ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE
Paper ID #37397

Lessons Learned: Developing Homebrew Software Tools to Enhance and Combine Grading, Assessment, and Research

Benjamin Daniel Chambers (Associate Professor of Practice)

Dr. Ben Chambers is an Associate Professor of Practice in the Department of Engineering Education at Virginia Tech, and Director of the Frith First Year Makers program and of the Minecraft Museum of Engineering. His research focuses include creativity-based pedagogy, the interactions of non-humans with the built environment, and the built environment as a tool for teaching at the nexus of biology and engineering. He earned his graduate degrees from Virginia Tech, including an M.S. Civil Infrastructure Engineering, M.S. LFS Entomology, and a Ph.D. in Environmental Design and Planning.

David Gray

David Gray is an Associate Professor of Practice in the Engineering Education Department at Virginia Tech. David is currently serving as the Assistant Department Head for Undergraduate Programs. Dr. Gray teaches in a two-sequence Foundations of Engineering course, several courses within an Interdisciplinary Innovation Minor, and is leading the new Interdisciplinary Senior Design Capstone course within the College of Engineering. David maintains an active undergraduate research group. His research interests focus on teamwork and interdisciplinary curricula.

Matthew B James (Associate Professor of Practice)

Matthew James is an Associate Professor of Practice in Engineering Education at Virginia Tech. After working in the Civil/Site Development engineering field for a number of years, he returned to Virginia Tech to pursue teaching. His primary role is teaching within the first-year general engineering undergraduate program. He also is interested in study abroad, expanding service learning opportunities for students, and serves as the faculty advisor for the Engineers in Action student design team.

Stephen Moyer

Po-Jen Shih

Lessons Learned: Developing Homebrew Software Tools to Enhance and Combine Grading, Assessment, and Research

1. Introduction

This lessons learned paper describes the development and deployment of software tools designed to facilitate grading, assessment, and research efforts by interfacing the standard learning management system (LMS) used by our university with spreadsheet-based systems developed in-house. Faculty in instruction-focused roles are charged with providing the high quality, timely feedback that enables student success [1]. They may also need to capture student performance for internal and external assessment purposes or pursue their own educational research. Each of these efforts carries significant overhead in terms of time and energy. As such, time-saving measures are constantly being developed and explored, and are a major appeal of an LMS [2]. There is a large body of work in automated grading systems beyond just multiple choice, including short answers [3], essays [4], and computer programming [5]. However, manual grading is still the norm, and tools to support it are integrated into LMSs such as the SpeedGrader function in Canvas [6]. There is also a long history of embedded assessments or other combinations of assessment and grading efforts to reduce faculty workloads or streamline processes, e.g. [7], [8], [9]. Additionally, faculty conducting data collection for internal assessment and educational research often wish to evaluate aspects of student submissions for their own curiosity and development purposes, in ways that are not related to grading, and therefore are not captured in grading rubrics.

In our university, the official LMS, Canvas [10], currently has limitations both in its ability to extract assessment metrics and to modify rubric criteria and format during grading. What was needed was a grading tool that included rubric lines that could be designated for either students or internal review, allowed for rapid display and analysis of rubric metrics, and automatically linked with our LMS. After separate efforts to develop tools for their particular needs, three faculty members discussed their needs and wants for these tools and worked with two graduate teaching assistants (GTAs) to create generalized and flexible tools that all faculty in the department could employ. These tools were mostly spreadsheet based and coding was done with Google Apps Script. In the two semesters since sharing the generalized versions, the tools have been adopted by several faculty and have successfully supported grading, course administration, assessment, and research efforts.

