Detection of Energy Consumption and Performance of Third-party Applications using Machine Learning and Artificial Intelligence

22_23-J 87

Project Proposal Report

Jayasingha H.M.C.P.

B.Sc. (Hons) in Information Technology Specializing in

Software Engineering

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

June 2022

Declaration

We declare that this is our own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Jayasinghe H.M.C.P.	IT19048338	chathura

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: Appendix 1 Date: 13/10/2022

Abstract

Nowadays, most of the industries have moved to information Systems. Those Information Systems also not standalone applications that are web-based then web-based applications are growing. When the web application grows the IT industry is growing and because of that so many web application developers are coming into the field. They all are humans, they make mistakes and also everyone does not follow best practices and best methods to develop the applications. Any developer can write a working code but not a better code. With this huge amount of traffic in web applications nowadays the applications need better code not only the working code because this huge traffic breaks the not better code that traffic only can handle quality code also these days with this huge amount of traffic web applications getting slow down then for that part also we need to better quality code. This Non-Specific Technology Risk Calculator (NSTRC) is a tool that enables a developer to assess the code quality level, performance level, and site speed level as a percentage and also developers can get suggestions for how to improve the site speed by using machine learning-trained models with the newly developed framework. This framework helps developers to write better quality standard code by themself with help of this framework.

Keywords: Detection performance level , Machine learning

Table of contents

Declaration i

Abstract ii

Table of contents iii

List of figures iv

List of tables iv

List of abbreviations iv

1. Introduction 1

1.1 Background & Literature survey 2

1.2 Research Gap 3

1.3 Research Problem 5

2. Objectives 6

2.1 Main Objectives 6

2.2 Specific Objectives 6

3. Methodology 7

3.1 System Diagram 8

3.2 Connectivity 8

3.3 Technologies to be used 9

3.4 Project Requirements 9

3.5 Work Breakdown Structure 10

3.6 Gantt chart 10

4. Commercialization 11

5. Reference list 12

6. Appendices 13

List of figures

Figure 1.3. 1: NSTRC Logo

Figure 3. 1: Layer pattern in the system

Figure 3. 2: how package manager library contain

Figure 3. 3: Example library inside

Figure 3. 4: Calculation methods

Figure 3. 1.4: High level system diagram

Figure 3.5. 1: Work Breakdown Structure

Figure 3.6. 1: Gantt chart

Figure 7.1: NSTRC Logo

List of tables

Table 1.2. 1: Research Comparison

List of abbreviations

NSTRC – Non-Specific Technology Risk Calculator

1. Introduction

After the software development, developers need to test the application. Software testing has many of the testing to do, including unit testing integration testing, performance testing and code quality testing. If performance and code quality is not tested properly, applications will crash in the future. Therefore, carrying out the tests are most important.

Our application is a framework targeted to developers and QA Engineers. In this application we guide all developers (interns, juniors, seniors) to test their code performance and also the code quality of the developed part and also integrated performance testing.

In our application, we will get code line count and variable count and also a number of ‘for’ loops and image count, and proceed to the output every parameter that has some specific percentage for the calculate the performance level. In here, we will mainly focus on two sides of testing:

1. Functional / Component Performance Testing and Code Quality Testing

In this testing part we can test the function or some component part of the code. This is like unit testing but here we are testing not the functionality we are testing performance level of the code.

2. Integrated Performance Testing and Code Quality Testing

In this testing part we are testing the integrated component testing like this we can do a whole system performance testing as well.

1.1 Background & Literature survey

Nowadays, the software development field is becoming a very popular field with this COVID pandemic situation. After the COVID pandemic situation, most companies moved to the work from home mode. Then employees are not in one place, they hire developers all around the world. So then there are very experienced developers and also interns, training people and also freshers. Then they cannot all maintain the same code quality and code performance level in the code.

After the COVID pandemic, most people are doing their work from home, buying stuff and managing all their work online. The software should have a proper performance level, otherwise when the large amount of traffic comes to the application, it will crash. So, then the company lost its customer base.

