

**Hor Weng Liang WES140009**

**Muhammad Hazim bin Kamaruzzaman WES140022**

**Luqman Hakim bin Ahmad Tamrin WES140014**

**Siti Sarah binti Azmi WES140035**

**Siti Aishah binti Aminuddin WES140033**

**Adreen binti Mohd Affendy WES140001**

**Nur Aainaa binti Ismail WES140026**

Prepared for:

ASSOCIATE PROF. DR. ZARINAH BINTI MOHD KASIRUN

GROUP 1 (basic COCOMO model)

WKES3108 SOFTWARE EVOLUTION & CONFIGURATION

TEAM MEMBERS

Table of Contents

1.0 [**INTRODUCTION** 2](#_Toc501489371)

1.1 [Github 2](#_Toc501489372)

1.2 [Cocomo 2](#_Toc501489373)

2.0 [**TOOL ASSESSMENT** 0](#_Toc501489374)

3.0 [**CHANGED PROPOSED** 0](#_Toc501489375)

4.0 [**GITHUB RECORD** 0](#_Toc501489377)

# Table of Figures

Figure 1 : No "back" button in browser………………………………………………………………………………………………….

Figure 2 : Default value without any input-KLOC value is empty……………………………………………………………

Figure 3 : COCOMO front page before maintenance……………………………………………………………………………..

Figure 4 : COCOMO front page after maintenance………………………………………………………………………………..

Figure 5 : COCOMO result page before maintenance……………………………………………………………………………

Figure 6 : COCOMO result page after maintenance………………………………………………………………………………

Figure 7 : COCOMO result page with back button………………………………………………………………………………..

Figure 8 : Mode options prompt window……………………………………………………………………………………………..

Figure 9 : Front end design commits…………………………………………………………………………………………………….

Figure 10 : Contributions of commits…………………………………………………………………………………………………..

Figure 11 : Visitor Frequency………………………………………………………………………………………………………………..

# **INTRODUCTION**

## Github

GitHub is an open-source version control system created by Linus Trovalds. It is a place where developers can create an application or system, make constants changes on the code, release new version more than once and store the modifications in a central repository. GitHub also allows developers to easily collaborate as they can download new version of the software, make changes and upload the latest version of it. This way every developer can see the new changes, download and contribute.

## Cocomo

Constructive Cost Model (COCOMO) is a procedural software cost estimation model developed by Barry W. Boehm which use to estimate size, effort and duration base on the cost of the software. COCOMO are divided into three mode which are

* Organic

“Small” teams with good experience that work less than rigid requirements

* Semi-detached

“Medium” team that are got mixed experience working with a mix rigid and less than rigid requirements

* Embedded

Developed within set of “tight” limitations. It is also a combination of organic and semi-detached between hardware, software and operational calculation.

**Basic COCOMO Formula**

The basic **COCOMO equations** take the form

Effort Applied (E) = ab \*(KLOC)bb [ [man-months](https://en.wikipedia.org/wiki/Man-month) ]

Development Time (D) = cb\*(Effort Applied)db [months]

People required (P) = Effort Applied / Development Time [count]

where :

* KLOC = **Estimated number of delivered lines** (expressed in thousands) of code for project.
* The coefficients *ab*, *bb*, *cb* and *db* are given in the following table (note: the values listed below are from the original analysis, with a modern reanalysis[[4]](https://en.wikipedia.org/wiki/COCOMO#cite_note-4) producing different values) :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Software project | ***ab*** | ***bb*** | ***cb*** | ***db*** |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

# **TOOL ASSESSMENT**

For the tools assessment of Basic COCOMO, we get the main reference from <http://groups.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/kutcher/kutcher.html>.

Several reviews and comments had been made after a thorough assessments is done on this tool. In terms of functionality, user able to choose the mode that they want and they able to key in the value required to calculate COCOMO. If user give characters or letters as the input, there’s prompt window that state user must give float value to calculate. Even if there’s no input given by user, the result page will display default value of the variables “A”, “B”, “C”. This may give user a clue on what value should be key in.

However, there’s no error handler for the mode options. This tools still display the result even when none of the mode is chosen. Fortunately, there’s no error or crashing happen when user click on “Calculate COCOMO” button or “RESET” button. All button functions perfectly.

