

Framework for Deploying Web Trustability in Cloud Environment

Amanpreet Singh¹, Amandeep Kaur² and Deepali Gupta³

Submitted: 27/05/2023

Revised: 08/07/2023

Accepted: 26/07/2023

Abstract: Cloud Computing is an online strategy for the dynamic allocation of services by reducing the use of hardware and software resources. Existing models of cloud computing include different components: end-user computers, contact networks, access control frameworks, and software infrastructures. In a cloud environment, one needs to design a framework that calculates trustability. This paper focuses on the different web page components that make the current cloud computing framework more trustworthy. Any website becomes more trustable when the total load time of the web page is less, and the website becomes more secure with the Hypertext Transfer Protocol Secure protocol. In this paper, a new framework has been proposed and deployed in a cloud environment to ensure web trustability. This research showed that page load time was reduced by up to an average of 9.92 seconds when comparing the performance of different websites through the proposed algorithms. A comparison has been made of trustability factors among different websites with various threshold values.

Keywords: Cloud Environment, Page load, Protocol, Institutions, Web Trustability

1. Introduction

Cloud Computing provides dynamic resource allocation in consolidated resources by merging numerous parallel computing, distributed computing, and platform virtualization technologies, the new software system [1]. With cloud computing, an online service provider, users may get services as needed. Chrome Architecture Extensions Within Chrome's multi-process architecture, a single web browser kernel function operates in a position of privilege to access base and system components on behalf of several renderer processes. A web page is running in a tab corresponding to each renderer process. As a result of its sandboxed environment, renderer processes are prevented from directly accessing the network and resources. Only the browser process may get such requests from the network. The architecture for Chrome extensions was created with the presumption that they are useful yet problematic. Virtualization and a dynamic pool of resources are offered by cloud platforms. To effectively manage the resources of the service contributor, load balancing is necessary [2].

The goal of load balancing is to achieve optimal resource usage, reduce average response time, and prevent overload by distributing the workload among several virtual machines on a server via the network. Performance may be enhanced, and better services can be offered to users through the proper utilization of load-balancing techniques in the cloud environment [3]. In this paper, we have covered a wide range of load-balancing methods

utilized in the cloud computing environment. The extension core and content scripts make up the most powerful Chrome extensions. A JavaScript-written content script can be inserted into a web page as it loads. After that, when operating in the renderer process area, it makes use of the Document Object Model tree. In one or more historical contexts, the extension core is made up of Hypertext Markup Language (HTML) and JavaScript web pages that execute in a distinct renderer process. Because it has the fewest rights and is not able to see any object outside of the renderer process zone, the content script must link to the extension core using the Chrome intercommunication between processes [4]. Although it works in a sandboxed environment, the extension core contains most of the enhanced privileges. It cannot, therefore, directly access the host platform and network resources. It can only connect to other external web resources via XMLHttpRequest.

Websites are used in academic institutions, notably in universities and colleges, as a teaching tool and as a tool for online internationalisation. Internet users who connect to software over a Local Area Network (LAN) anticipate the same level of performance. Since the content of the pages is retained locally, websites on LANs often load more quickly. The client-side performance of websites may be evaluated by the amount of time it takes for a page to load. This is the time interval between the user's request for a new page and the browser's completion of rendering a response to a particular request. In other words, they gradually show their content as the browser loads it [5].

For most pages, additional recommended practices may be applied to improve progressive rendering. The user-visible content should be rendered first on fast sites, then

^{1,2,3} Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India.

¹⁾ amanpreet.singh@chitkara.edu.in

²⁾ Corresponding author: amandeeep.bhullar@chitkara.edu.in

³⁾ deepali.gupta@chitkara.edu.in

the off-screen content. Second, a speedy page could render and load text and other small resources before loading larger resources like video and other content. Alternatively, some methods are known to obstruct continuing rendering. Even for layout, some browsers forbid progressive rendering when using large tables. Even though style sheets are not necessary for the first-page load, adding them too late in the content might prevent progressive rendering from continuing [6]. Numerous factors, including bandwidth, browser, geographic access point, and page content, have an impact on the performance of the website. In many cases, it is possible to decrease the amount of data that goes into creating the content of the web page without impairing the page's appearance or operation. Fewer files must be downloaded by the client, which speeds up page loading [7].

2. Proposed Framework and Its Working

Small programs are extensions that provide an option to personalize Chrome to suit the user's requirements or interests. Extensions for Google Chrome are programs that may be installed in Google Chrome to change the performance of the browser. To make things easier, one will also touch on a number of the most well-liked as well as lesser-known yet beneficial Extensions suitable for use with the Chrome browser [8]. In this paper, the new framework is deployed in a cloud environment to check the web's trustability".

Google Extensions are browser extensions that improve browsing. They can improve the effectiveness and user-friendliness of online searches. By merging already-

existing web browser functionalities, the extension can add new functionality while allowing users to perform many tasks at once. Examples of features offered by Google Chrome extensions include ad blocking, password management, and website translation [9].

For a variety of reasons, Google Chrome extensions provide a good development platform. Because it is built on and installed inside the Chrome browser, the Google Chrome Extension has a negligible impact on how many resources are used in the computer system. Users may simply access Chrome, which is one of the most widely used browsers in the world. Second, as most users view and compose emails through browsers, the built-in features of the browser, like extensions, will be the simplest and most reliable method to identify phishing emails. Thirdly, the Google extension will be compact and simple to add to the browser. Because it only requires one click to install, the extension is secure and convenient. Users only need to click once to disable the add-on if they wish to stop it [10].