In the normal software development process, they do the code review part, but now with this busy situation, senior people and supervisors have a lot of work because of a lot of projects coming up. Then if we miss the code review part, some not-quality code goes to a live environment. That is a big mistake that happens now in software companies.

Let's consider some code parts. The part is functionality wise working fine on any bugs from QA side also, but code performance level is low. Those issues are not coming now when the application growth traffic is increasing, so then these performance issues and code quality issues come. If we fix those issues at that stage, it will be very expensive work because the developer is usually not in the company when the performance issues arise.

So, if we can test these performance levels and code quality issues during the development stage, we can confidently build our production environment with no performance or code quality issues. And, we can stop wasting money on resolving these issues in the future.

1.2 Research Gap

There are some research and mobile applications developed for test developed application performance for the developers and QA engineers. One of the applications is LightHouse [1] this is a browser extension works Firefox and chrome both. From this can test performance by page only and cannot test the whole application, but gives very accurate results for the particular page. But here, the research gap that I have identified is this can't be use for part of the code (function wise) and this can be used as an extension only. Power Mapper also is a similar tool that we can test accessibility mainly. Here, we can test whole application but in hear research gap that I identified is this can't use offline so if we want to test application, we must have to deploy it to the server.

WebLoad is another tool for performance testing on a software application this application is a good application in hear we can do advance testing also but in here research gap I identified is this is language dependent.

Table 1.2. 1: Research Comparison

Research/ Product	Using whole application	Using development environment (ex:- localhost)	Using code parts (features)	Using pages	Identify syntax errors	Using offline	Use as a separate app from the browser
LightHouse	x	x	x	x	✔	✔	x
PowerMapper	x	x	x	x	✔	✔	x
WebLoad	x	x	✔	✔	x	x	x

1.3 Research Problem

Nowadays, software is an important thing in human lives. If we consider an office, it must have a software system to do that and also, in day-to-day human activities, people use software everywhere. Then this is all software, not stand-alone software. Web-based software now accounts for 90% of all software. So the software is on a server and then the server has to handle the traffic in these situations. Most people are using these web-based applications. Then the traffic must be high.

But nowadays, with this COVID pandemic, a lot of people are starting to do their work on online platforms, so we must optimise our applications, otherwise the cost of the software is very high.

We can manage this traffic by cloud computing also, but that is a very costly solution when the traffic increases from the cloud server because it increases the RAM and CPU to handle and manage that traffic without crashing (Auto Scaling). Normally, companies do not go to the solution because if we go to the solution, the company will lose.

With this traffic and all the website performance levels also going down, loading time going down, accessibility levels going down, we must consider these facts when we develop software. Some critical points are listed below.

- Need to reduce image size without losing the quality of the image
- Need to reduce unused JavaScript
- Reduce the size of the JS and CSS files
- Save server storage

Therefore, there is the need to develop applications to identify these problems and suggest solutions for this problem in our application (NSTRC FRAMEWORK) can do this all features.

2. Objectives

2.1 Main Objectives

Main Objective of this Project is to implement a solution for identifying code quality issues and performance issues and suggest proper solutions for improving the code quality and to avoid performance issues.

In this project the main consideration area is a developer side more than QA side, before handover the project to QA for the developer side code level he can identify more performance and code issues developer self.

When the developer completes their development task they can test code issues and performance Issues under NSTRC Framework and get a proper solution at that time.so after the fix those issues also can get the.

When given site code as an input and set parameter to site speed test then as an output give suggestions to improve the site speed.

2.2 Specific Objectives

Identified sub objects to achieve the main object of this project are given below.

- Collect data set and create data set
- Model training to identify the code issues and performance issues
- Develop a framework and package manager
- Develop GUI to test applications in various situations.

3. Methodology

First of all before implementing a tool for the testing we need to develop a framework and package manager. The framework is a base part of this project. Framework handling all the activities of the package manager. Package manager is the second layer of this application and package manager has packages that user can install on the framework and above to the package manager have the testing tool, testing tool can do all the testing with support of the package manager. This is the high-level view of this application.

