Towards Improving Accessibility of Web Auditing with Google Lighthouse

1st Thomas McGill
Web Developer
Forrit
Edinburgh, United Kingdom
ThomasMcGill1996@gmail.com

2nd Oluwaseun Bamgboye
Sch. of Computing, Engineering and Built Environment
Edinburgh Napier University
Edinburgh, United Kingdom
O.Bamgboye@napier.ac.uk

3rdXiaodong Liu
3rd Xiaodong Liu
Sch. of Computing, Engineering and Built Environment
Edinburgh Napier University
Edinburgh, United Kingdom
X.Liu@napier.ac.uk

4th Chathuranga Sampath Kalutharage
Sch. of Computing, Engineering and Built Environment
Edinburgh Napier University
Edinburgh, United Kingdom
chathurangasampath.kalutharage@napier.ac.uk

***Abstract*—Google Lighthouse is a tool made by Google for auditing web pages performance, accessibility, SEO, and best practices with the intention of improving the quality of the websites. This allows software developers to understand areas of improvement within a website. However, accessibility becomes an issue when it comes to rapidly handling multiple input for generating performance report consisting different formats with better accuracy. In this work, a test on set of improvements to enhance the reference Google Lighthouse architecture was conducted to enable it to process multiple file input types with the focus of producing different output format including better accuracy. Multiple experimental runs involving different size of file input were conducted by comparing the results from using the modified Lighthouse with the original Lighthouse. The results proves the feasibility of the enhancement and can further improve the accuracy of performance audit report with known Lighthouse performance metrics.**

***Index Terms*—Accessibility, Google Lighthouse, Web Audit, Performance Metrics, Web Performance**

I. Introduction

Web auditing and performance testing is an important aspect of software practice, and it is often achieved using one of the most popular tool such as the Google Lighthouse. Google Lighthouse is a tool made by Google for auditing web pages using metrics such as performance, accessibility, Search Engine Optimisation (SEO), and best practices. This tool also allows software developers to understand areas of improvement needed on the web applications. As an open-source tool for developers, it is used for performance measurements and accessibility auditing to improve the quality of website's traffic including user experience and search results rankings [1] [2].

However, the issue comes from weakness of its accessibility point of view that tends to slow down the overall performance when running audit report from large websites. This is because the tool is able to handle a single input, mainly in json format and comes with limited output type when producing the performance reports of web analysis. There has been a growing need for the Google Lighthouse to support multiple input types and report output [3]. Also, the unnecessary time it takes the Google Lighthouse in the running the report of a single page from a website rather the whole pages of a website is considered as one of the issues resulting from weak accessibility and performance. Based on the wide acceptability and popularity of Google Lighthouse for providing insight into a user's experience with metrics for performance determining speed and accessibility audit, it has been concluded that the tool as well as related categories still requires further improvements [4].

The aim of this paper is to provide an enhancement to Google Lighthouse [5] by extending the reference architecture with API extension that makes the tool more accessible and usable by software practitioners for various accessibility audit and improvement of website performance. Thus us followed by series of test to ascertain the feasibility of the enhancement. The architectural enhancement enable Google lighthouse with capability to produce more detailed accessibility audit report that reflect the actual error points, thereby making changes to be possible in a fast manner with web traffic. In addition, the tool will be able to accept multiple input types (including files, URLs and sitemap URL) when running the performance report of web apps. Finally, the extension allows to export multiple report formats (such as Json, csv and XML) thereby making it even more accessible to software practitioners and developers trying to improve the website speed and increase traffic from Google search engines.

II. Literature Review

The large community of web developer have continue to rely on the use of Google Lighthouse for the identification and fixing of issues related to web performance. The suitability

