

bronze score calculation

- [redacted]
- [redacted]
- [redacted]
- [redacted]
 - [redacted]
 - [redacted]
- [redacted]
 - [redacted]
 - [redacted]
- [redacted]
 - [redacted]
- [redacted]
 - [redacted]
- [redacted]
- [redacted]

Introduction

[redacted] is an open source project based on Ruby on Rails framework. Expertiza allows the instructor to create new courses, assignments and customize new or existing assignments. It also allows the instructor to create a list of topics the students can sign up for. Students can use the expertiza platform to form teams to work on various assignments/projects. Students can sign up for various OSS topics to work on, peer review other students submission. Students can submit their assignment in the form of URLs, upload files.

Login Instructions

- For Instructor, username: [redacted], password: [redacted]
- For Student, username: [redacted], password: [redacted]
- For Student, username: [redacted], password: [redacted]

Problem Statement

Sometimes when a reviewer fills out a review rubric, one or more fields can be left empty by the reviewer. While calculating the peer-reviewed score, the program is counting the empty fields as zero. However, this is not correct. The empty fields should not be counted at all, as this may affect some other metrics (like average

score) of the reviewee.

Solution

Explanation

While calculating the peer-reviewed score, the program checks whether the score is empty or not. If a score is empty, it should not be counted while calculating average score. Only not null review scores are to be counted while dividing by the total number of reviews, for the average score calculation.

GitHub links

GitHub link

-

Pull Request link

-

Our Work

File Modified

/app/models/ rb

Explanation

The method 'average_score_for_row' is used to calculate the average peer-reviewed score. We added a variable 'no_of_columns' to count the number of review scores that are not null. We changed the way the average score is calculated. Previously, 'row_average_score' was calculated by dividing by '@score_row.length', where the length will include both null and not null values. Now we divide 'row_average_score' by 'no_of_columns', which does not include null scores.

Testing

File Added

/spec/models/.rb

Explanation

We added a RSpec file for the class 'VmQuestionResponseRow' to test the method 'average_score_for_row'. We tested the following scenarios for calculating average score:

- **All scores are not nil**
 - This is the scenario where no fields are left empty. Average score was calculated correctly in this scenario before the issue fix. We test that the score is calculated correctly after the issue fix too.
 - All not nil scores that are all zero are also tested to ensure that the output is zero.
- **All scores are nil**
 - This test ensures that when all fields are empty or are nil, average score is also nil instead of being zero.
- **Mixture of nil and not nil scores**
 - This test ensures that when some fields are left empty, they are not taken into consideration while calculating average score. To compute average score, the total score will be divided only by the total number of fields that are not nil.
 - We tested this scenario by including or excluding zeroes in the scores and varying number of nil scores to verify correctness in average score.

Screencast

-

References

-

-

-

RSpec Documentation

- <http://rspec.info/documentation/>

Retrieved from