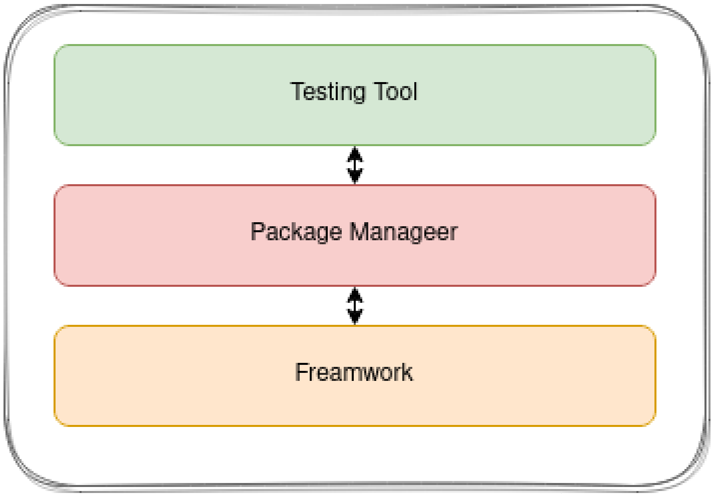
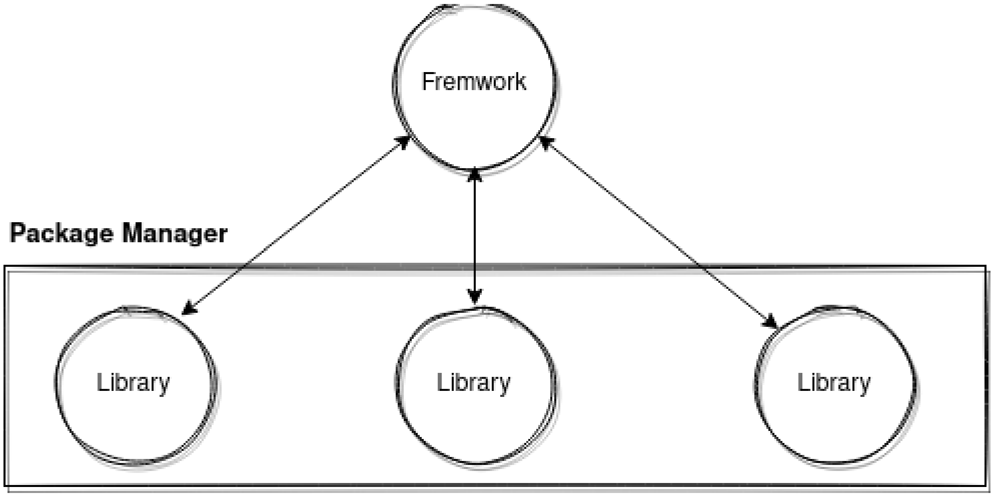


Figure 3. 1: Layer pattern in the system

The first part of this project is the framework and this is a client side framework. Framework is a collection of libraries. In here why I use the framework is we can use a library for this also but if we use the library then we have to contain all of the details in this library. But when we use a framework, we can use separate libraries for each programming language.



PHP package JavaScript package Java package

Figure 3. 2: how package manager library contains

In this project as a package we want the syntax, best practices and issue types in the programming language then from those packages we can compare the written code issues and suggest solutions for that.