2. Department context

The faculty and graduate students involved in the development of these tools teach in the general engineering program at a large mid-atlantic university. All incoming engineering students pass through a two-semester introductory engineering course sequence. The program serves well over 2000 students per year, supported by a team of about 15 instructional faculty, each working with at least one graduate teaching assistant and undergraduate grader. Faculty have one to four sections of 72 students each per semester. The team operates in a model of consistency with autonomy, working with a basic course framework and requirements, actively sharing content and discussing course design, and modifying the course to suit their individual strengths and expertise. As faculty in a department focused on engineering education, most instructors have

some expectation to conduct scholarship of teaching and learning. We also support program assessment and accreditation with assignments and completed grading rubrics. Our university has annual internal assessment requirements for our courses, which helps maintain our teaching standards while avoiding some common traps related to preparation for intermittent accreditation visits [11].

Research using course data is supported by a “blanket” consent form for research on materials from the first-year courses and approved by our IRB. At the beginning of their first semester, students have the option to provide informed consent. The IRB reviews any proposed research for existing consenting data. We then de-identify submissions and remove non-consenting submissions.

Our university subscribes to Canvas, a popular LMS. It is extensively used in the first-year program and plays a central role in our first-year engineering course administration. The LMS is used to share all course materials, make announcements, and submit all student assignments where possible. The caveat to this is that most documents are supplied through Google Drive files, though links are made available within the LMS.

3. Development of tools

The Canvas LMS has an application programming interface (API) to support modifications and an online community of instructors who seek their own modifications in response to needs not yet officially supported by the LMS [12]. For example, it does not have the ability to generate detailed reports on specific rubric lines, nor is it built to support rubric adjustment after grading has begun. These are needed for when instructors discover errors in rubric construction or assignment instructions, or wish to change the weight of a particular rubric line after seeing the results of grading. The LMS also does not yet have any built-in features for summarizing student performance based on individual rubric items.

Several faculty members in our department have built tools using Microsoft Excel, Google Docs, and Google Sheets to support grading efforts in a way that ensured rubrics were appropriately designed and consistently applied. Similar tools were also created for various other administrative support functions, such as copying or migrating shared Google Drives between semesters, emailing individualized files to students, and processing and de-identifying student submissions for research purposes. We quickly found ourselves discussing and sharing our tools with each other and borrowing elements to support our individual purposes. Descriptions for a selection of successfully deployed tools are included in Appendix A.

As other faculty became interested in these projects, it became clear that we would benefit from standardizing some of the grading tools. After determining a set of criteria and standards for developing a more universal set of tools, which included guidelines for ease of use and adaptability, two GTAs were given the task of synthesizing the various existing grading tools while adding a few desired features. The result was grading tools in Google Sheets that allow graders and instructors to review various grading statistics, rapidly modify rubrics, use and apply standard comments, and automatically generate, share, and upload complete rubrics for each student or team to the LMS. There are currently two grading tools in use with some minor differences according to instructor preference, but they are functionally quite similar and thus

combined for the sake of this paper. Detailed information on the functions included in the grading tool is included in Appendix B and example interface images in Appendix C.

4. Deployment and outcomes

Upon completion, the tools were shared with other teaching faculty and GTAs in the department. A particular focus of this sharing was the grading tool which was seen as the most broadly useful. The tools were discussed during a semester kickoff meeting for instructional faculty, and then continued to be promoted at monthly meetings. One GTA who worked on development also held a seminar for the other GTAs to teach them how to use it. Several additional faculty members adopted the grading tool and continue to use it a year later. Some newly hired faculty members have also expressed interest or adopted the tool as well. For this Lessons Learned paper, the authors reflected on their experiences and the conversations they have had with other faculty using the tool, and synthesized these outcomes and lessons.

The tools, and the grading tool in particular, have reduced the time required for course administration activities. Rubrics are now easily copyable and adjustable during grading without having to go back and correct each individual student grade that had already been input. The back-end file management and emailing tools have also saved hours of tedious setup work each semester. This carries through to university assessment projects which require extracting results on specific rubric lines from several assignments at the end of each semester. For teams using the LMS alone, this was a tedious task for graduate students that usually took at least half a day. Those who use the grading tool are now able to quickly complete similar regular tasks of uploading student feedback, regardless of student numbers. For example, in Spring 2018, it took one of the authors about 33 minutes to fully engage in manually uploading detailed feedback (grade breakdown and detailed comments included) on one programming assignment (with two problems) for a total of 57 students. The grading tool has reduced this labor to a few minutes. We normally have 72 students per section, for which we'd expect about 42 minutes per assignment. For a semester with ten such assignments, that would mean about 7 hours of tedious feedback uploading work per section. For a single faculty load of four sections per semester, that is meaningful savings.