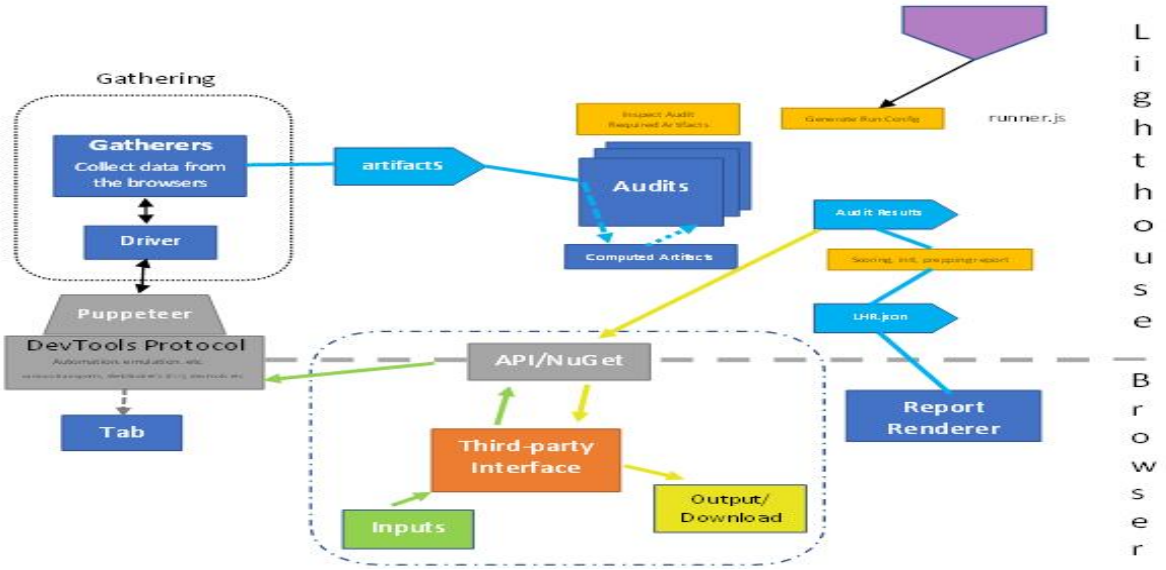

Fig. 1. The modified Architecture.

of Google Lighthouse to support the tasks has made it to be recognised as a widely acceptable tool for tracking web page metrics. However, it is well recognised that the tool can also render some poor performance under certain situation that may allow developer not consider as a suitable option for performance tracking [6].

There have been effort to maintain high performance of websites with some degree of modification to Lighthouse. One step in that direction was the decision to rebuild some of the default components of Lighthouse tool in attempt to increase page performance through accessibility [7]. The improvements allow the total sign-ups by users to increase by 15% while reducing the response time search engine ranking by 40%. The improvement is promising but not yet standardised as it was only tailored for customised usage with little support for complete Lighthouse auditing metrics.

In another application [8], Google Lighthouse has been used to compare the performance of two JavaScript libraries (ReactJs and VueJs) using the speed metrics while evaluating the web pages. The work clearly indicate that Google Lighthouse is the appropriate tool for the task and does achieve the purpose of testing multiple web pages, but does not clearly suggest any significant improvement to accessibility issues that relates to processing multiple file inputs and audit reports.

III. System Overview

The modified architecture presented in figure 2 is based on the Google Lighthouse reference architecture [5]. The architecture is originally built on the Gathering and Auditing Module with the Puppeteer and DevTools components. The modification comes with an additional module consisting a set of additional API/NuGet module that communicates directly with the browser and related web applications.

The API/NuGet assists with the flow of input supplied via a third interface built on top of the google browser. The third party interface consists of the input and output module respectively, which tries to improve LightHouse accessibility of file input/output during the performance audit process. The input module can handle any combination of inputs to maximise input accessibility during the auditing process. The input types can include multiple URLs, file inputs (such as CSV, json and XML) or a sitemap URL. The API then sends the input generated from the third party interface to DevTools Protocol which eventually passes it into the Gathering module via the Driver.

The DevTools Protocol interface the Driver through a Puppeteer and consists of a collection of APIs. Part of the API maintains the client state that is attached to an active remote chrome instance during debugging. The gatherer within the Gathering module also uses the Driver to collects relevant information about the web page based on the metrics and later render this as artefact in the Auditing module.

The auditing Module consists of the Audit unit that checks and asserts the computed artefact received from the Gathering module satisfies the required metrics or optimisation using the scoring method. The audits are tests for a single feature/optimization/metric. Using the Artifacts as input, an audit evaluates a test and resolves to a numeric score. The audit report is received by the output module of the Third Party Interface. The user will normally be able to user in other format (such as csv and XML formats) apart from the traditional json format.