What contains in a package (Library):

- Specific language syntax errors
- Language standards
- Language best practices

Package Hierarchy

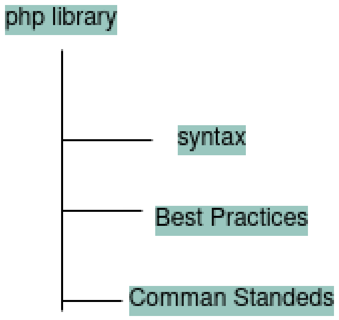


Figure 3. 3: Example library inside

In the Testing tool developer open the testing tool and in there they can upload their code to the testing tool then can test the application by following ways:

1.Peformance Level Testing

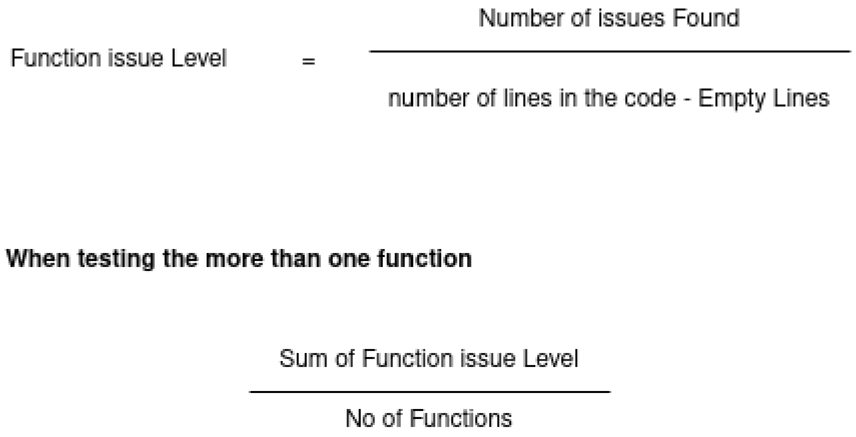
2.Site Speed Testing

3.Code Quality Testing

When the code testing happens, the testing tool interacts with the related library for the programming language that the developer is using; this language needs to be configured when the framework is installed. The system also needs to download packages related to the programming language.

In the comparison part of the library data and the developer code line data done by trained machine learning model and gives the output as a percentage (performance level percentage/speed percentage/code quality percentage)

When giving output, get the below formula.



$$\text{Function issue Level} = \frac{\text{Number of issues Found}}{\text{number of lines in the code - Empty Lines}}$$

