the user to see the correct design. As mentioned in the previous sections, the use of CDN server is also very important for style files.

In the figure below, the loading time of a style file included in the local project and delivered to the user from the same server with the project is 71ms. However, this period is quite long. At the same time, as the user's internet speed drops, the download time will increase.

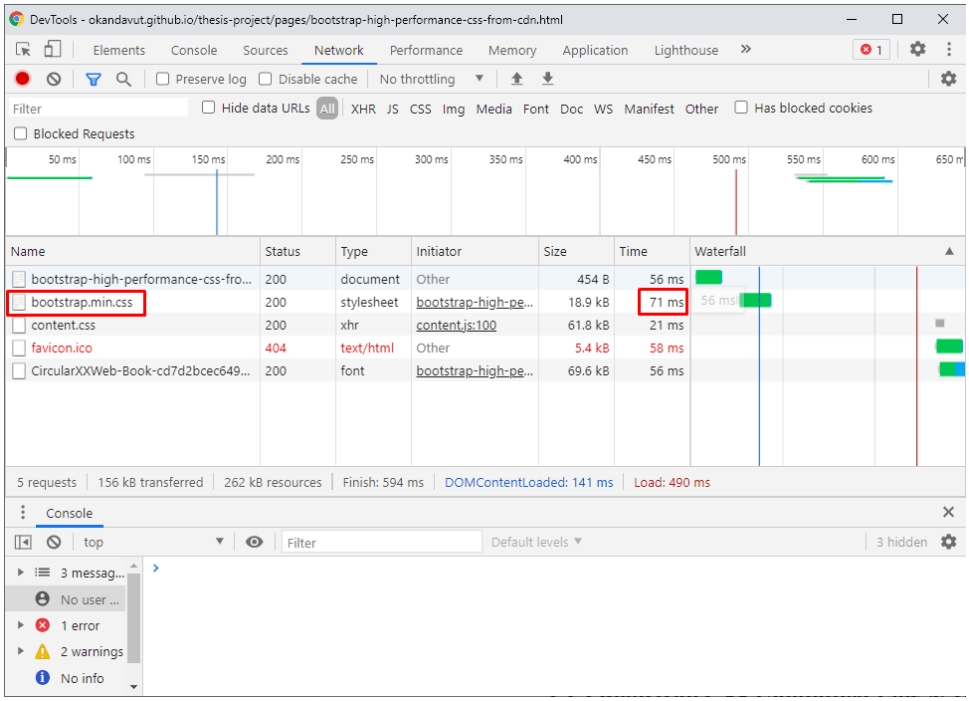

Figure 5.8 Bootstrap minified less performance load time on browser

Looking at the second figure, when the same style file belonging to the Bootstrap library is used on CDN, it is downloaded faster for 62ms. Although the time difference of 9ms seems small here, it is of great importance for an application that contains too many style files and has many users.