Instructors with multiple GTAs and undergraduate graders who wanted to confirm grading consistency were formerly only able to compare overall final scores on assignments between graders. With our tool the team can review specific lines to identify if and where inconsistencies occur and also adjust those lines, their values, and the student scoring in a matter of minutes. Graders working simultaneously can also check themselves for consistency. Similarly, comments and feedback can be easily standardized and edited with this tool. Under the basic LMS system, comments are input in a comment box and must be read and adjusted individually if there is a need. With the grading tool, we can write (and edit) standard comments, quickly scroll through specific individual comments or search them for keywords, and upload and manage all feedback in batches.

In addition to major time saving on course administration, the grading tool has made it much easier for faculty to perform research and assessment based on assignment responses. The tool allows for quick deidentification and filtering for consent. More importantly, in addition to graded rubric items, the grading tool can include rubric lines for coding characteristics of responses that are excluded from reports generated for students. These provide faculty the

opportunity to explore possible research questions during grading. The tool is also easily used for more detailed research coding in the same fashion. So far, the faculty using the tool have explored numerous minor assessments of assignments that have resulted in a variety of improvements to those assignments. These explorations conducted during grading have also led to several research projects that utilized assignment responses as data. These explorations and the grading synergies have so far assisted one of the teaching-focused faculty in publishing two conference papers and one journal acticle.

5. Lessons for other faculty

We hope that this work inspires other faculty to consider ways to leverage their programming and spreadsheet skills and the flexible aspects of their own LMSs to save themselves time and combine efforts. Before rushing off to start coding, though, there are several things that should be considered.

Adoptability is important if you want to share your work with your colleagues. The first iterations of our tools were seen as overwhelming by some folks who were less confident in their programming skills, or who had through years of experience streamlined their own processes. We also noted that the timing of sharing tools had an impact. During onboarding and semester preparations, some faculty and GTAs were inclined to postpone learning new tools. On a related note, while homebrew systems allow teams to address their own unique set of needs, these systems also create challenges with maintenance as faculty and GTAs move on to new jobs, as LMS features change and grow, or as structures or methods within APIs are changed. Expected use rates and lifespan should be considered before starting a project, as well as repositories of existing tools.

Developing new tools can take a lot of time. We got excited about what we were doing and had fun with it, but it did take time away from other tasks. It has paid off for us, but if we hadn't been able to finish developing the tools or had they not worked well, the time might have been better spent elsewhere. We discovered during this process that the online communities for the LMS and Google Scripts were excellent resources and many components of our tools were informed by prior work. Before starting any similar project, it is a good idea to check whether someone has already done part or all of it, or whether the LMS is planning to implement a similar feature in the near future.

Overall, this project has proven valuable in saving faculty time on grading, course administration processes, and facilitating career-advancing research projects. We have been pleased with our results and encourage others to seek or develop their own process improvements and synergies. We are also happy to share our tools upon request.

6. References

[1] V. J. Shute, “Focus on formative feedback,” *Review of educational research*, vol. 78, no. 1, pp. 153–189, 2008.

[2] S. Lonn and S. D. Teasley, “Saving time or innovating practice: Investigating perceptions and uses of Learning Management Systems,” *Computers & education*, vol. 53, no. 3, pp. 686–694, 2009.

[3] S. Patil and K. P. Adhiya, “Automated Evaluation of Short Answers: a Systematic Review,” *Intelligent Data Communication Technologies and Internet of Things*, pp. 953–963, 2022.