For developing the Google Chrome extension, first, create the "manifest.json" file that must be kept in the folder initially. The JSON file includes the name, version, description, permissions, and content_scripts of the extension, all the fields are mandatory in the latest version of the manifest file. Links can be provided in content scripts and popup page load file. The code for the "manifest.json" file can be seen in the following Fig. 1. The "java.min.js" and "performance.js" content_scripts is connected in line 10. It links to the "pageload.html" popup script on line 20.

```
1={
2  "name": "FRAMEWORK FOR DEPLOYING WEB TRUSTABILITY",
3  "version": "1",
4
5  "description": "page load time, response time, availability",
6  "permissions": [
7    "activeTab",
8    "storage"
9  ],
10 "content_scripts": [{
11   "matches": ["*://*/**"],
12   "js": ["java.min.js", "performance.js"]
13 }],
14 "manifest_version": 3,
15 "background": {
16   "service_worker": "bg.js"
17 },
18 "action": {
19   "default_title": "popup content",
20   "default_popup": "pageload.html",
21   "default_icon": {
```

Fig. 1: manifest.json file

An extension manifest provides the browser with details about the extension, including the most crucial files and potential functionalities. When a new manifest version is released, the extension platform's functionalities are

updated. Created a file called "manifest.json" and uploaded the folder from any Personal Computer (PC) by going to the Google Chrome extensions developer page and selecting the "Load unpacked" option. The extension

then appeared in the browser and was available for use. The "pageload.html" HTML that users desire to launch when they click the button. A button and instructions may be found in the HTML document. For the detection

function to begin, the user must follow the instructions and press the button. In the following Fig. 2, the popup.js file is included at the end of pageload.html to minimize the load time of the page.

```

53         <span id="loadTotal"></span>
54     </div>
55     <div class="row" id="r-total">
56         <span>Total</span>
57         <span id="total"></span>
58     </div>
59
60
61 </div>
62 <script src="java.min.js"></script>
63 <script src="popup.js"></script>

```

Fig. 2: popup.js included using script tag

After the successful installation of the extension, when clicking on the icon of the extension, the pop-up window is seen in Fig. 3.

Page Load, ssl certificate, request, response time, load event timings in (ms)

Website is Most Trustable

Event	Start	Duration	End
DNS	26	0	26
Connect	26	0	26
Request	49	227	276
Response	276	190	466
DOM	466	2019	2485
Load event	2487	15	2502
Total			2502

Fig. 3: Pop-up window

The above Fig. 3 displays the message "Website is Most Trustable" and other parameters like DNS, Connect, Request of the page, Response Time, DOM, load event, and finally total load time of the page.

The suggested framework for deploying Web Trustability in a cloud environment is shown in Fig. 4. It illustrates how trust management between users and cloud service

providers works. Request time, response time, DNS, Protocol, and Page load time are the main parameters. Below is an explanation of each of these parts. No

framework has been put out to create a solid trust pathway between the user and the resources. A strong approach considers resource security based on utility availability

and user authentication as security measures. The suggested paradigm is represented diagrammatically below, showing the trust channel between the entities.

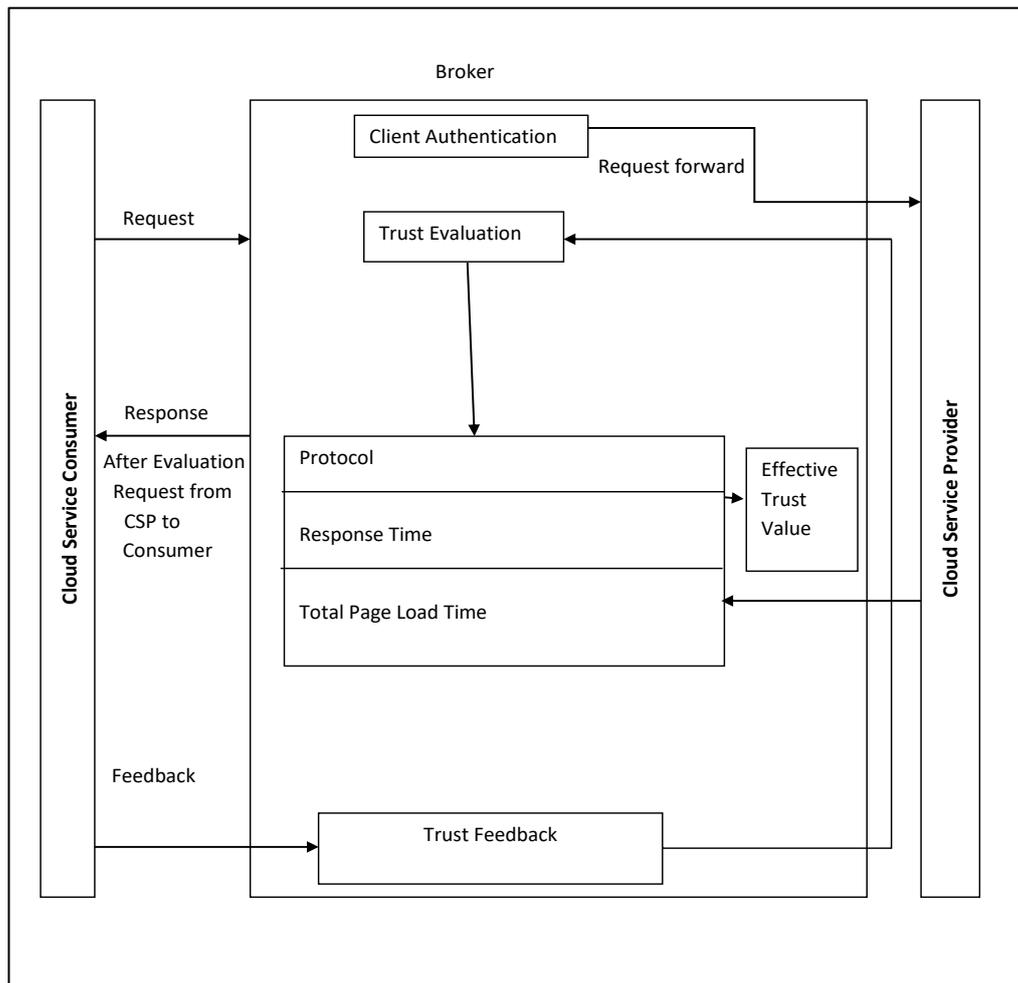


Fig. 4: Framework for Deploying Web Trustability in Cloud Environment

A page load begins whenever the visitor clicks on the link and enters a website address into a browser [12]. This stage is referred to as the initial request or the commencement of navigation. The end user's input sends a network request to the server hosting the website [13]. The request for information is routed to the application to be executed. (It might require a while to complete the request. This might be caused by request waiting). The program finishes its work and sends an HTML output to the client's computer across the network. This is sometimes referred to as the response start [14]. The DOM has finished loading at this moment, which is referred to as DOM ready. In the browser, the user begins rendering the page using the DOM. The page has completed rendering in the user's browser, and the window load event has occurred [15].