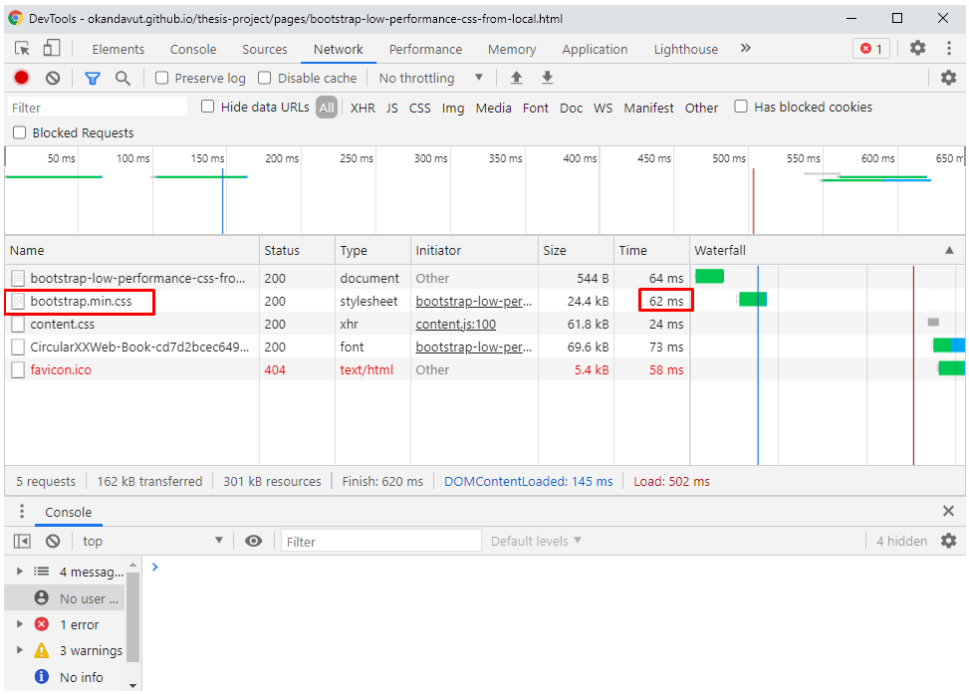

Figure 5.9 Bootstrap minified more performance load time on browser

5.2.5 Bootstrap CSS Unminified Low & High-Performance Low Performance

The process of minification of style files is as important as JavaScript files. Minified CSS files are loaded much faster, while files that are not downloaded and left despite the code are downloaded slower. Therefore, the user reaches the design in the web application later.

The download time specified as 68ms in the figure below belongs to the Bootstrap style library without minifying.

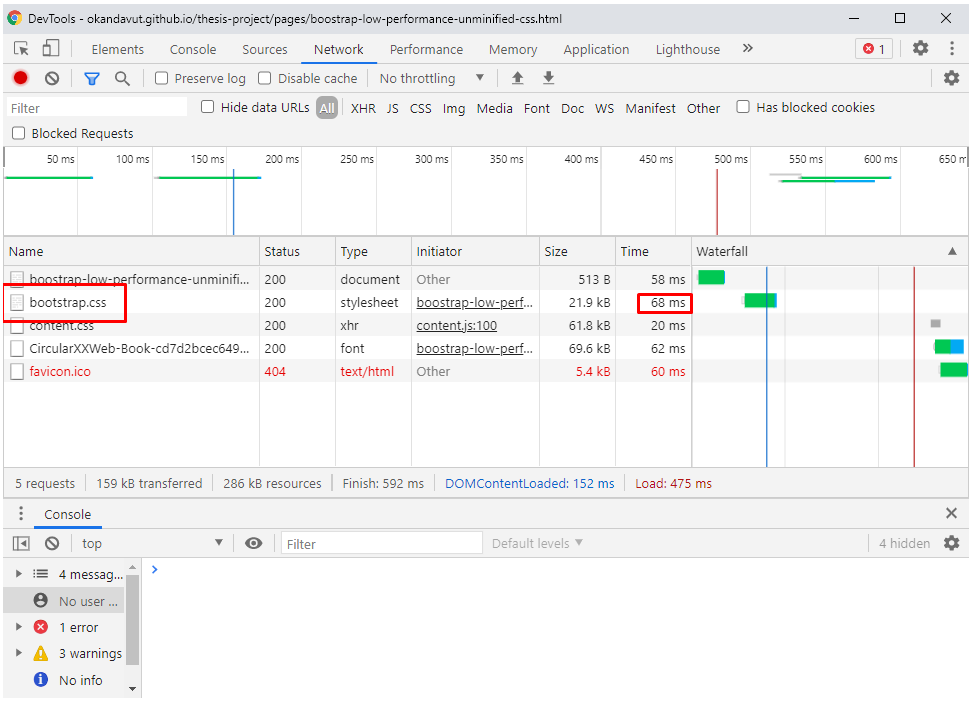

Figure 5.10 Bootstrap unminified low-performance load time on browser

When the next figure is examined, the download time of the minified style file of the Bootstrap library was measured as 59ms. The Minify operation has made a 10ms difference here, but this value will change for different users.

