Float Moodle - Second Round

(Here, we describe how you can earn 25 points)

Recall that you got a score N1 of upto 13 from the first round. You were told min(N1, 10). You earn N2 points from this round of evaluation. min(N1+N2, 30) will count to your course total, and the excess over 30, if any, will be taken into consideration while making grading decisions.

In the first round, we got started with the basic functionality one would expect from an online assignment submission portal. Now, we will think about adding features that make using our tool a more complete experience. Some of these are inspired by the creativity you've shown in the first round. It's really nice that each of you gives the class ideas to build better projects!

We have indicated credit against each feature we mention here; but feel free to implement more creative stuff. We value hard work, and if you do something non trivial that we haven't formally mentioned here (eg. deploy well), it is indeed worth reasonable credit (3-4 points for the example).

Basics from the previous round (3 points)

If you haven't accounted for these features already, now is the time to tie up the loose ends. Several teams have already organised the Assignments into Courses, and, in this round, 1 point is reserved for creating courses.

The roles a User can play in a course are a bit more nuanced. There are, of course, Students and the Teacher. However, on Moodle, you might also have noticed that there are "non-editing teachers" (often TAs are non-editing teachers). A teacher can enroll or deregister students and TAs, add Assignments, make Announcements, etc. A TA, however, must ideally have intermediate privileges.

1 point for introducing intermediate level of privileges (eg. can upload grades but cannot create assignments, cannot enroll students, etc) and a TA role. 1 point more if the Teacher can decide how much editing power the TAs have.

Deadlines and To-do lists (3 points)

While creating an assignment, there should be an option to add a due date and time, and submissions should be disabled past the due time. 1 point for providing this functionality. Then, in the student's view, you must create a to-do list of all assignments that are due across all courses. Similarly, in the Teacher's view, create a to-do list of all assignments that are yet to be graded. 2 points for these to-do lists. It would be nice if you could visualise them on a calendar like Moodle does.

Communication (3 points)

A good community enriches the learning experience. Can you create discussion forums in courses, so that students and the teacher can ask and answer doubts publicly? Can you slightly adapt the Assignment to enable the teachers to make Announcements? 1 point for creating Announcements and/or discussion forums.

In addition, you can also support private messaging between students. 1 point for DMs. However, beware! This feature can be exploited by users to indulge in plagiarism. Your Moodle could be used to administer a test, and it would be nice to give the teacher power to temporarily disable discussion on course forums. 1 point for this.

Aggregates (7 points)

If you haven't done so already, you must allow a Teacher to upload grades and feedback for an assignment in bulk, via a CSV file. 2 points are reserved for that.

5 points depend on your creativity and how you decide to interpret and present the wealth of data you have at your disposal. As a teacher, one should be able to assign weights to every assignment, so that when students get their marks, they also see their course total.

The course total is what you call an aggregate, and that's only scratching the surface of what you can do. Another thing you could do is show the teacher statistics for an assignment that has been graded. By show, you could actually visualise it as a histogram, adding to just reporting the mean and variance. (the overwhelming majority of teams are based in python, matplotlib should be handy. Also, there's a bunch of javascript libraries too, if you're going that route.)

You could also show the teacher how class average (and variance) evolves as the course progresses. Something like a bar graph with error bars would work.

Even students can benefit from some analytics. They could see trends in their performance, possibly aggregated across all, or a selection of courses they're taking. An instructor could choose to warn students who are falling significantly below class average.

Here's a summary of what we suggested, but feel free to give wings to your creativity

- Allow the instructor to assign weights to each assignment (i.e. how much the assignment contributes to course total)
- Show students their course total
- Show teacher statistics of the assignment (mean, variance, histogram)
- Show teacher mean and variance of class average for all assignments, in a single graph
- Show students their progress, and possibly warn them if it is known that they're significantly below class average

Autograding (3 points)

Automation is the future. To earn these 3 points, you should allow the Teacher to submit an autograding script that evaluates an assignment when run from a student's submission directory. Using this script you should be able to generate marks and feedback for a student as soon as a submission is made, or you can choose to go over all the submissions after the deadline to grade them in bulk. You can leave it up to the Teacher whether to display these marks right away or wait for the deadline.

Setting up the framework for compiling and running students' code is a challenge in itself, not to mention all the error checking you should do because of the weird ways of submission some students choose. This one depends on you, you can go ahead and state the limitations of your platform saying that you only support so and so languages, and so and so packages. Or you could get creative and ask the teacher for a docker image or a setup script that installs all the required software. Also, you can't assume the Teacher to be perfect, you can expect errors in the scripts given by the Teacher, and your platform should handle these errors gracefully.

Command Line Interface (3 points)

This one is for all the students who feel they are cool when they do everything on the terminal.

Using your command line tool students should be able to do all the important stuff like submitting an assignment, checking grades or deadline for an assignment, fetching files uploaded for an assignment by the Teacher and fetching grades.

Like you did for your webapp, you will have to identify and authenticate the person making this request before giving away all that they ask for. You could directly use the username password combination or you could use Personal Access Tokens like github does, and store it somewhere on their machine so that they don't have to provide it again and again.

% Course Completed (1 point)

Based on the assignments and course material (if you decide to add it), you should be able to calculate the % of a course any given student has completed. You will get a point if you calculate this and display it on a nice progress bar.

Email Invitation and Password Update (2 points)

To earn these 2 points, you should use emails for course invitations, important updates like assignment submitted, assignment released, announcements and more if you want to.

As an additional layer of security, you should email the user a password update link / OTP if they wish to change their password.

Viva

To be conducted on 29th and 30th of November, you'll be assigned a slot on one of these days. Please plan your time well. A private GitHub repo with a README is mandatory,

make sure you give a brief overview of the technological choices (i.e. which framework, libraries, etc. so people can run code) and the key features you implement.

There aren't marks reserved for this, but note that our decision to award you credit for the things we've described above is based on how you present your work in the demo, and what we see when we take a glance at your README. So, from the grades point of view, it's nice to make a good impression, but more importantly, collaborating and building tools in this manner is the right thing to do.

Please plan your time well. More details about the viva will be conveyed in due course.