3. Methodology of Proposed Framework

It might be challenging at best to establish a reliable method for calculating the load time [11]. The following steps are required to find web Trustability.

- Install the “Framework for Deploying Web Trustability” Google extension on the browser.
- Start the Google Chrome browser and open the test page, then wait for it to load.
- Press Ctrl+Shift+Del to remove temporary files from Chrome, then choose the Cached Images and Files option, and now click on the Clear Data option.
- Type arp -d at the command prompt as an administrator.
- Delete the DNS cache with the command ipconfig /flushdns.
- Refresh the page.
- Now record the response time, request time, Page load time, and protocol of the webpage.
- Repeat the above steps.

Before loading the website on the browser, make sure the cache of the browser is clear if the user wants to check the total page load time accurately. In methodology, we have seen that the user must delete the cache of the browser by going to the option clear browsing history, then checking

the option "Cache images and files," and clicking on clear data. After that, go to the command prompt and type cmd using the run as administrator option. Type the arp -d command, which is used to flush the Address Resolution Protocol cache. Now delete the DNS cache by using the ipconfig /flushdns command. After that, the user is ready to examine Web Trustability, by using the proposed framework.

4. Deploying Web Trustability in Cloud Environment

Register yourself on the Chrome Web Store Developer Dashboard. The following link is used to register: <https://chrome.google.com/webstore/devconsole/register>. Accept the developer agreement and privacy policies by clicking on the checkbox given and pay the one-time registration fee if you want to register. Complete your purchase by adding credit or debit card information. Make Sure the user logs in with a Google account. After the successful payment, the User can add an extension by clicking on the link to the item on the left side menu.

There is a button for the new item on the top right side, click on the button. Afterwards, just upload the zip file that the user developed. At a minimum, the manifest file must include the following fields: The Chrome Web Store and Chrome browser both list this name as their "name", "version", "icons" and "description" [16]. Additionally, your zip file can contain any additional photos and files that the item needs. The ZIP file and the manifest's contents are determined by the particulars of your item.

2.1. Establish a developer account

End Users must choose a developer account to own their goods before publishing them on the Chrome Web Store. After that, set up an account in the Chrome Web Store and first sign up as a Chrome Web Store developer. Instead of utilizing your account, create a new one for a new item. The item must have the same name as your Google Play listing and be owned by the same developer account if you already host it on Google Play and want your Chrome Web Store listing to display an "Available for Android" link [17].

Users can set up management preferences, activate email notifications, and enter information about their developer profile. The website itself has instructions, but there are a few fields that deserve special attention, like Publisher name, Add email, Privacy policy, Physical address, and Trusted tester accounts [18]. Confirm the contact email address when creating a new developer account. Input an email address and then ask for verification by clicking Add Email on the account page. After that, the Chrome Web Store gives that address a verification link, which the user may use to confirm the address.

Use the Chrome Developer Dashboard to upload new items. Upload the item while creating its code, if required, and the item ID to finish the item's code. If not, the user can hold off until the product is complete. Before submitting an item for evaluation, upload it many times. Publish an item after it is complete, including its Chrome Web Store listing and all the websites it depends on [19]. Send an article to be published: The extension will appear as an item on the dashboard once you have uploaded it

2.2. Publicized Your Work

- Verify that the user has filled out all the information on the tabs.
- Push the button labelled "Submit for Review".
- The next dialogue asks you to confirm that you do wish to submit the item for evaluation.

The item will go through a review procedure after being submitted for review. The length of the review will vary depending on the item. Important communications, such as takedown or rejection messages, are automatically enabled [20]. You may activate alerts on the account page to get an email when the item is published or staged.

5. Evaluate and Analyze The Proposed Framework

In this section, the total load time for the various websites is assessed in the cloud environment. The trustability of a webpage is determined based on its total load time, response time, and other parameters:

- If the total load time of the website is more than 0 milliseconds and less than 2999 milliseconds and the protocol is HTTPS, then it indicates that "Website is Most Trustable".
- If the total load time of the website is more than 3000 milliseconds and less than 5999 milliseconds and the protocol is HTTPS, then it indicates that the "Website is Good and Trustable".
- If the total load time of the page is more than 6000 milliseconds and less than 8999 milliseconds and the protocol is HTTPS, then it indicates that "Website is Average but Trustable".
- If the total load time of the page is more than 9000 milliseconds and the protocol is HTTPS, then it indicates that "Website is Bad".
- If the total load time of the page is more than 0 milliseconds and less than 5999 milliseconds and the protocol is HTTPS, then it indicates that "Website is Good but not having SSL Certificate".
- If the total load time of the page is more than 6000 milliseconds and less than 8999 milliseconds and the protocol is HTTP, then it indicates that "Website is Average and not having SSL Certificate".
- If the total load time of the page is more than 8999 milliseconds and the protocol is HTTP, then it

indicates that the “Website is very bad and not Trustable”.

The following tables and figures show the comparisons of various websites and calculate the total Page Load time by using the DNS [21], Connect, Request, Response, DOM, and load event parameters [22].

2.3. Page Load Performance

From following Table I, It has been cleared that when opening the website before removing the cache images and files [96], the total page load time was 3777 milliseconds.

