Iteration: 10 (27/04 - 08/05)

Alex Taylor - amt22@aber.ac.uk

April 29, 2017

Version 0.3 (Draft)

Contents

Note: This iteration is slightly longer as it includes the half week between where iteration 10 would have ended and the hand in.

1 Story: Admins can then save the results as csv or xml

1.1 Analysis - breakdown of tasks

The most convenient way to save these results would be to convert the data to a csv format, as that can be read by many Spreadsheet programs. After some researching, an extension for Laravel was found that could easily create spreadsheets within a Laravel application(cite http://www.maatwebsite.nl/laravel-excel/docs).

There the list of tasks for this story:

- Add route, button and an action for this
- Install Laravel spreadsheet converter extension
- Pass the formatted answer data into the csv creator

1.2 Design

Due to the way the original answer hndling was designed, results are only saved for a session and not for quizzes. This means when a new quiz is run, the results are wiped. This means a good place for the download action would be the quiz admin panel, the one next and previous buttons. This would be an ideal place as it means lecturers can hit download before they end the quiz.

The action itself does not seem to fit onto any of the other controllers, and a new one should be created to handle this function. A sessionController would be the best fit as this action falls under a session and not quizzes so would not be sensible in a quiz controller.

The csv format would be in the form of a single sheet with many rows. The first row would be the question, the second row the answers and the third the number of times the answer was chosen. These three rows would then repeat for each question, so a three question quiz would contain nine rows.

1.3 Implementation

Setting up the new route, button and controller was a quick task. Artisan (TODO have i mentioend artisan before?) was used to generate the new controller and then all that needed manually doing was adding a route to the web routing file and creating a new button on the admin panel to call this route.

Installing the extension was done with Composer which installed it under the vendor/ folder. It was then added as a Facade(cite facade https://laravel.com/docs/5.4/facades) which allow the quick use and access to the functions it contained. The extension took an array of data and turned that into a spreadsheet of the type specified, in this case a csv. The array is looped over and each item in the array is a row in the sheet. If an item of the array is an arrays itself, then each item within these sub arrays would be a cell within the row. This means the sheet is built from an array of arrays.

Formatting the arrays was the primary task here. First all the questions from the session were selected, then looped over. Within this loop the question was added as the first row. Then all the answers associated with that question needed to be added. Getting them from the database used a simple Elooquent query. Two arrays needed to be created from this data, an array of the answer text and an array of the number of times these options were submitted. The total times these were submitted had not been calculated so this was done first, looping over the answers and counting the times an answer was given by creating a new array of the answers and times clicked in a key=¿value pairing. This array of answer=¿answeredNumberTimes could then be looped over and the each key added to the first array, and the values to the second array for the two rows.

The problem here was that the answer text in the databse is stored as "answer1" rather than what the student actually clicked. This meant that these answers had to be changed, this was accomplished by using the answer value, like "answer1", as the key in the current question variable and adding this value to a new "keys" array. Another problem was the way multi selection questions were stored, such as "answer1, answer2". This meant adding another extra step to break the string into an array and replacing them with the actual answer values and recombining them into a single string.

(TODO add code snippet and image of a csv)

1.4 Testing

TODO

2 Report writing

Report Writing: Restructuring, Iter10, Design, Eval, touching up, User Testing

3 Extra implemenation based on feedback from admin user tester

3.1 Quiz control buttons should be greyed out

3.2 Cancel button on edit pages

3.3 Bugs

Upload times Route logic issue