During the implementation, file input and output process-

Fig. 2. Functional Requirements

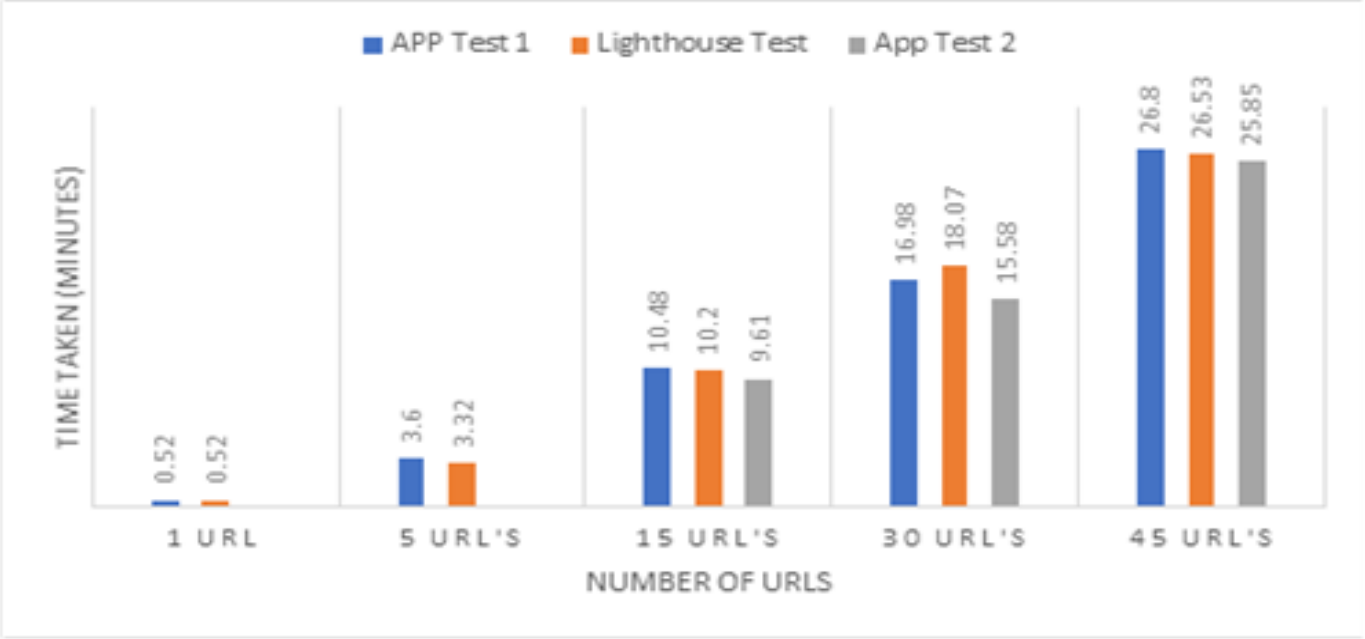

Fig. 3. Comparison of Execution Time

ing requirements were set out as part of the functional requirements in order to achieve the enhancement to Google Lighthouse. These requirements were presented in Figure 2. The requirements have been implemented using the appropriate architectural pattern (model-view-controller) that provides some degree of flexibility and fast development process.

IV. Experiment

An experiment involving repeated testing in an appropriate test environment were conducted with Google's Lighthouse to determine if the enhancement involving the additional API implemented are effective. Effectiveness will be determined by the actual time taken by the enhanced architecture to perform the full performance audits on selected websites in each experimental run of testing. The effectiveness of both the input and output module of the third-party interface added as enhancement to Google Lighthouse were considered during testing

The testing environment was designed to try ensuring as non-biased a test as possible. External influences are min-