Table I: Total page load time of the webpage before removing the cache

Event Name	Start	Duration	End (ms)
DNS	55	0	55
Connect	55	0	55
Request	128	257	384
Response	384	121	505
DOM	505	3236	3741
Load Event	3745	32	3777
Total			3777

ms= milliseconds

The following Fig. 5 shows the message “Website is Good and Trustable” using the proposed framework.

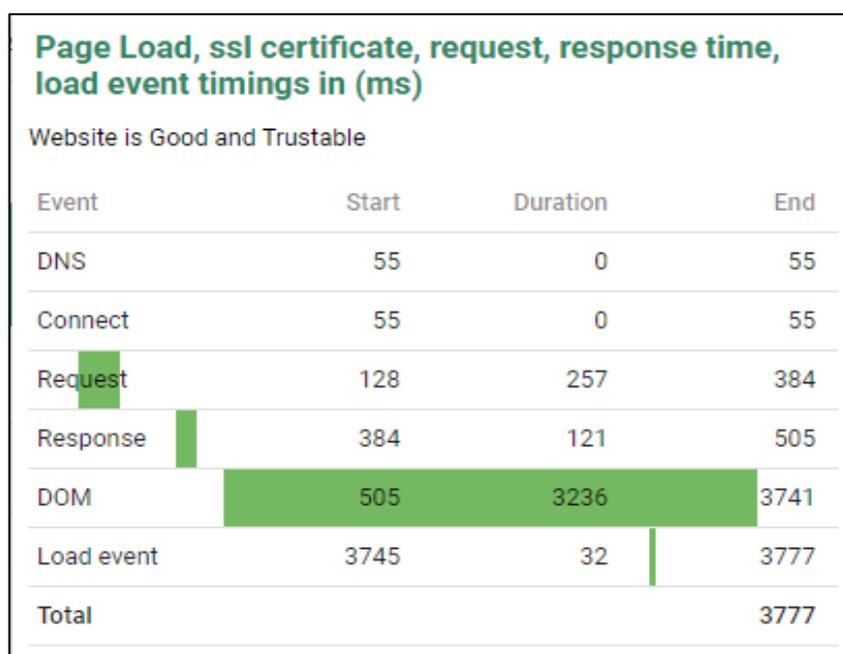


Fig. 5 Total Load time of website before removing cache

In the following Table II, the actual time taken to load the webpage is measured after removing the cache images and files. Now the total page load time value of the website

shows 2136 seconds. Other parameters like DNS, Connect, Request, Response, and DOM are also calculated.

Table II Total page load time of the website after removing the cache

Event Name	Start	Duration	End (ms)
DNS	4	0	4
Connect	4	0	4

Request	23	208	231
Response	231	116	347
DOM	347	1775	2122
Load Event	2123	13	2136
Total			2136

ms = milliseconds

The following Fig. 6 shows the message “Website is Most Trustable” after clearing the cache from the browser. The page loading timing reduces, hence the web trustability increases.

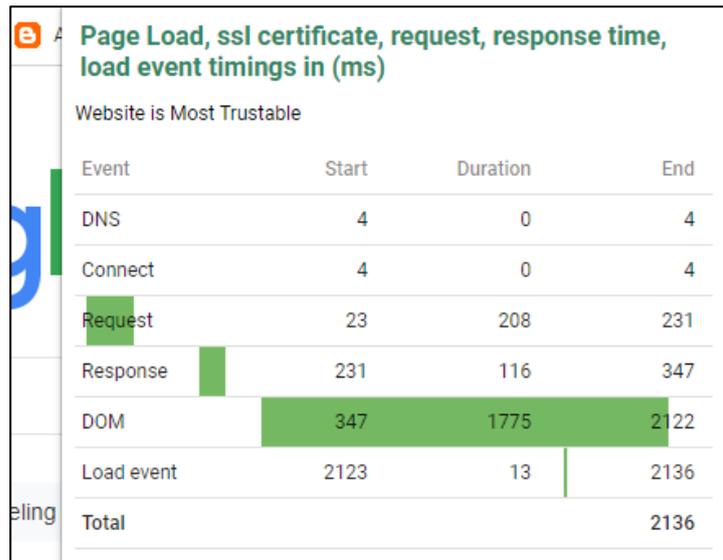


Fig. 6 Total page load time of the website after removing the cache

2.4. Comparative Analysis Between Total Page Load Time, Response Time, DOM, and Protocol

The following Fig. 7, shows the total page load time, Response time, and DOM time before and after clearing cached images and files in milliseconds for the website.

There is a difference between page load time before and after clearing the cache from the browser. Similarly, response time also reduces to 347 after clearing the cached images and files. The DOM time value was reduced to 2122. Hence the Web trustability of the website increases.

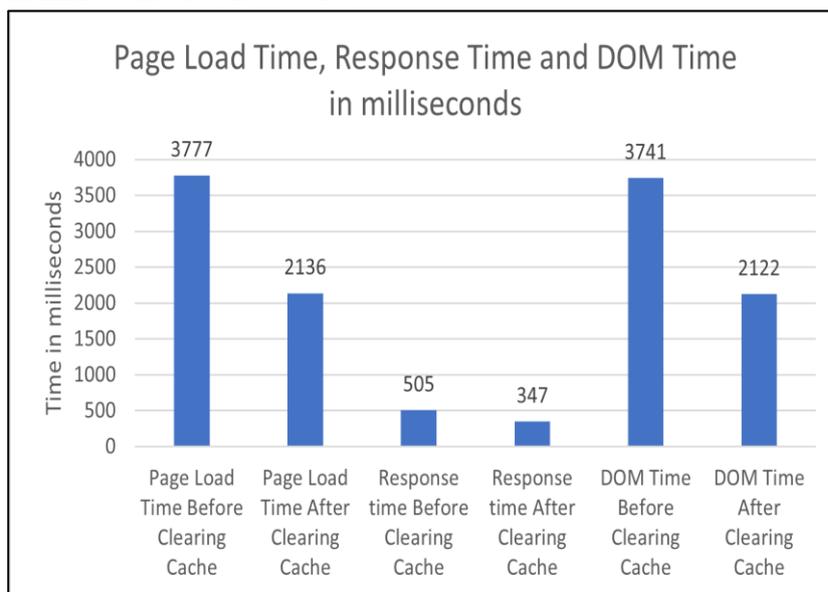


Fig. 7: Page Load time in milliseconds

2.5. Comparison of Proposed Framework With The Existing Tools

In the following Fig. 8, the total load time is shown as 3.71 seconds when examined the website on the Pingdom website speed test [23].