When testing the more than one function

$$\frac{\text{Sum of Function issue Level}}{\text{No of Functions}}$$

Figure 3. 4: Calculation methods

3.1 System Diagram

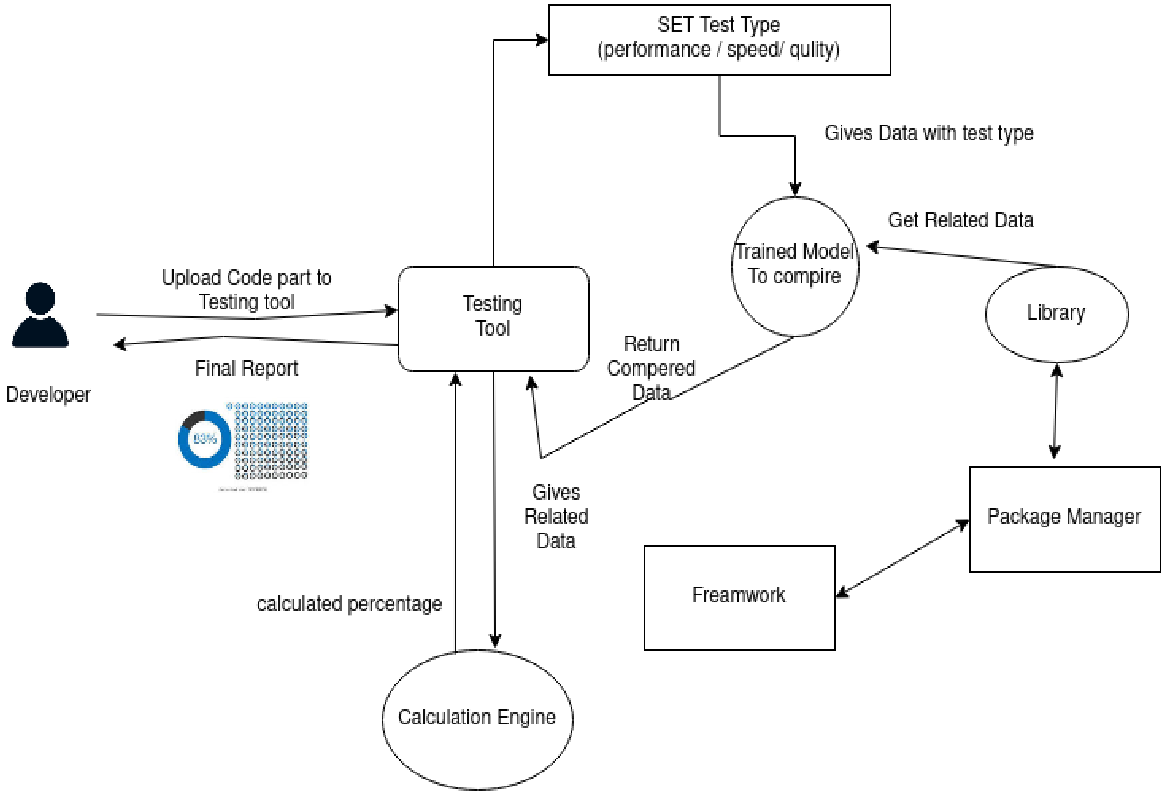


Figure 3. 1.4: High level system diagram

3.2 Connectivity

In this proposed system First developers download and install the NSTRC Framework. Then the Developer needs to install related packages to that which is his development programming language. After Developer needs to configure the testing tool environment

Testing tool default coming under the Framework. When a developer wants the testing tool then the Developer can enable the testing tool and configure the environment. After that the developer develops a function then the developer needs to upload the developed function to the testing tool and set what is the testing type that developer needs to do, Then Developer can get quality level as a percentage on the output screen.

3.3 Technologies to be used

In this project we are going to develop a framework , package manager and also a testing tool for web application testing. Then for the first part we are going to develop a framework and package manager to develop those two parts of this project using Javascript. For the initializing framework and for the package manager going to use javascript based technologies.

Then in the second part we want to develop a measurement level detector module on the framework. For that part we want to train a machine learning model for that we need Python and PyCharm and also some calculation parts also in there we need a javascript.

As the last step we need to develop a Graphical User Interface for the testing tool for that we are going to use HTML , CSS and javascript and VUE js as a frontend framework.

3.4 Project Requirements

- Functional Requirements

  - Installable framework from the web.
  - Installable package Manager from the web.
  - Testing Tool.
  - Can test code quality level.
  - Can test performance level.
  - Can test site speed.
  - Give suggestions to improve site speed.

- User Requirements

  - Developers need to check their code quality level.
  - Developers need to check their code performance level.
  - Developers need to test site speed and site load time.
  - Developers need to get suggestions to improve the site speed.

- Non-functional Requirements

  - Compatible with any of the operating systems.
  - Analyzing speed should be high
  - Output result data should be reliable
  - Understandable GUI for testing tool