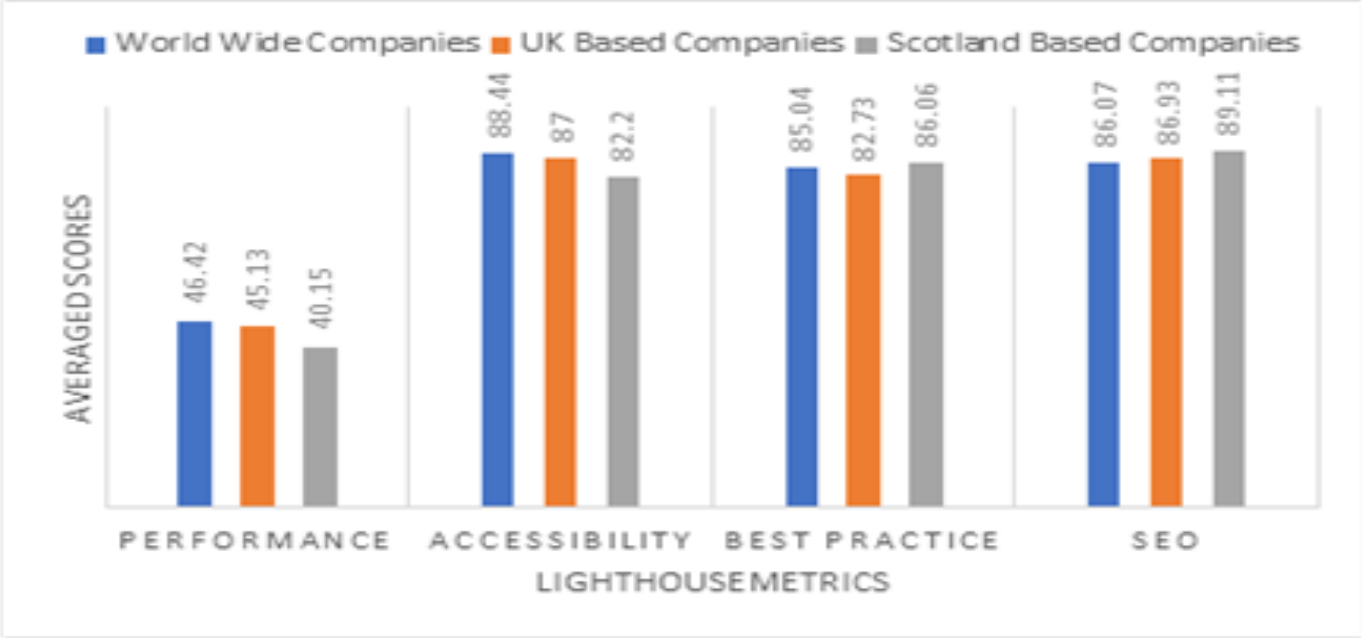

Fig. 4. Accuracy of Google Lighthouse Metrics

TABLE I
TEST RESULTS FROM ENHANCED ARCHITECTURE

URL Test Input	No of URLs	Time Taken mins	Avg. Performance Score	Avg. Accessibility Score	Avg. Best Practice Score	Avg. SEO Score	No of Download
1 URL TEST	1	0.31	N/A	N/A	N/A	N/A	1
	1	0.34	N/A	N/A	N/A	N/A	1
	1	0.29	N/A	N/A	N/A	N/A	1
5 URL TEST	5	03.20	N/A	N/A	N/A	N/A	5 + 3FULL
	5	03.16	N/A	N/A	N/A	N/A	5 + 3FULL
	5	03.22	N/A	N/A	N/A	N/A	5 + 3FULL
15 URL Test Worldwide Companies	15	13.24	43.13	88.27	84.87	86.07	15 + 3FULL
	15	10.48	48.07	88.6	84.87	86.07	15 + 3FULL
	15	11.34	48.07	88.47	85.4	86.07	15 + 3FULL
15 URL Test UK Based Companies	15	08.41	44.67	87	82.73	86.93	15 + 3FULL
	15	09.46	45.73	87	82.73	86.93	15 + 3FULL
	15	09.13	45	87	82.73	86.93	15 + 3FULL
15 URL Test Scotland-Based Companies	15	10.17	40.93	82	86.2	89.07	15 + 3FULL
	15	10.17	40.80	82	86.2	89.2	15 + 3FULL
	15	10.24	38.73	82.6	85.8	89.07	15 + 3FULL
30 URL Test	30	16.17	44.87	87.7	83.6	86.47	5 + 3FULL
	30	17.19	45.7	87.77	83.33	86.5	5 + 3FULL
	30	17.23	43.23	87.8	83.33	86.5	5 + 3FULL
45 URL Test	45	26.30	44.11	86.07	85.2	87.11	3FULL
	45	26.01	45.64	85.87	85.2	87.04	3FULL
	45	27.40	44.24	85.71	85.18	87.09	3FULL