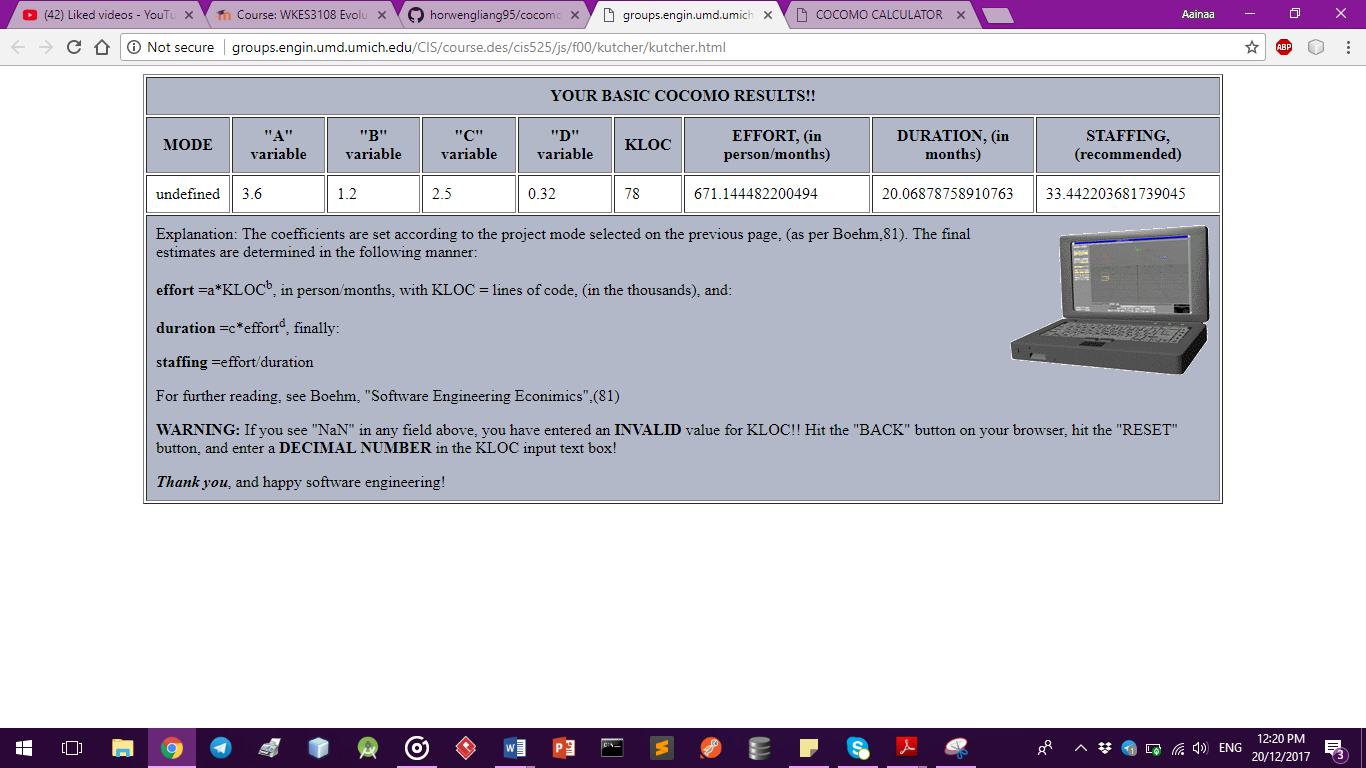


Figure 1 : No "back" button in browser

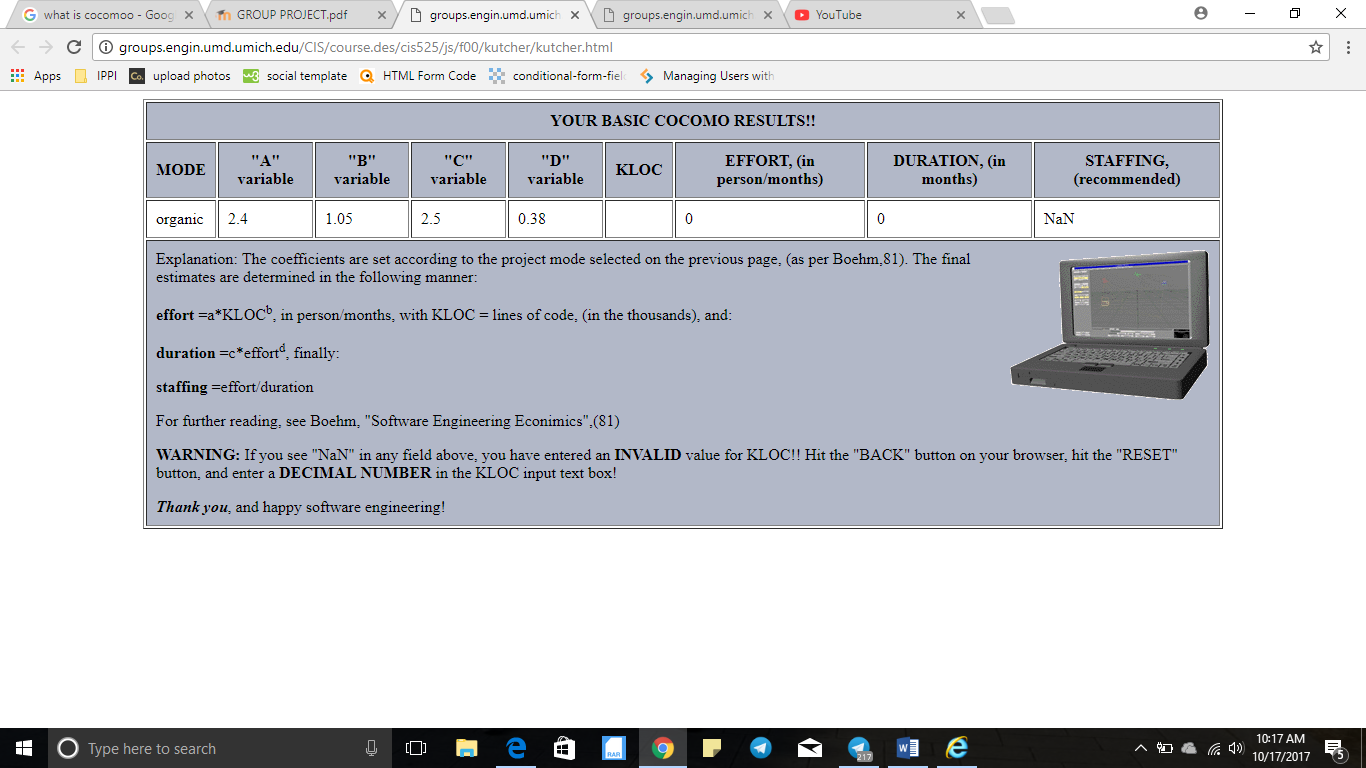


Figure 2: Default value without any input-KLOC value is empty

In terms of usability and user experience, there are major flaws that need to be improved for better user experience. The first thing is, there’s no back button or return or any button that indicate user to return to home page after calculation. First-time user may be flustered or confused because the back button is not visible on the browser also. Next the alignment of the mode options. User may accidently skipped or forget to choose the mode because the mode options is aligned with the “computer” icon and the input text box. In addition to that, with the type and size of font, the radio button looks like pointer rather than a button in the first glance. Because of that, if user, directly key in the value without choosing the type of mode, the result displayed will be inaccurate. Moreover, there’s no statement stating that if there’s no choices made upon the mode options, the default output value will calculated from embedded mode.

The first impression that we get from this tools is it has an outdated and boring design and it fails to attract our attention. The color contrast, alignment and arrangement of the text is particularly boring. The size and type of font makes it too formal and reduce the optimization of user experience. Without much satisfaction from user, user may lose interest to use the system.

# **CHANGED PROPOSED**

There are a few changes and improvement that we proposed to improve the tools that we assessed. This perfective maintenance is necessary to increase usability and user experience from a low to a higher and better level.