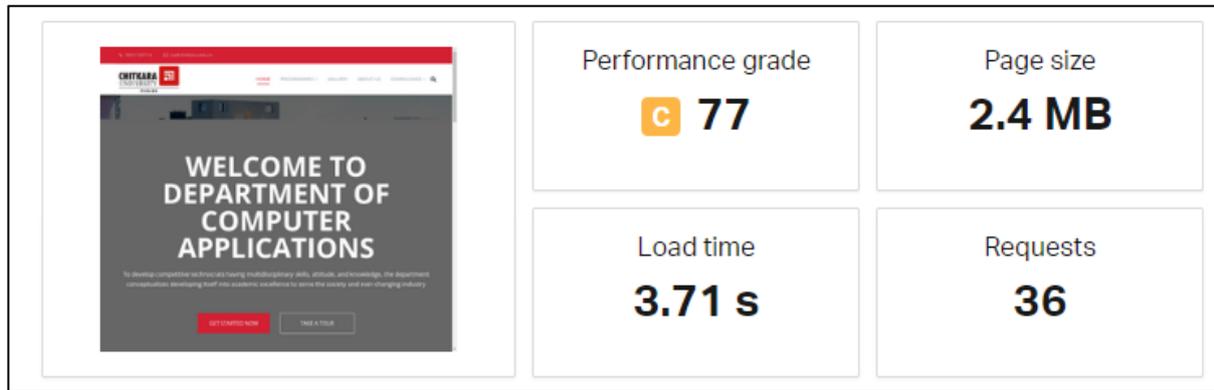


Fig. 8: Load time with Pingdom Website Speed Test [23]

In the following Fig. 9, JMeter showed the total Page Load [24] of the website was 2687 milliseconds.

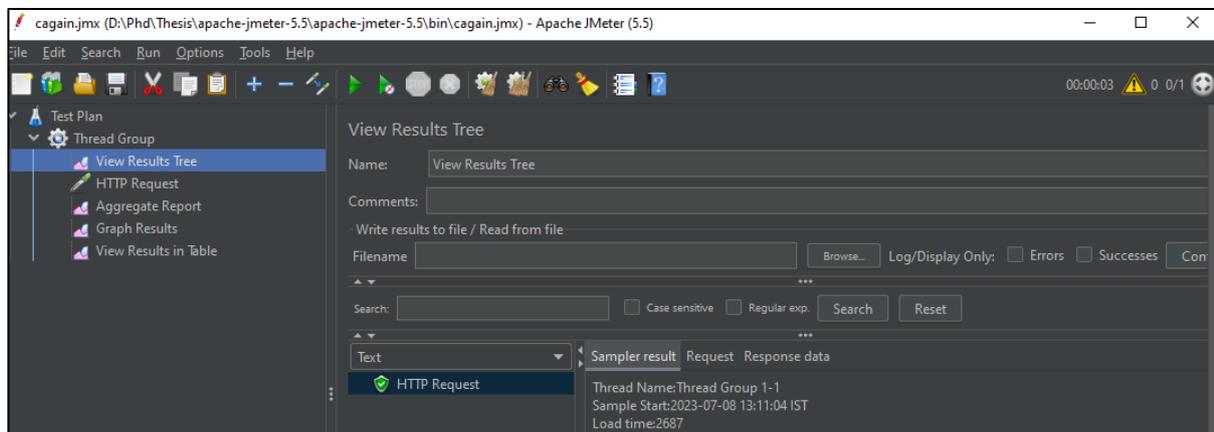


Fig. 9: Page Load time with JMeter [24]

For Pagespeed Insights, the following Fig. 10 showed the total speed index [25] (load time) was 11.9 seconds.

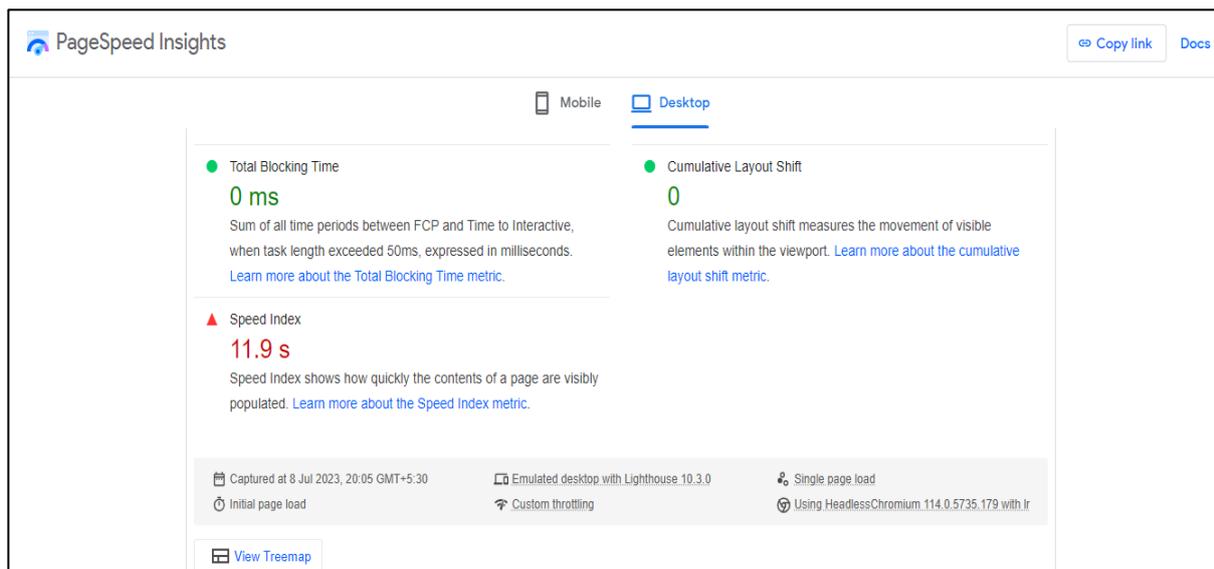


Fig. 10: Load time Using PageSpeed Insights [25]

For the Lighthouse Report, the following Fig. 11 shows the total speed index [26] (load time) was 8.9 seconds.