imised as much as possible during the experiments. All the tests were conducted on a stable broadband connection with uninterrupted router device. The audit was performed on the selected websites using Google Lighthouse v7.3.0 on Windows 10 Enterprise with 11th Generation processor (i3-1115G4 @ 3.00GHz). There are three independent experimental runs for each experiment with each consisting of different size for multiple inputs of URLs (1, 5, 15, 30, 45). The experiment with the enhanced architecture was repeated to validate the result of the individual run. The URLs are consists of dynamic top-ranked websites from medium size to large size organisations. These organisations are majorly worldwide companies and UK-based companies(based in England and Scotland). Considerations were given to all the standard auditing metrics (Performance Score, Accessibility score, Best practices score, SEO score) and execution time (measured in minutes) throughout the experiments.

The metrics are used to capture the analysis of the tool based on the report of the URL inputs and, was expressed as numerical values. An average score for each of the metric was computed for multiple URLs consisting of 15, 30 and 45 URLs respectively for each experimental run. This done to eliminate

TABLE II
TEST RESULTS FROM ORIGINAL ARCHITECTURE

URL Inputs	No of URLs	Time Taken (Mins)	Avg. Performance score	Avg. Accessibility score	Avg. Best Practice Score	Avg. SEO Score	No of Download Files
1 URL TEST	1	0.3	N/A	N/A	N/A	N/A	1
	1	0.34	N/A	N/A	N/A	N/A	1
	1	0.29	N/A	N/A	N/A	N/A	1
5 URL TEST	5	3.32	N/A	N/A	N/A	N/A	5
	5	3.33	N/A	N/A	N/A	N/A	5
	5	3.45	N/A	N/A	N/A	N/A	5
15 URL Test Worldwide Companies	15	10.54	33.33	87.46	84.53	86.53	15
	15	10.40	32.93	87.33	81.67	87.13	15
	15	10.41	34.99	87.43	82.22	87.9	15
15 URL Test UK Based Companies	15	10.01	28.8	86	84.46	87.93	15
	15	09.50	28.6	86	85.53	87.93	15
	15	11.03	29	86	85.21	87.93	15
15 URL Test Scotland-Based Companies	15	09.30	31.8	76.66	82.26	82.13	15
	15	09.12	26.7	76.36	82.26	82.12	15
	15	10.04	28.36	76.36	82.26	82.12	15
30 URL Test	30	18.32	N/A	N/A	N/A	N/A	5
	30	17.40	N/A	N/A	N/A	N/A	5
	30	18.02	N/A	N/A	N/A	N/A	5
45 URL Test	45	26.21	N/A	N/A	N/A	N/A	0
	45	26.36	N/A	N/A	N/A	N/A	0
	45	26.40	N/A	N/A	N/A	N/A	0

any unnecessary bias with the result being generated. The test data was randomly selected and contains URLs of top-ranked websites that were separately experimented with both the enhanced and traditional Google Lighthouse architecture. This is to get an understanding of the effectiveness of the suggested modifications. Due to the fact that original Lighthouse is designed to support auditing of single URL at a time, the results from the processing multiple inputs were manually computed based on the specified metrics. This was achieved by determining the combined aggregate score for each instance of the multiple URLs.

V. RESULTS AND DISCUSSION

The distribution of the data from performance audits were carefully analysed based on the information in Table I and Table II. Both tables shows the average time, Lighthouse audit metrics(performance score, accessibility score, Best practice, SEO score) and the number of report downloaded (including total no of fully and partially downloaded files) during the experiments. The results were compared using data from the enhanced Google Lighthouse with the ordinary lighthouse. The 3FULL in Table I represents the audit report including the results that were successfully produced in the three output file types (json, CSV, XML). It shows that significant number of URLs are downloaded in the three output file formats with multiple file inputs ranging fro 15 to 45 input URLs. This is not the case with the ordinary Google Lighthouse application, as it shows there is weak support for multiple file input and report output formats.