1. **UI Design**

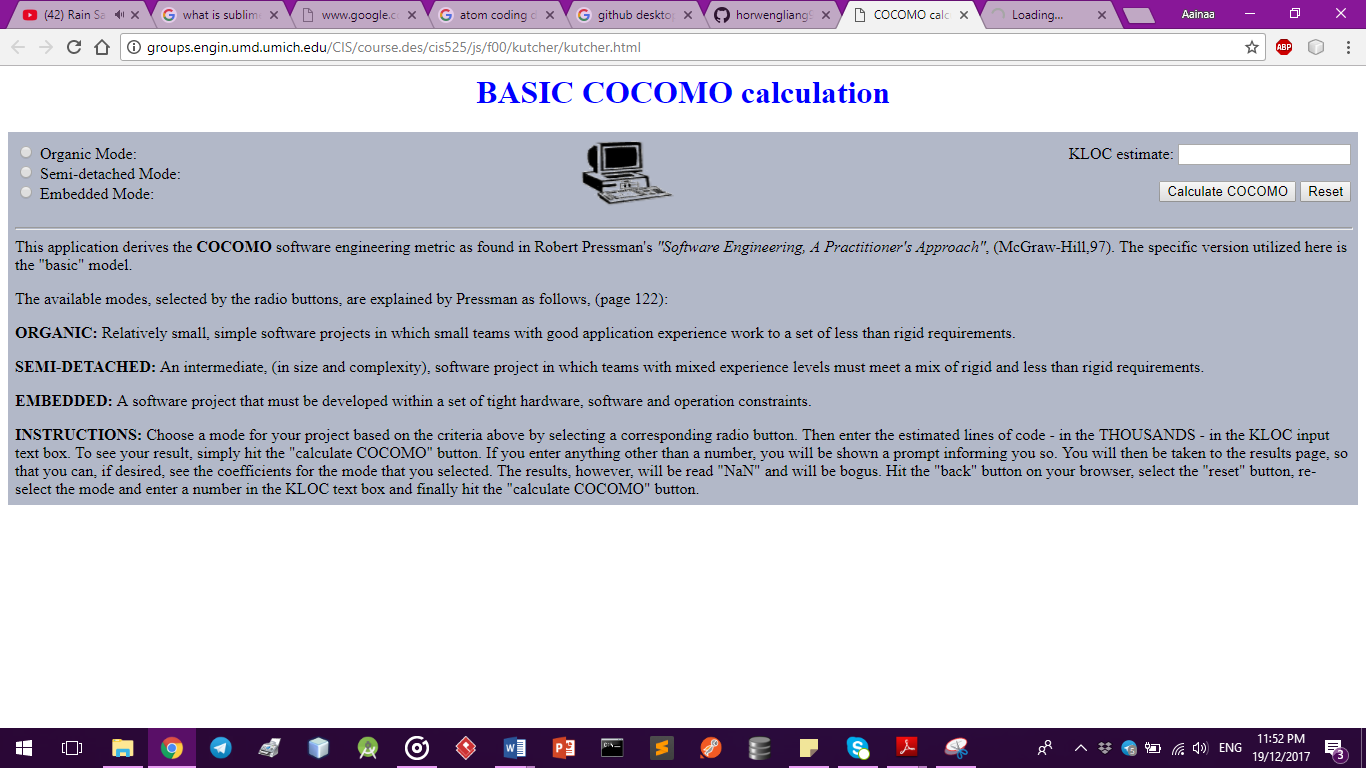
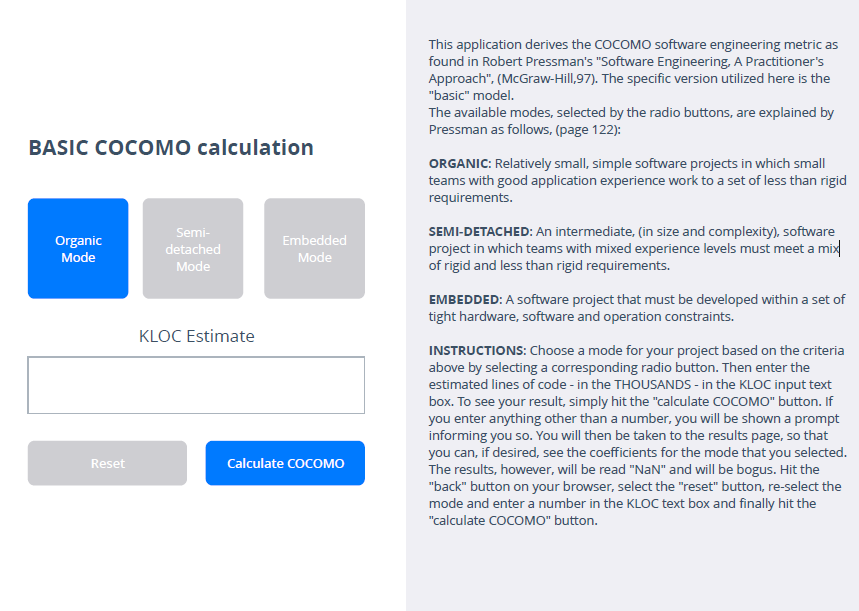
The first thing that is critical is the attractiveness of user interface design. The existing webpage is boring and dull. The color, layout and arrangement of the interface need to be changed to a more cheerful and pleased-to-eyes. The navigation of the webpage also need to be improve to enhance and fully utilize user-friendly’s concept.