3.5 Work Breakdown Structure

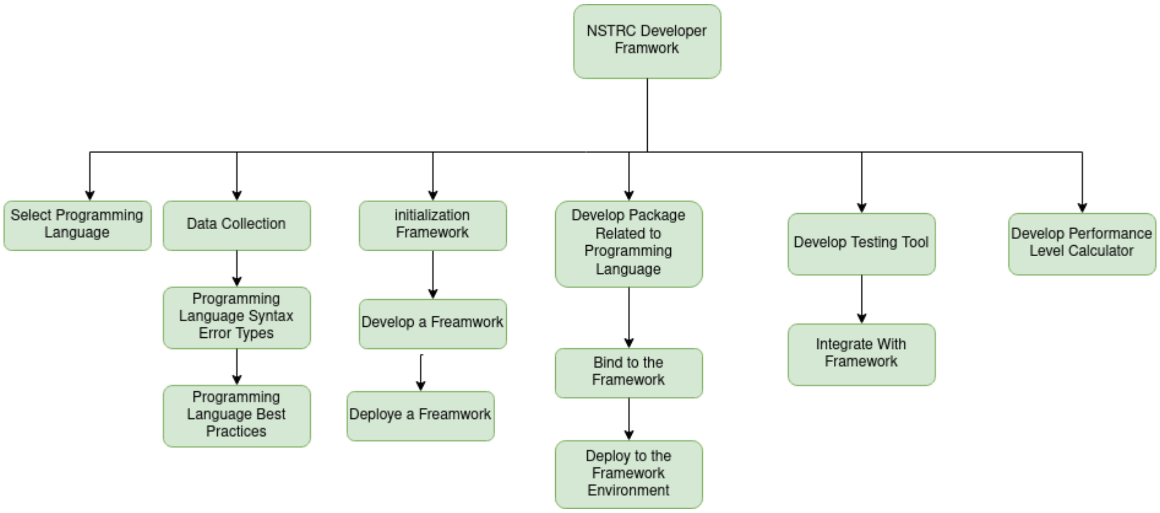
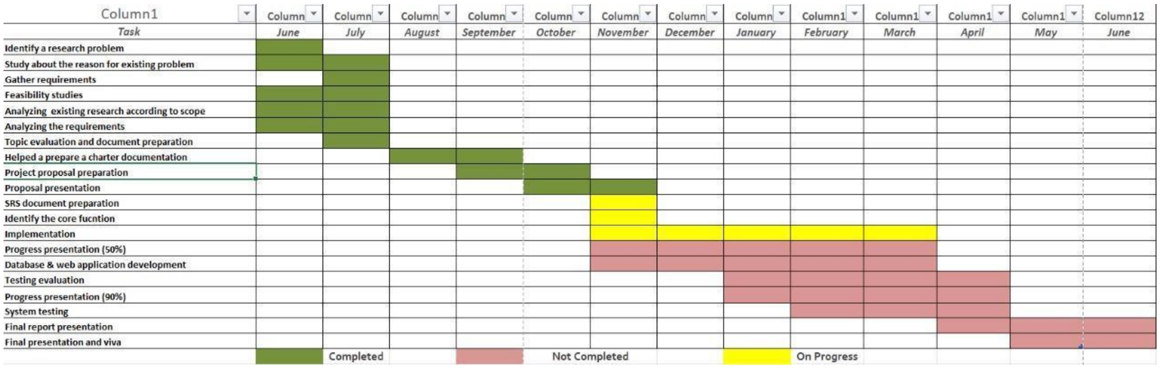

NSTRC Developer Framwork
Select Programming Language
Data Collection
Programming Language Syntax Error Types
Programming Language Best Practices
initialization Framework
Develop a Freamwork
Deploye a Freamwork
Develop Package Related to Programming Language
Bind to the Framework
Deploy to the Framework Environment
Develop Testing Tool
Integrate With Framework
Develop Performance Level Calculator

Figure 3.5. 1: Work Breakdown Structure

3.6 Gantt chart



Column1	Column	Column	Column	Column	Column	Column	Column	Column	Column1	Column1	Column1	Column1	Column12
Task	*June*	*July*	*August*	*September*	*October*	*November*	*December*	*January*	*February*	*March*	*April*	*May*	*June*
Identify a research problem	🟩												
Study about the reason for existing problem	🟩	🟩											
Gather requirements		🟩											
Feasibility studies	🟩	🟩											
Analyzing existing research according to scope	🟩	🟩											
Analyzing the requirements	🟩	🟩											
Topic evaluation and document preparation		🟩											
Helped a prepare a charter documentation			🟩	🟩									
Project proposal preparation				🟩	🟩								
Proposal presentation					🟩	🟩							
SRS document preparation						🟨							
Identify the core fucntion						🟨							
Implementation						🟨	🟨	🟨	🟨	🟨			
Progress presentation (50%)						🟥	🟥	🟥	🟥	🟥			
Database & web application development						🟥	🟥	🟥	🟥	🟥			
Testing evaluation								🟥	🟥	🟥	🟥		
Progress presentation (90%)								🟥	🟥	🟥	🟥		
System testing									🟥	🟥	🟥		
Final report presentation											🟥	🟥	🟥
Final presentation and viva												🟥	🟥
	🟩	Completed		🟥	Not Completed			🟨	On Progress				

Figure 3.6. 1: Gantt chart

4. Commercialization

The NSTRC application is going to use different types of users: Mobile application users, desktop application users, and web application developers. The application is to prevent harm to the user device or the web applications deployed on servers. The application will be released version-wise in the first step of the framework part and will be released for the most popular programming language according to proven statistical information.

