Moodle Grading Assistant

Department of Computer Science
Title: Online Grader Helper App
Authors: Emily Costa & Michael Sisko
Date: December 6, 2021
Advisor: Victor T. Norman

Problem

Grading is a time-consuming and redundant task, and some common complaints about Moodle's user interface are that it is clunky, slow, and frustrating. When someone wants to grade an assignment, you can enter the feedback there or pull down a CSV file, modify it, and reupload it. One consistent issue is redundancy, meaning similar feedback often applies to multiple students. Another issue is the difficulty graders have in modifying the table efficiently. If a grader wanted to change something globally, it would require fixing each individual entry tediously. We believe our application is crucial since it minimizes administration time and permits graders to allocate more time elsewhere.

Background

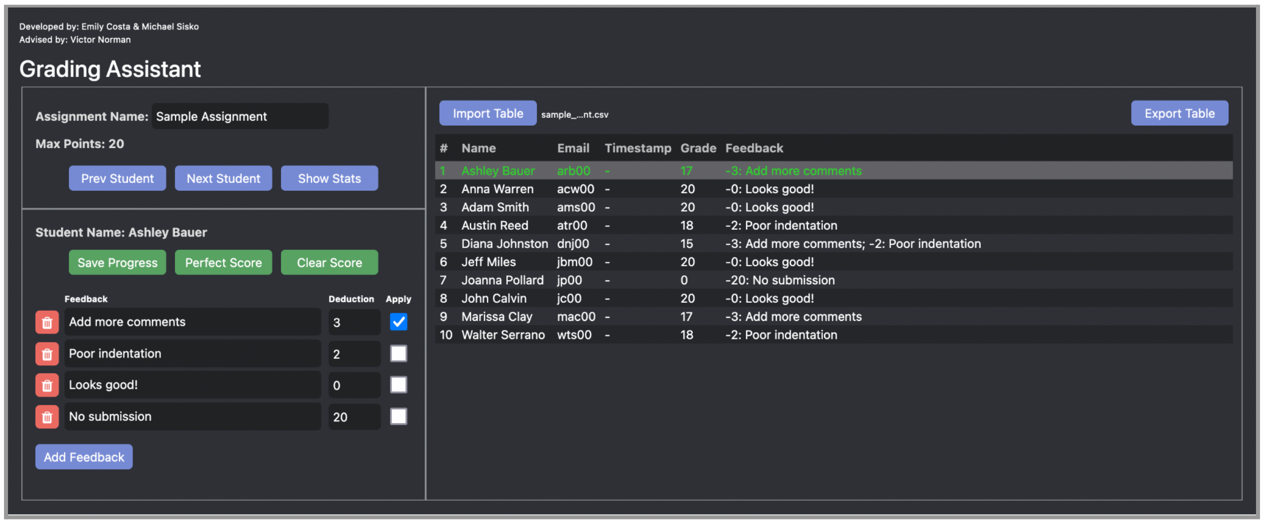
With our application, we plan to make the process quick and painless. Often students make the same error, and without our app, it is tedious for a grader to modify feedback should they change their mind. We expect our application to save graders and professors time through features that allow for efficient operation and minimize busywork wherever possible. We believe our application enables professors and graders to maximize what they enjoy without the needless hassle of the current grading software.

Solution Design

For our design, our application has a left panel that allows the user to add feedback and a right panel that displays a table of students. On the left panel, after you import a table, you can apply feedback to a particular student by inputting a string and a deduction value, then checking a box to insert it into that cell. Graders can create, update, and delete options at any point in the process. Furthermore, we implemented feedback consistency, meaning that feedback value and wording changes will be correspondingly reflected wherever called. When the grader finishes the class, they can save the feedback into a CSV file to reupload to Moodle. Graders can also name assignments, reflected in the title of the CSV download along with a timestamp.

With our application, a student-grader or professor would download the CSV file for the assignment from Moodle; import the file where it is parsed to a table to display each student's information. The table shows the student's name, email, submission timestamp, and feedback. The CSV file does provide an image URL of each student, but we have decided to omit this information from our application to promote objective grading. All feedback strings and deduction values are stored temporarily in a multidimensional array under the hood. From here, the grader can navigate through the list of students, enter feedback strings and the deduction value, and apply to students as necessary. By selecting the relevant custom feedback, this process eliminates the redundancy of entering the same feedback across multiple submissions. Our application keeps track of the feedback given to each student, so if the grader modifies their feedback or deduction value, the change is global. Finally, the application displays assignment statistics: the frequency of a feedback, the frequency of a grade, average, minimum, and maximum scores. Once the grading for an assignment is complete, the grader can export the modified CSV file and upload it to Moodle.