Figure 3 : COCOMO front page before maintenance



The changes that we proposed is to change the default background color to white color. We also divide the webpage into 2 columns. The left side is the mode options and the input box for user to key-in input to get the calculated value. While the right-hand side, is where all the instructions and statements that need to be announced to user for clarification. The size and type of font also is changed to one that is more pleased-to-the-eyes type.

Figure 4: COCOMO front page after maintenance

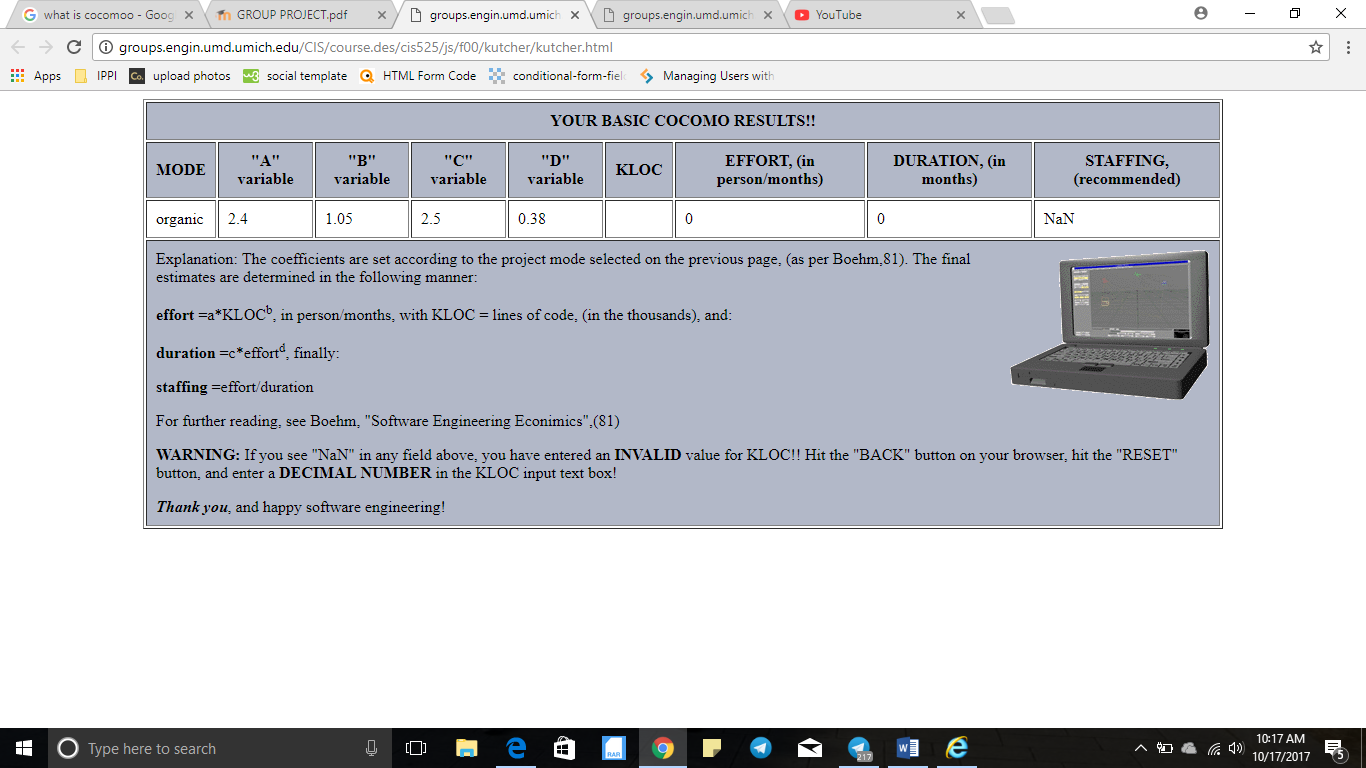
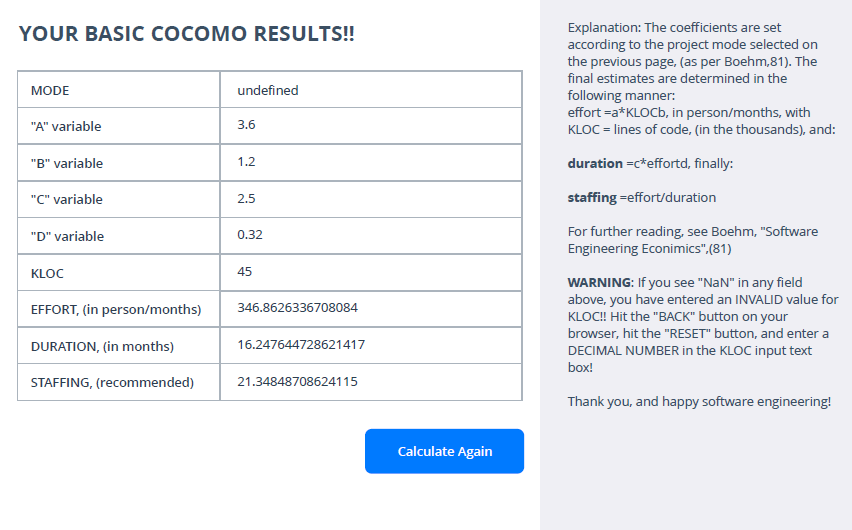