In the current times, many user devices are struggling because of installed and invalid compatible third-party apps. Therefore, the NSTRC application will be able to help many users to protect their devices. Currently, there are many intern and trainee developers initiating their career into the technology field. As a result, they will need to be supported to write code in the proper way for proper functioning. Our tool will be able to provide them with the necessary convenience and application in order to carry out the functionalities. When the first release is available, we will make the entire application available for free open source. As time moves forward, new releases with versions will be implemented with better features and trusted resources.

As a desktop mobile application, we need to get closer to the user in order to reach our target. GIGABIT is a short and memorable name, and we designed a logo that is also simple and recognizable. The version is also embedded into the logo to make users aware of the version they are using in, and update it whenever a new release has been made.

Figure 7.1 NSTRC Logo

5. Reference list

[1] Heričko, T., Šumak, B. and Brdnik, S. (2021). Towards Representative Web Performance Measurements with Google Lighthouse. *Proceedings of the 2021 7th Student Computer Science Research Conference (StuCoSReC)*. [online] doi:10.18690/978-961-286-516-0.9.
[2] Himeur, Y., Ghanem, K., Alsalemi, A., Bensaali, F. and Amira, A. (2021). Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. Applied Energy, [online] 287, p.116601. doi:10.1016/j.apenergy.2021.116601.

[3] Eid, S., Makady, S. and Ismail, M. (2020). Detecting software performance problems using source code analysis techniques. Egyptian Informatics Journal, 21(4), pp.219–229. doi:10.1016/j.eij.2020.02.002.

[4] Hrci Marketing. (n.d.). Top 5 Common Performance Problems. [online] Available at: https://www.hrci.org/community/blogs-and-announcements/hr-leads-business-blog/hr-leads-business/2022/02/14/top-5-common-performance-problems [Accessed 15 Jul. 2022].

[5] Heričko, T., Šumak, B. and Brdnik, S. (2021). Towards Representative Web Performance Measurements with Google Lighthouse. Proceedings of the 2021 7th Student Computer Science Research Conference (StuCoSReC). [online] doi:10.18690/978-961-286-516-0.9.

[6] https://www.cmcrossroads.com/sites/default/files/article/file/2012/XDD3579filelistfilename1_0.pdf

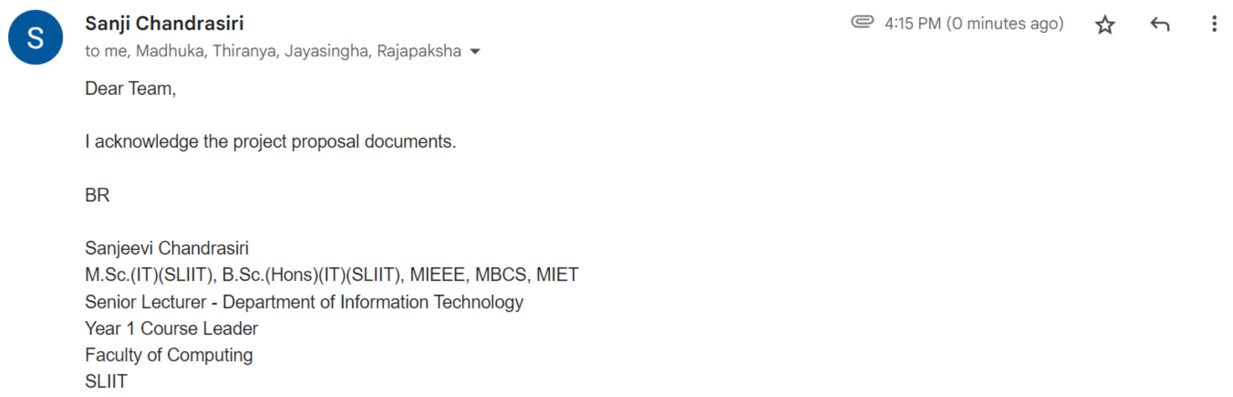
[7] Software Testing. (n.d.). [online] Available at: http://www.sci.brooklyn.cuny.edu/~sklar/teaching/s08/cis20.2/papers/software-testing.pdf.