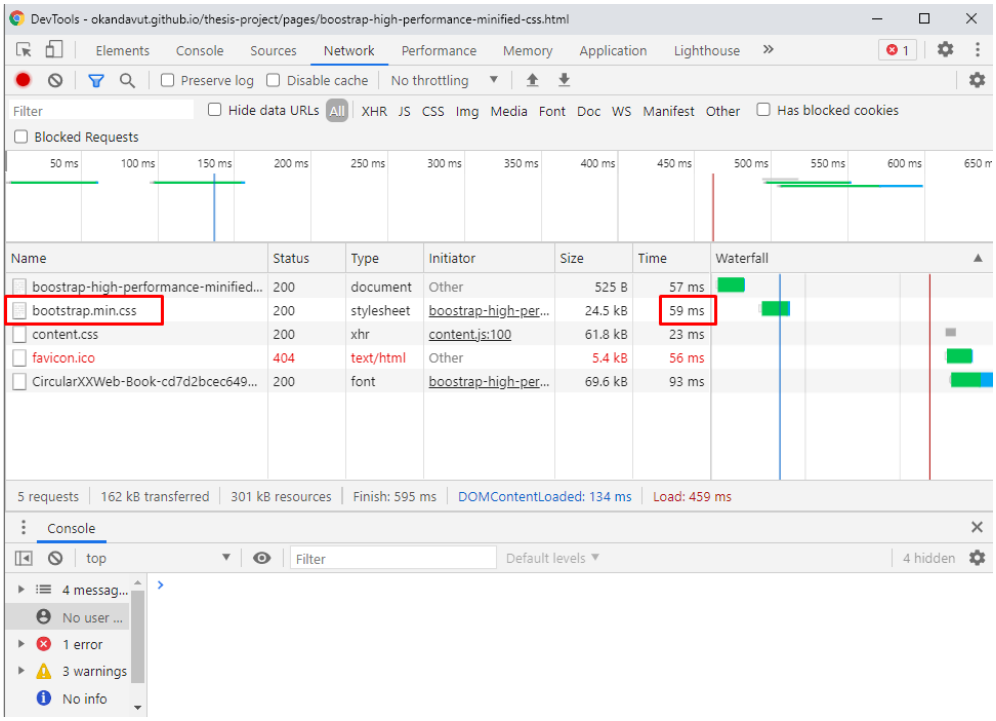

Figure 5.11 Bootstrap minified high-performance load time on browser

5.2.6 Image Sized in HTML and Out Low Performance

When it comes to the last comparison category, the effect of the size in which an image is intended to be used in the web application will be seen on the performance.

Resizing the image to be used in the web application with code in the application is a bad situation in terms of performance. Because the image will be downloaded in its original size but will be displayed with the changed size on the code side. If this sizing process is done by the designer before adding to the code, it means that it will be displayed with a lower size downloaded. Lower size means lower download. This means increased performance and lower download times.

The figure below shows an image file that was originally added with the code but downloaded according to the desired dimensions. The download time is a very high number of 2.59 seconds for the user.