Figure 5 : COCOMO result page before maintenance

Figure 6: COCOMO result page after maintenance

1. **Add back button**

Can back to previous page when click back button on browser

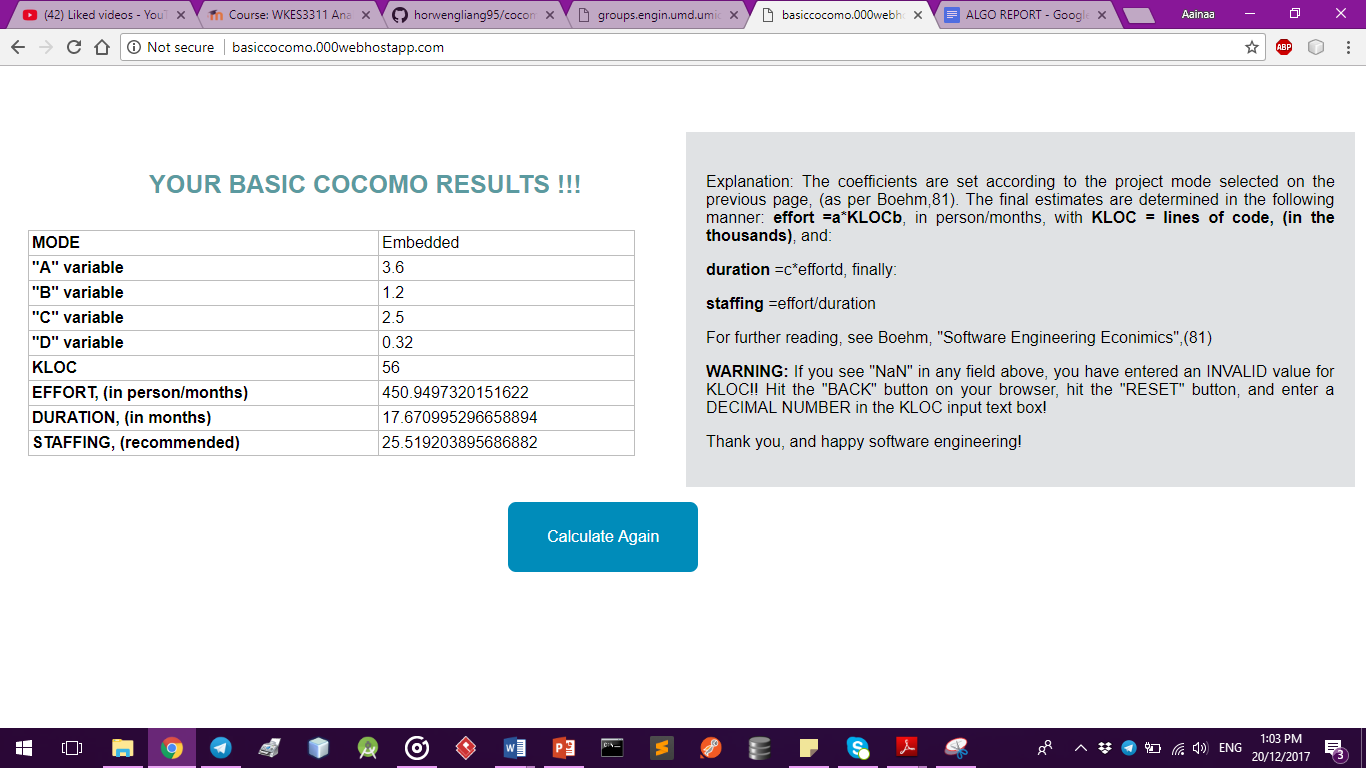