Figure 1.0 - Potential Application State



Developed by: Emily Costa & Michael Sisko
Advised by: Victor Norman

Grading Assistant

Assignment Name: Sample Assignment
Max Points: 20

Student Name: Ashley Bauer

Feedback	Deduction	Apply
Add more comments	3	✓
Poor indentation	2	
Looks good!	0	
No submission	20	

sample_...nt.csv

#	Name	Email	Timestamp	Grade	Feedback
1	Ashley Bauer	arb00	-	17	-3: Add more comments
2	Anna Warren	acw00	-	20	-0: Looks good!
3	Adam Smith	ams00	-	20	-0: Looks good!
4	Austin Reed	atr00	-	18	-2: Poor indentation
5	Diana Johnston	dnj00	-	15	-3: Add more comments; -2: Poor indentation
6	Jeff Miles	jbm00	-	20	-0: Looks good!
7	Joanna Pollard	jp00	-	0	-20: No submission
8	John Calvin	jc00	-	20	-0: Looks good!
9	Marissa Clay	mac00	-	17	-3: Add more comments
10	Walter Serrano	wts00	-	18	-2: Poor indentation

Description: Here we see a few potential options for feedback for an assignment; the feedback options are displayed in the table of students on the right. This snapshot displays some of the currently available functionality the application provides. The names on this list are not real students.

Design Norms

For the applicable design norms, we decided that cultural appropriateness through relieving the burden of administrative time, transparency through the clarity and execution of the design, and trust for its reliability, were the most relevant given our application's purpose and use cases.

Cultural Appropriateness

Our application alleviates the burden on the staff that organizes the curriculum. Teaching, guiding, and mentoring students require plenty of time, not to mention that professors must also grade the work for all students. With the grading that professors have to do, our application will reduce the burden of administrative tasks. Since our application will help to lessen the load, we believe it is conducive to the success of students and professors alike. We believe our application helps to extend our professors and allows them more time to inspire our students. Furthermore, we refrained from including images of students to be displayed in the grading table to promote privacy and objectivity. We feared that the inclusion of images makes way for discrimination beyond the quality of one's submitted work.

Transparency

We are also confident our application is easy to learn and requires little training. Since our target audience is professors of any discipline, teaching assistants, and student-graders, we assumed that most of our users were not tech-savvy. Considering this, we intended our application to be simple to prevent confusion. Also, we aimed for our documentation to provide ample information on the inner workings to allow others to build, modify, and expand our existing design however they see fit.

Trust

We believe our application is reliable and is not prone to failures. We did thorough tests on our design through Cypress and various user tests to provide reasonable trust in the application's reliability. We believe that our users can consistently and reliably use our application for the intended purpose of grading a set of students. Some helpful practices we have implemented include the incremental and thorough testing of each application feature. This process allowed us to eliminate many potentially application-breaking design flaws sooner than later.

Developmental Approach

Through our project, we used the services and libraries of Angular, Bootstrap, Apex Charts, Cypress, and GitHub. For the design of our project, we implemented it as a web application built using Angular typescript. We decided to use Angular for our development because we were familiar with the framework and thought it provided most of the functionality and capabilities we needed for application development. Throughout our project, we have implemented end-to-end (e2e) testing to test each of the functionalities our application introduces to ensure that each component is covered and error-handled properly. This process of implementation followed by thorough testing allowed us to mitigate several bugs before pushing our work into the main codebase.

Problems Encountered

One of the issues we encountered early in the project development was the best way to parse a CSV file into an easily readable, accessible, and indexable table. We were stuck on this problem for a few weeks before figuring out the best method of attack. Eventually, we landed on a design that parsed the CSV file into an array of student objects that was easily read and displayed through the usage of a mutable data service that could handle, create, read, update, and delete functions. Similarly, we struggled with the interplay of the two tables that contain student feedback and student assignment statuses. Both issues were solved across a couple of weeks with advice from our advisor Victor Norman and various research online. Our solution was to store a boolean array within each student object corresponding to the feedback array. Whenever the user applies a feedback option to a student, that specific student's boolean array would be updated. Other problems we encountered involved how to best display statistics and feedback for the table, but this was quickly resolved after a team meeting and discussing potential solutions. We decided to use a simple histogram to show grade frequency and display the grade statistics in a table.

Application Testing

Across the duration of our project, we implemented end-to-end development using Cypress. This process allowed us to streamline the testing of many anticipated user stories. Some of these include importing and exporting the CSV file to and from the filesystem, using each of the buttons, and several other error handling type events that could occur during application usage. We also intermittently did some in-person user testing where we provided the application to a user and instructed them to complete specific tasks. We would observe the difficulty required to perform each task, and ask for their objective feedback on their experience using the application.

Demonstration Video

Please view the app demonstration here.

Potential Developments

Some future developments could include persistent storage of progress. As the application currently stands, if the user wanted to close the application and return later or revisit their progress, they would see an empty application state on return. For future development, saving partial progress would be ideal, as this would save feedback items like the state of the table, assignment title, and other relevant data or metadata. For the application, other minor features include auto-deductions based on submission time and sorting columns alphabetically by toggle-clicking the headers. Beyond this, modifications may be in the realm of preference, a persistent settings page that could include custom styling, default feedback options, and others.

Acknowledgments

Special thanks to Professor Victor Norman for advising throughout the project and coming up with the idea and the proposed plan of attack. Thanks to Charles Kornoelje, Noah Cummisford, Micah Meindertsma, and William Terpstra for volunteering as application testers and providing feedback pertinent to the app’s development. Lastly, thank you to Andrew Feikema for helping with asynchronous functions in the CSV parsing service.