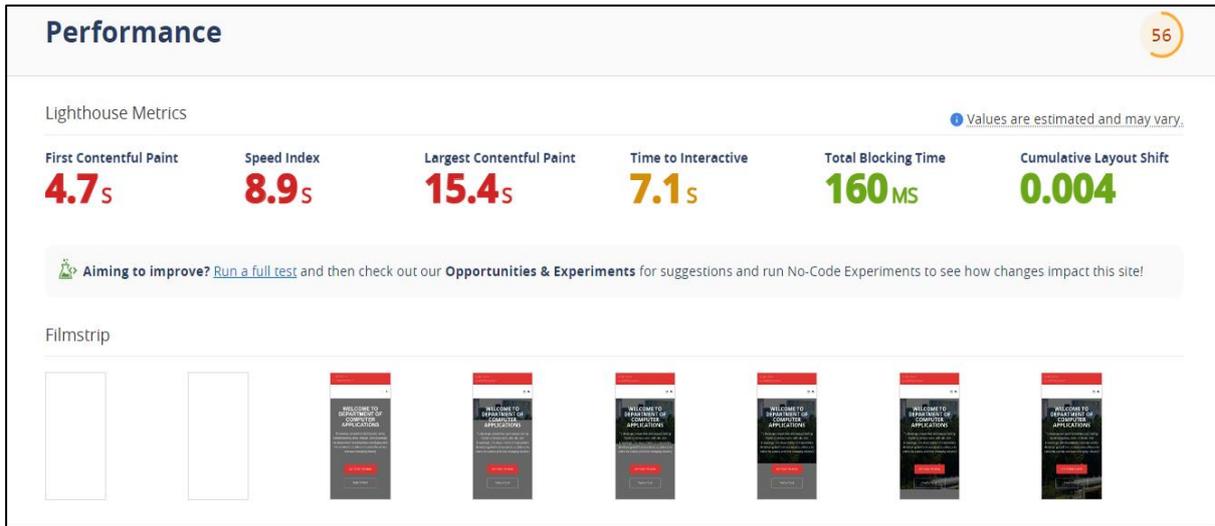


Fig. 11: Load time Using Lighthouse Report [26]

The following Fig. 12 showed Uptrends Report, the total page load time was 6.8 seconds

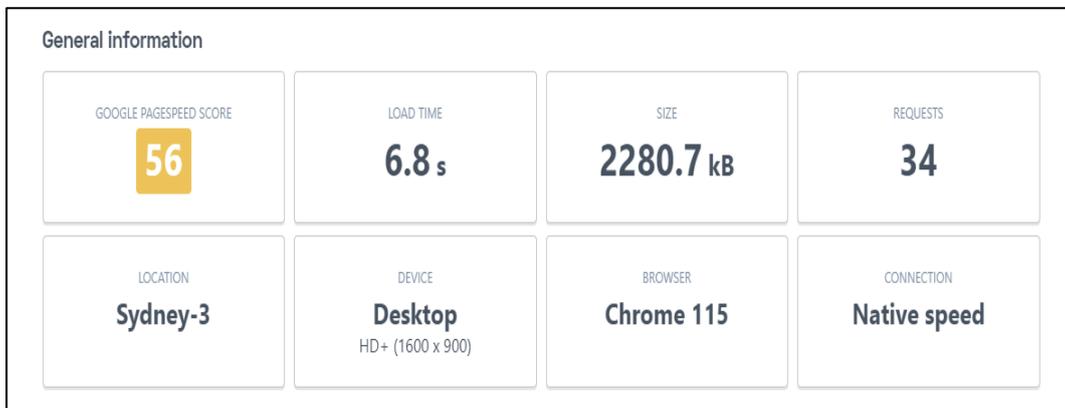


Fig. 12 Uptrends Report [27]

The following Fig. 13 calculated the report of webpagetest, total speed index was 3.864 seconds.

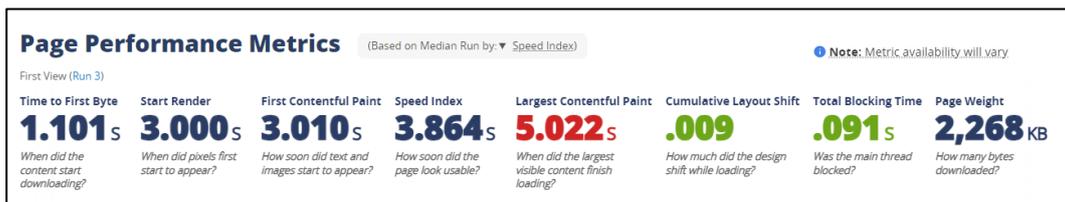


Fig. 13 Webpagetest Report [28]

From the above figures, the Total load time of the page with the Pingdom Website Speed Test was 3.71 seconds. For the JMeter, the total load time of the page was 2.687 seconds. For PageSpeed Insights, the speed index (load time) was 11.9 seconds. For LightHouse Report [26] the

speed index of the page at 8.9 seconds. According to Uptrends Report [27], the load time was 6.8 seconds. As per the webpagetest tool report [28], the speed index was 3.864 seconds. But the proposed framework showed better results, that is, 1.98 seconds (refer to Fig. 14).