Figure 7 : COCOMO result page with back button

1. Mode options

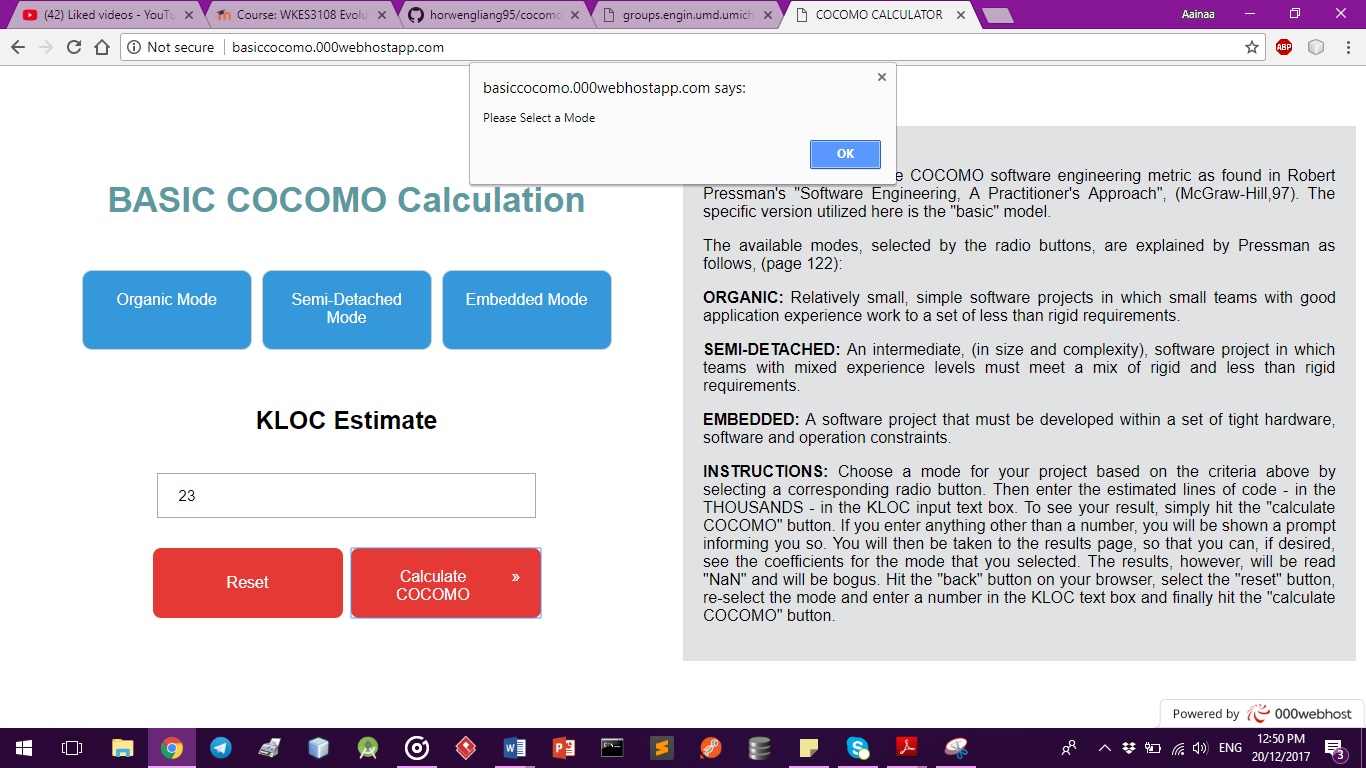
When user doesn’t choose any mode, this prompt window will pop-up to ask user select a mode.

