Software Analytics Report

Quality Assessment for C402

**Done By:** Femi Sowemimo

[Project Overview](#h.50vl43xzynyn)

[Quality](#h.vqsq2tami7ey)

# Project Overview

**Name:** Votetell

**Website:** https://votetell.com/

**Languages:** html, javascript, css

**Github:** <https://github.com/awaseem/votetell.com>

Votetell is a website created by a close friend of mine in Calgary. Its is a simple poll creator that allows users to vote and also send a custom response to the creator. This program was developed with Meteor JS and Semantic UI.

The initial requirement for this project is to use our previous Cmput 401 project and run the Kiuwan Code analysis software on it, but my Cmput 401 project was done in a VM environment, and there wasn’t a way to try and wrap all this into a zip file and run the analysis software. A secondary issue with using my Cmput 401 project was the fact that our code was to make changes to an already functioning moodle environment, this having hundred of thousands of code and the Kiuwan wouldn’t have been able to fully analyze this. Talked to professor prior to this and she recommended using an open source project, and I felt there was no better open source project rather than something created by a friend of mine.

Initial requirements given for the Kiuwan assessments were:

* **Maintainability:** 90
* **Security:** 90
* **Efficiency:** 90
* **Portability:** 90
* **Reliability:** 90

No specific reason for giving it all a performance rating of 90, it was mentioned by Prof. Teresa that this shouldn’t be a big deal but as long as the minimum is a PR (performance rating) of 70. I decided to stick with giving it all a 90, because seeing as the type of project this is I felt the 90s made sense.

Maintainability of 90 works, because seeing as this is a web application the idea of having such a high PR means the web application can be repaired in an ease and fast manner so the system is never down for an extended period of time for any reason.

Security having a PR of 90, also means user inputs are all valid, where the user wouldn’t be able to perform any code injection to cause the web app to run into any problems.

Efficiency having a similar high PR shows how reliable the code is, having a consistent coding style and following the appropriate code ethics. This can be in terms of assigning variables with proper names along with proper type attribute.

Portability talks about how easily the software can move from one platform onto another, for a web app this is very key because we are faced with people using platforms such as Tablets, Phones, Laptops & Monitors along with a variety of web browsers. Making sure all this checks out is why I gave the portability such a high PR.

Reliability focuses on the software having a specified level of performance. Making sure a web application loads all appropriate information in a fast smooth manner is very key. Hence why the 90 PR.

# Quality

Result from the Kiuwan Assessment

**Maintainability:** 4

**Reliability:** 96

**Portability:** 96

**Efficiency:** 100

**Security:** 100

**Global Indicator:** 76

Maintainability scored very low in terms of this web app project, taking a look at a more detailed look at repairs we are faced with the following issues.

|  |  |  |  |
| --- | --- | --- | --- |
| **Defects** | **Files** | **Rule Name** | **Characteristic** |
| 11 | 7 | Avoid statements without semicolon | Maintainability |
| 2 | 2 | Do not update control vars in ‘for’ loop body | Maintainability |
| 1 | 1 | Avoid popup windows | Reliability |
| 13 | 6 | Define Variables with var | Reliability |
| 2 | 2 | Avoid declaring with names already used | Reliability |
| 64 | 27 | Non-portable function check | Portability |
| 48 | 18 | Duplicated code: big block | Maintainability |

Maintainability having such a low score greatly has to do with the problem of “Duplicated code: big block”. This issue occurring 48 times and having shown up in about 18 files and totaling up a time of 192h in being able to fix this issue.

Going into the Kiuwan main website and observing the Maintainability analysis code, the problem with the duplicated code is due to the fact that a semantic-ui is being used.

Semantic-ui is a development framework that creates HTML with ease, along with possible templates that can be used. This making sense for codes being duplicated so much, because if we are using a button element from a template it doesn’t group them into one. It simply duplicates the code since this is how the Semantic-ui works along with most template platforms.