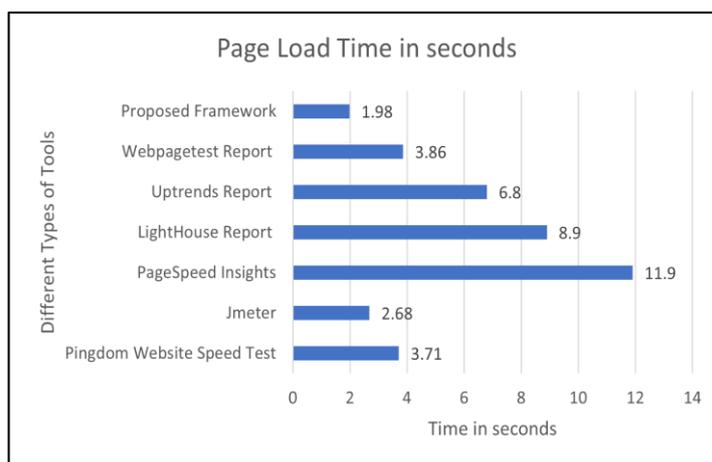


Fig. 14: Comparison of Total Page Load Time in seconds

Various websites have been analysed, and the page load time of the various subjects are as follows: For Youtube, the page load time of the proposed framework shows 4518 milliseconds, for Amazon it was 4969 milliseconds, for Flipkart it was 4736 milliseconds, for Wikipedia it was 3360 milliseconds, for Google Scholar it was 1902

milliseconds, for Facebook it was 3204 milliseconds, for Panjab University, it was 4299 milliseconds, and for I.K. Gujral Punjab Technical University Jalandhar it was 7097 milliseconds. The page load time shown in Fig. 15 was according to the methodology in the Proposed Framework.

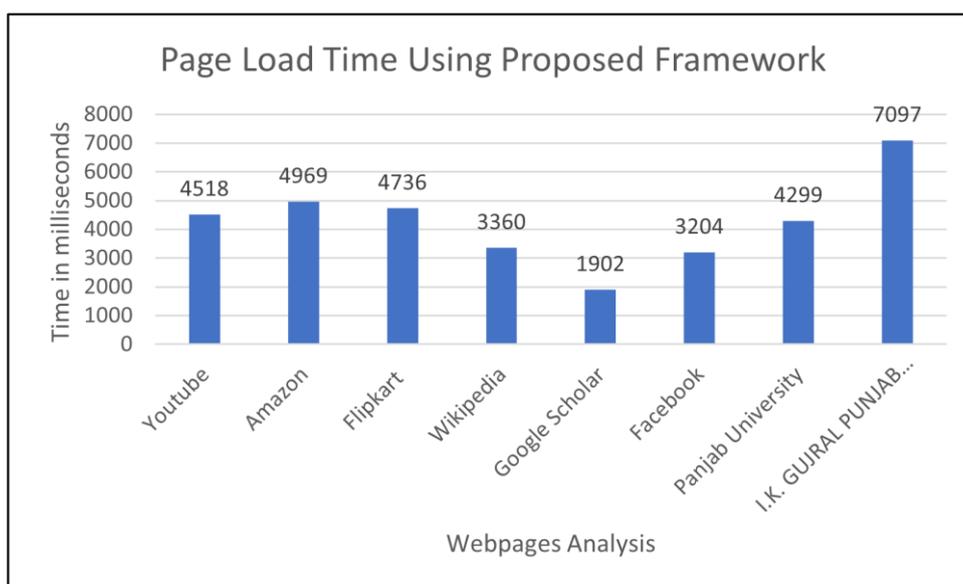


Fig. 15 Comparison of the page load time of various websites

Detailed analysis of the websites has been done and has concluded that if compared with the existing tools in the market, the proposed framework shows better results.

Another parameter is website protocol; if the websites have the protocol HTTPS, it means that they have SSL certificates. These websites are more reliable and secure, hence more trustworthy. On the other hand, if the website protocol is HTTP, that means it has no SSL certificate; this means that the website is not reliable and less trustworthy.

6. Conclusion

This paper focuses on the different web page components that make current cloud computing models more

trustworthy. Any website becomes more trustworthy when the total load time of the web page is less, and the website becomes more secure with the Hypertext Transfer Protocol Secure (HTTPS) protocol. In this research, a new framework has been proposed and deployed in a cloud environment to ensure web trustability. This research showed that page load time was reduced by up to an average of 9.92 seconds when comparing the performance of different websites through the proposed algorithms. A comparison has been made of trustability factors among different websites with various threshold values. Web Trustability was determined by examining the QoS metrics of several websites in terms of protocol, DNS, request time, response time, DOM, and overall page load

time. While compared with other tools like Pingdom Website Speed Test, PageSpeed Insights, LightHouse Report, Uptrends Report, and Webpagetest Report, JMeter still shows better results. The customized Google extension is deployed in a cloud environment and evaluates the web trustability of websites.