Figure 8: Mode options prompt window

# **GITHUB RECORD**

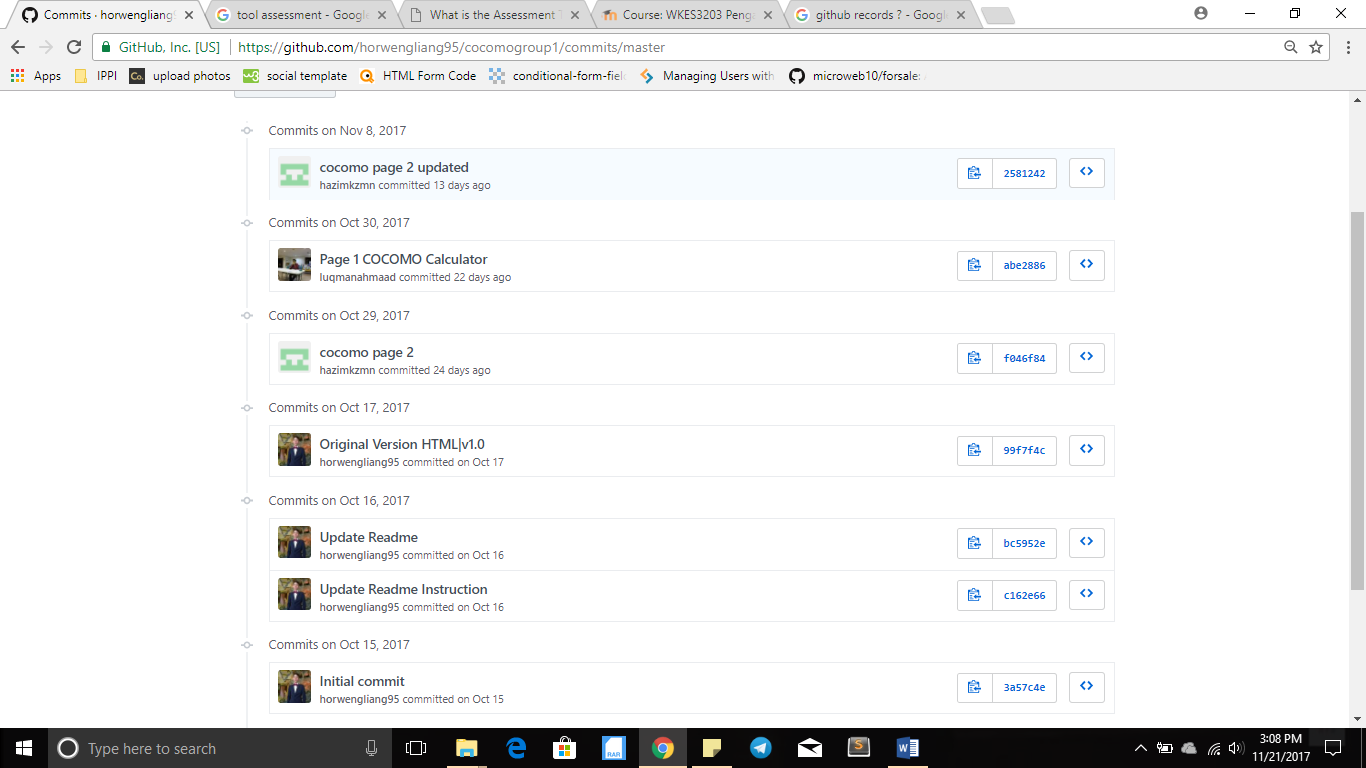
1. Front end design commits that display contributors that commit in repository

Figure 9 : Front end design commits

2) Contributions of commits for the past 2 months

Figure 10 : Contributions of commits

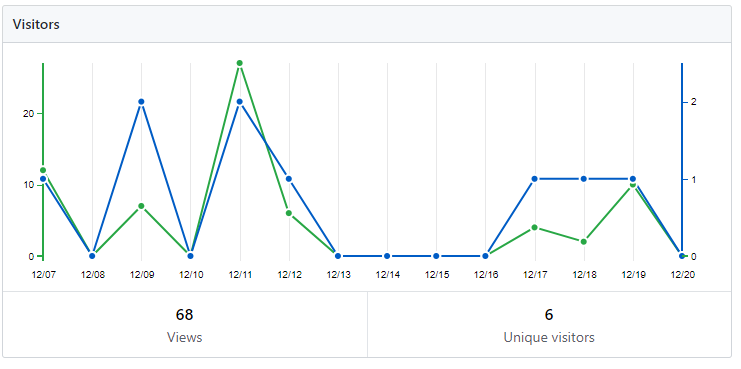


Figure 11 : Visitor Frequency