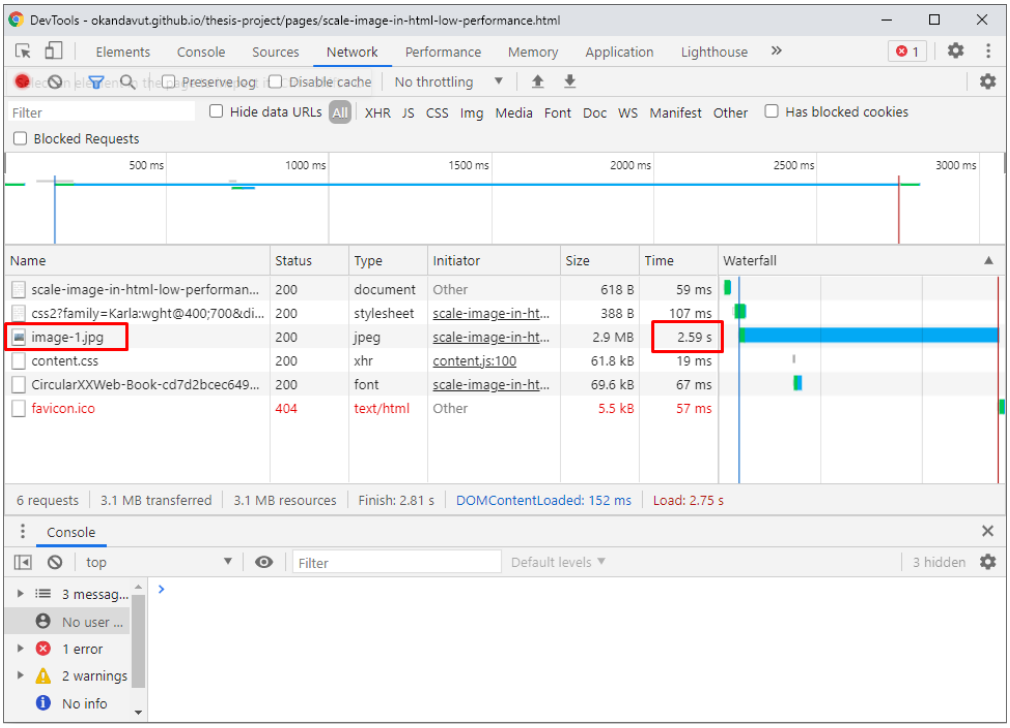

Figure 5.11 Image non-scaled load time on browser

If this file is resized using a different design tool, the loading time will be reduced to 131ms. There is a difference of more than 2 seconds. As mentioned earlier, image files are very critical to performance.

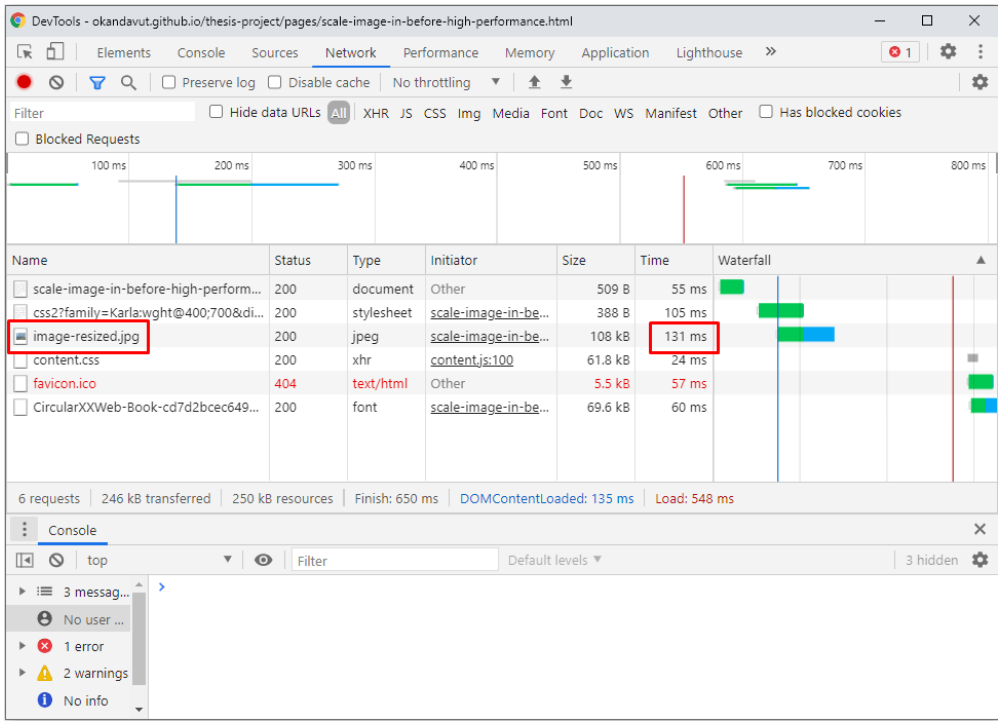

Figure 5.12 Image scaled load time on browser

5.3 Survey on People

To understand the effect of performance on people, the following questions were asked after the pages whose durations were measured were shown to the users. The questions here have been asked to certain age groups.

- 5 people between the ages of 10-20
- 5 people between the ages of 20-30
- 5 people between the ages of 30-40
- 2 people between the ages of 65-75

Survey questions can be reviewed in Appendix A. With these questions and surveys, it will be seen how much importance users attach to performance even in a simple web application. The differences of age groups are also to see the age range that gives more importance.

5.4 Results of Survey

The images below show the results of the questionnaire. As seen in the first graphic, the difference between the pages is reflected in 77.8 percent of the users who participated in the survey. While 11.1 percent of the users did not feel the difference, the same number of users were not sure. Different internet speeds can be considered as a factor here. Most users have noticed the difference.