References

- [1] Y. Ruan, and A. Durresi, "A trust management framework for clouds", *Computer Communications*, vol. 144, ISSN 0140-3664, 2019, pp. 124-131.
- [2] H. Chen, and M. Hossain, "Developing a Google Chrome Extension for Detecting Phishing Emails", *EPiC Series in Computing*, vol. 77, 2021, pp. 13-22.
- [3] L. Abrams, "What are Google Chrome Extensions?", April 2017, [Online] Available:<https://www.bleepingcomputer.com/tutorials/understanding-google-chrome-extensions/>.
- [4] O. Moris, "Extending Knowledge: Learn More about Google Extensions", 2019.
- [5] R. Hidayanto and P. Sawitri, "Performance Testing of e-Payment Website Using JMeter", *International Research Journal of Advanced Engineering and Science*, vol. 4(3), 2019, pp. 350-352.
- [6] P. Mehta, "Introduction to Google Chrome Extensions," *Creating Google Chrome Extensions*, Apress, Berkeley, CA, pp.1-33, 2016. [Online]. Available: https://doi.org/10.1007/978-1-4842-1775-7_1.
- [7] T. Bhardwaj, H. Upadhyay, and S. C. Sharma, "Framework for quality ranking of components in cloud computing: regressive rank", In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020 January, pp. 598-604, IEEE.
- [8] CodeIT, "The Importance of Chrome Extension Development," April 2018. [Online]. Available: https://medium.com/@codeit_llc/the-importance-of-chrome-extension-developmentb8cbb9405bf2.
- [9] Y. Akgül, "Accessibility, usability, quality performance, and readability evaluation of university websites of Turkey: a comparative study of state and private universities", *Universal access in the information society*, 20(1), 2021, pp. 157-170.
- [10] P. De Boeck and M. Jeon, "An overview of models for Response Times and processes in cognitive tests", *Frontiers in psychology*, 10, 2019, pp. 102.
- [11] B. Csontos and I. Heckl, "Accessibility, usability, and security evaluation of Hungarian government websites", *Universal Access in the Information Society*, 20, 2021, pp. 139-156.
- [12] D. Norris, J. Hall, S. Butterfield and M.P. Page, "The effect of processing load on loss of information from short-term memory", *Memory*, 27(2), 2019, pp.192-197.
- [13] J. Van Bulck, D. Moghimi, M. Schwarz, M. Lippi, M. Minkin, D. Genkin, Y. Yarom, B. Sunar, D. Gruss and F. Piessens, "LVI: Hijacking transient execution through microarchitectural load value injection", In 2020 IEEE Symposium on Security and Privacy (SP), 2020, May, pp. 54-72. IEEE.
- [14] M. Kaur and R. Aron, "Focalb: Fog computing architecture of load balancing for scientific workflow applications", *Journal of Grid Computing*, 19(4), 2021, p.40.
- [15] N.P. Hoang, A.A. Niaki, P. Gill and M. Polychronakis, "Domain name encryption is not enough: privacy leakage via IP-based website fingerprinting", 2021, arXiv preprint arXiv:2102.08332
- [16] E. Uzun, "A novel web scraping approach using the additional information obtained from web pages". *IEEE Access*, 8, 2020, pp. 61726-61740.
- [17] D. Giomelakis and A. Veglis, "SEO and Web Analytics in Journalism: Case Studies from the Greek News Media Landscape", In *The Emerald Handbook of Digital Media in Greece*, 2020, pp. 51-65, Emerald Publishing Limited.
- [18] J. Gong and T. Wang, "Zero-delay lightweight defenses against website fingerprinting", In 29th USENIX Security Symposium, USENIX Security 20, 2020, pp. 717-734.
- [19] S. Lei and R. Law, "Functionality evaluation of mobile hotel websites in the m-commerce era", *Journal of Travel & Tourism Marketing*, 36(6), 2019, pp. 665-678.
- [20] S. Henri, G. García, P. Serrano, A. Banchs and P. Thiran, "Protecting against website fingerprinting with multihoming", *Proceedings on Privacy Enhancing Technologies*, 2020(2), 2020, pp. 89-110.
- [21] N. Khattar, J. Sidhu and J. Singh, "Toward energy-efficient cloud computing: a survey of dynamic power management and heuristics-based optimization techniques", *The Journal of Supercomputing*, 75, 2019, pp. 4750-4810.
- [22] M. S. U. Islam, A. Kumar and Y. C. Hu, "Context-aware scheduling in Fog computing: A survey, taxonomy, challenges and future directions", *Journal of Network and Computer Applications*, 180, 2021, pp. 103008.
- [23] S. Yason and A. Yunus, "ANALISIS PERFORMA WEBSITE SCLEAN MENGGUNAKAN PINGDOM TOOLS DAN PAGE SPEED INSIGHTS", *Kharisma Tech*, 17(1), 2022, pp. 113-124.

- [24] J. Wang and J. Wu, "Research on performance automation testing technology based on JMeter", In 2019 International Conference on Robots & Intelligent System (ICRIS), 2019, June, pp. 55-58, IEEE.
- [25] N. Q. Haeruddin, M. R. Faizal and S. H. Baharuddin, "ANALISIS KINERJA WEBSITE PARAMA PELINDO MENGGUNAKAN PINGDOM TOOLS DAN PAGESPEED INSIGHTS", *Jurnal Informatika Progres*, 15(1), 2023, pp. 33-40.
- [26] M. Siahaan and V. O. Vianto, "Comparative Analysis Study of Front-End JavaScript Frameworks Performance Using Lighthouse Tool", *Jurnal Mantik*, 6(3), 2022, pp. 2462-2468.
- [27] M. Kinnunen, "Evaluating and improving Web performance using free-to-use tools", (Master's thesis, M. Kinnunen), 2020.
- [28] M. T. Hossain, R. Hassan, M. Amjad, and M. A. Rahman, "Web Performance Analysis: An Empirical Analysis of E-Commerce Sites in Bangladesh", *International Journal of Information Engineering & Electronic Business*, 13(4).
- [29] Verma, R. ., Dhanda, N. ., & Nagar, V. . (2023). Analysing the Security Aspects of IoT using Blockchain and Cryptographic Algorithms. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(1s), 13–22. <https://doi.org/10.17762/ijritcc.v11i1s.5990>
- [30] Prof. Nikhil Surkar. (2015). Design and Analysis of Optimized Fin-FETs. *International Journal of New Practices in Management and Engineering*, 4(04), 01 - 06. Retrieved from <http://ijnpme.org/index.php/IJNPME/article/view/39>
- [31] Anupong, W., Azhagumurugan, R., Sahay, K. B., Dhabliya, D., Kumar, R., & Vijendra Babu, D. (2022). Towards a high precision in AMI-based smart meters and new technologies in the smart grid. *Sustainable Computing: Informatics and Systems*, 35 doi:10.1016/j.suscom.2022.100690