[8] https://openarchive.nure.ua/bitstream/document/17610/1/Boston_1.pdf

[10] Di Lucca, G.A. and Fasolino, A.R. (2006). Testing Web-based applications: The state of the art and future trends. Information and Software Technology, 48(12), pp.1172–1186. doi:10.1016/j.infsof.2006.06.006.

6. Appendices

Appendix 1 - Acknowledgement of the supervisor - Ms. Sa



Sanji Chandrasiri
to me, Madhuka, Thiranya, Jayasingha, Rajapaksha

4:15 PM (0 minutes ago)

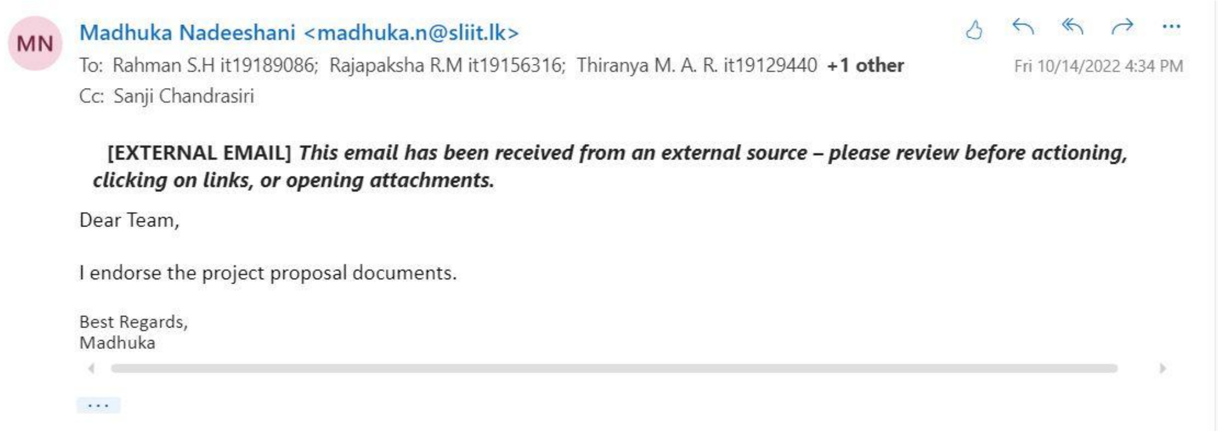
Dear Team,

I acknowledge the project proposal documents.

BR

Sanjeevi Chandrasiri
M.Sc.(IT)(SLIIT), B.Sc.(Hons)(IT)(SLIIT), MIEEE, MBCS, MIET
Senior Lecturer - Department of Information Technology
Year 1 Course Leader
Faculty of Computing
SLIIT

Acknowledgement from Co Supervisor, Ms. Madhuka Nadeeshani



Madhuka Nadeeshani <madhuka.n@sliit.lk>
To: Rahman S.H it19189086; Rajapaksha R.M it19156316; Thiranya M. A. R. it19129440 **+1 other**
Cc: Sanji Chandrasiri

Fri 10/14/2022 4:34 PM

[EXTERNAL EMAIL] ***This email has been received from an external source – please review before actioning, clicking on links, or opening attachments.***

Dear Team,

I endorse the project proposal documents.

Best Regards,
Madhuka