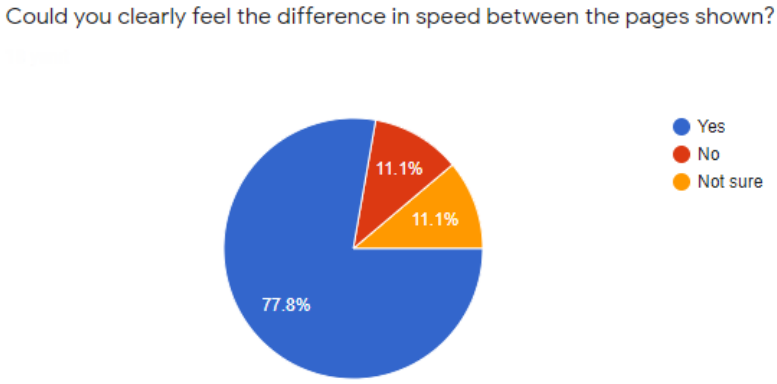

Figure 5.13 Survey first question answer graphic

Coming to the second graph, 100 percent of the users felt the difference in the loading of the image files. From here, it is revealed how important image files are for a web application, regardless of age.

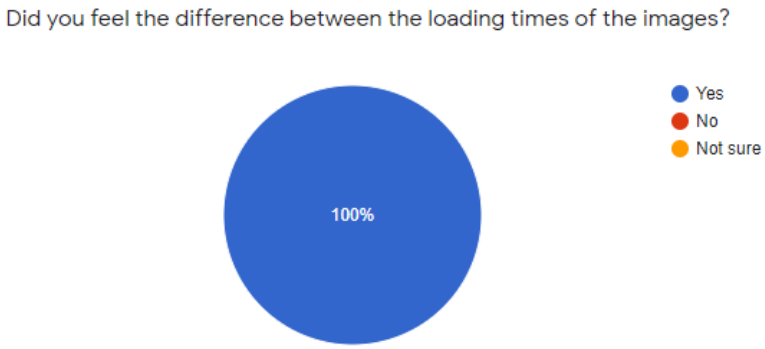

Figure 5.14 Survey second question answer graphic

The third chart conveys the answers about the importance of page speed. Here, 61.1 percent of users paid attention to this issue. This is a huge number. For the remaining 38.9 percent of users, this is not a big deal.

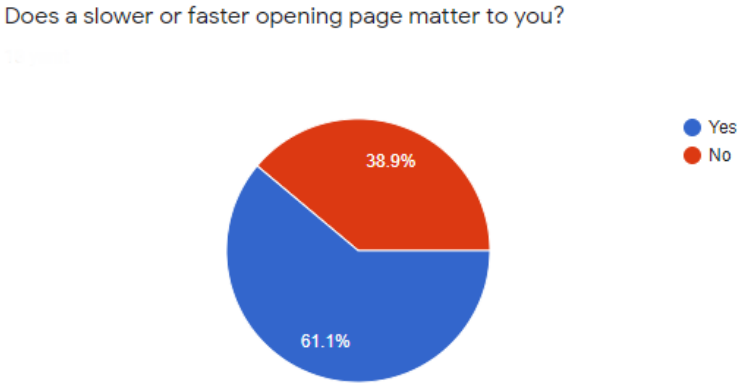

Figure 5.15 Survey third question-answer graphic

The purpose of the last graphic is to access the information which one will prefer when the user is given the right to choose. With this graph, 66.7 percent of users stated that they would choose more performance applications. However, the remaining 33.3 percent of users did not care about this issue.

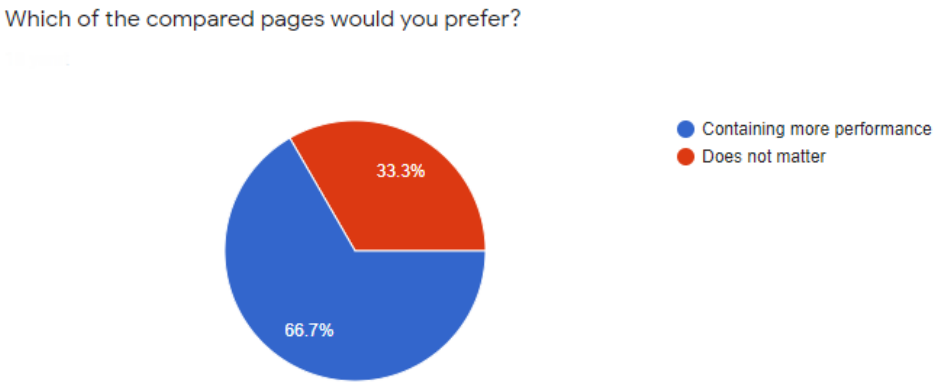

Figure 5.16 Survey fourth question-answer graphic

CHAPTER SIX
CONCLUSION AND FUTURE WORKS

It has been seen that previous studies in the literature have examined performance evaluations in web applications in specific sections. In this thesis, the performance of web applications is discussed with all its aspects. In this way, this study can be used as a guide when it is desired to measure the performance of a web application.