The execution time required to run each set of audit report is also captured in both tables. The execution time for generating audit report based on plain text and XML URL file input are denoted by legends APP Test1 (blue) and APP Test2 (grey) respectively as shown in Figure 3. The figure compare the performance in terms of execution time when running report with the enhanced version and traditional Google lighthouse. The figure shows the enhancement can produce the report at reasonable amount of time when compared to the ordinary Lighthouse with a single URL. There was a slight improvement in speed when using the enhanced lighthouse with average execution time of 38seconds at multiple URL inputs between 15 and 45 URLs. This means there was reduction in the time takes to produce a audit report with the applied that is applied to Google Lighthouse. Obviously, the process of multiple file input is more successful at higher number of URLs, which can makes the process of loading a file from the user's directory to be more time consuming. This however shows that the enhancement can outperform the ordinary Google Lighthouse in terms of speed.

Further analysis involving the accuracy of the report produced with and without the enhancement on google Lighthouse. This was achieved by using the internal Google Lighthouse metrics were completed using the various URL inputs. The URLs belongs to categories of company size that are randomly selected from World-wide companies, UK based companies and Scotland-based companies. Figure 4 shows the average scores of each of the metric based on the 15 URL inputs. It is noticed that there are some higher degree of accuracy with the enhancement when compared to the ordinary Google Lighthouse. The accuracy can also be seen to be consistent across all the internal metrics for Google Lighthouse. The accessibility metric produced a higher score with the enhancement even when using an alternative file input format. However, there has been a consistent drop in the performance score across both versions of system with other metrics maintaining better accuracy.

Finally, considering the overall performance with multiple input and input types it is obvious Google Lighthouse can be made more accessible with insignificant computing resources. This enhancement also allow new functionality that can increase Lighthouse capabilities and availability to a wider audience, thereby making it even more attractive and accessible to companies with massive websites to avoid additional cost (in terms of money and manpower tracking) in providing improvement to quality of the web auditing.

VI. Conclusion

This project aims to make possible enhancements to Google's Lighthouse through adding functionality as alternatives to improve the accessibility of the application. This is successfully achieved with inclusion of Application Programming Interface to support the Lighthouse architecture for processing multiple input URLs and generating multiple file output format when running the performance audit reports. Specifically, three new input and output methods have been introduced, which also allow page comparison including computation of average scores for all input URLs being audited to be possible with insignificance drop in performance.

The results show that enhancement to the reference architecture for Google Lighthouse is quite effective for improved accessibility. The multiple inputs and outputs have been considered to have increased accessibility. Previously Google offered one report download option, but this project offers that same option in addition to two alternatives, this increases the options for users therefore increasing accessibility. Offering multiple download options per report allows data to be manipulated to suite the user's need. This work establishes the need for further improvement to maximise the potential of Google Lighthouse in attempt to improve the user experience and possibly reduces the web traffic when visiting the company's website.

Furthermore, the paper provides the need for improvement to Google Lighthouse from the business and user accessibility perspectives. There is other requirement for Google Lighthouse to allow projects to be connected rather than using the third-party libraries. This will greatly improve the accuracy and time during performance auditing.

References

[1] I. Arapakis, B. Xiao, and B. Barla Cambazoglu. "Impact of response latency on user behavior in web search," In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, 2014, pp. 103-112.

[2] S.K. Shivakumar, "Web Performance Monitoring and Infrastructure Planning" In Modern Web Performance Optimization, Apress, Berkeley, CA, 2020, pp. 175–212.

[3] T. Heričko, S. Boštjan, and B. Saša, "Towards Representative Web Performance Measurements with Google Lighthouse." In Proceedings of the 2021 7th Student Computer Science Research Conference, p. 39. 2021.

[4] K. Yamada and H. Yamana, "Effectiveness of Usability & Performance Features for Web Credibility Evaluation," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 6257–6259.

[5] Google Lighthouse. (n.d). Retrieved January 10, 2023, from https://developers.google.com/web/tools/lighthouse.

[6] K. Chan-Jong-Chu, T. Islam, M. M. Exposito, S. Sheombar, C. Valladares, O. Philippot, E. M. Grua and I. Malavolta. "Investigating the Correlation between Performance Scores and Energy Consumption of Mobile Web Apps," In Proceedings of the Evaluation and Assessment in Software Engineering, 2020, 190–199.

[7] V. J. S. Meder, V. Antonov, and J. Chang. "Driving user growth with performance improvements" Pinterest Engineering, 2017.

[8] M. Pikkanen, " React and Vue performance comparison" Unpublished Thesis, 2021.