Using performance evaluation criterias, companies and developers can decide which parameters to use for their applications. At the same time, they can easily increase performance in their applications together with performance enhancement processes.

By the project implemented within this thesis, all parameters that affect performance in web applications have been examined. It is stated that which criteria will be taken into consideration while evaluating the performance in web applications. Along with the developed project, examples of slow and fast web applications are given. The methods described in the thesis were handled and the applications with lower and higher performance were compared. The following list shows the effects of the applied processes on web performance in a millisecond.

- Minify Operation
    - jQuery library Minified – 62ms.
    - jQuery Library Unminified – 116ms.
- Image from Content Delivery Network
    - Low performance – 5s.
    - High Performance – 600ms.
- JavaScript CDN & Local
    - CDN – 60ms.
    - Local – 112ms.

The time differences shown above may seem simple for the application of the thesis. However, in large e-commerce applications or different web applications, there are too many file uploads or too many image files. Considering how high the number of users is, even a 1ms difference in loading time is of great importance.

In addition to these, the results of the survey give detailed information about the effect of performance on age groups. Performance is an important criterion for all age groups. This effect also affects the rate at which users use applications.

In the future, following the examination of this thesis, new studies can be made by examining the results of the evaluation and performance improvement processes described in the thesis in web applications with more users. Further, studies can be carried out for performance measurements in web applications with artificial intelligence or different data mining processes.

REFERENCES

Asrese, A. S., Eravuchira, S. J., Bajpai, V., Sarolahti, P., & Ott, J. (2019). Measuring web latency and rendering performance: Method, tools, and longitudinal dataset. *IEEE Transactions on Network and Service Management*, *16*(2), 535-549.

Garrett J. J. (2005). *Ajax: A new approach to web applications*. Retrieved May 15, 2021, from https://www.scriptol.fr/ajax/ajax_adaptive_path.pdf

Gracas R. A. (2020). *Measuring Performance in Network-Intensive Web Applications*. Doctoral dissertation, Faculdade de Ciencias E Tecnologia Universidade Nova De Lisboa, Lisbon.

Jevremovic A., Popovic R., Zivkovic D., Veinovic M., & Shimic G. (2012). Improving web performance by a differencing and merging system. *IJCSI International Journal of Computer Science Issues*, *9*(1), 349-355.

Lumsden A. (2012). *A brief history of the world wide web*. Retrieved May 10, 2021, from https://webdesign.tutsplus.com/articles/a-brief-history-of-the-world-wide-web--webdesign-8710.

Manhas J., (2013). A study of factors affecting websites page loading speed for efficient web performance. *International Journal of Computer Sciences and Engineering*, *1*(3), 32-35.

McPeak A. (2018). *A brief history of web browsers and how they work.* Retrieved April 20, 2021, from https://smartbear.com/blog/history-of-web-browsers/

Rodman T. (2013). *Why your website is slow: image load performance*. Retrieved March 20, 2021, from https://www.yottaa.com/why-your-website-is-slow-image-performance/

Sofi , A. A., & Garg A. (2015). Analysis of various techniques for improving web performance. *International Journal of Advance Research in Computer Science and Management Studies*, *3*(3), 271-277.

Souders S. (2008). *State of performance 2008*. Retrieved May 5, 2021, from https://www.stevesouders.com/blog/2008/12/17/state-of-performance-2008/

Souders S. (2010), *2010 State of performance*. Retrieved June 1, 2021, from https://calendar.perfplanet.com/2010/state-of-performance/

APPENDICES

Appendix 1: Survey Questions

- What is your age?
    - Show number answer.
- Could you clearly feel the difference in speed between the pages shown?
    - Yes
    - No
    - Not sure
- Did you feel the difference between the loading times of the images?
    - Yes
    - No
    - Not Sure
- Does a slower or faster opening page matter to you?
    - Yes
    - No
- Which of the compared pages would you prefer?
    - Containing more performance
    - Does not matter