[4] J. G. Borade and L. D. Netak, “Automated grading of essays: a review,” in *International Conference on Intelligent Human Computer Interaction*, 2020, pp. 238–249.

[5] H. Aldriye, A. Alkhalaf, and M. Alkhalaf, “Automated grading systems for programming assignments: A literature review,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, pp. 215–221, 2019.

[6] Instructure, “How do I use SpeedGrader?,” https://community.canvaslms.com/t5/Instructor-Guide/How-do-I-use-SpeedGrader/ta-p/757. 2022. [Accessed 22 Mar 2022].

[7] D. Lalush, C.F. Abrams, P. Mente, M. McCord, H.T. Nagle, E. Loboa, S. Blanchard, “Rubrics Cubed: Tying Grades To Assessment To Reduce Faculty Workload” in *2004 ASEE Annual Conference*, Salt Lake City, Utah. 10.18260/1-2--14030, 2004.

[8] W. A. Richman and L. Ariovich, “All-in-one: Combining grading, course, program, and general education outcomes assessment,” *National Institute for Learning Outcomes Assessment Occasional Papers*, vol. 19, 2013.

[9] K. Dahm, “Combining the tasks of grading individual assignments and assessing program outcomes in project-based courses,” *Journal of STEM Education*, vol. 15, no. 1, 2014.

[10] Instructure, “Canvas by Instructure,” https://canvas.instructure.com. 2022. [Accessed 22 Mar 2022].

[11] K. Shryock and H. Reed, “ABET accreditation: Best practices for assessment,” in *2009 Annual Conference & Exposition*, 2009, pp. 14–148.

[12] Instructure, “Canvas Developers Group,” 2022. https://community.canvaslms.com/t5/Canvas-Developers-Group/gh-p/developers. [Accessed 22 Mar 2022.]

Appendix A: Table of descriptions of selected successful tools

	General Description	**Purpose**	**Inputs**	**Notes**
Batch File Rename	A simple script using Microsoft Excel and Visual Basic for Applications (VBA) that takes a list of students with study ID codes along with folder locations as input, then creates renamed copies of the files in a new folder. Students without an associated code are skipped.	To perform research and assessment on student submissions, it may be necessary to remove identifiers such as file names. Batch downloads of submissions to the LMS automatically rename files with student information.	Student information list with name and email as formatted in Canvas, list of study ID codes	All written assignments are submitted through the LMS, and batch downloads rename files with the student names. We are able to simplify de-identification by removing “Name” from submission templates and relying on Canvas to add student names.
Drive Folder Copy	A script in Google Apps based in a Google Sheet, that copies files and subfolders from one Google Drive folder to another.	For student privacy, logistical, and archival reasons, our instructors typically create new shared drives for their courses each semester. Copying over individual files can be tedious.	Source folder, destination folder	This was in response to older versions of Google Drive and Google Drive for Desktop, which did not permit copy/paste of files between folders in the same way as the Windows File Explorer.
Individual Email Generator	A script in Google Apps based in a Google Sheet that creates and sends emails with attachments to individual students.	For some assignments, we wanted students to have individualized files to prevent plagiarism. This was created to make distribution easy for instructors with lots of students.	Student name, email, subject line, body components, attachments.	We added a send confirmation on the sheet for each student in case of process termination. Google Apps Scripts terminate after 30 minutes
Canvas Gradesheet Interface	A script in Google Apps based in a Google Sheet that automates information transfer between Sheets and Canvas. Users can download course rosters to the spreadsheet, or upload/download rubric line items and scores.	To enable spreadsheet-based grading. Assignments can be scored in either Google Sheets or Canvas, and then automatically synced between the two. Facilitates grade archiving, and the scoring of multiple assignments.	Canvas API authorization key, Canvas course number, and Canvas assignment number.	Custom-defined tool menus in Google Sheets allow for downloading course roster from Canvas to Sheets.
Canvas Team Upload	A script in Google Apps based in a Google Sheet that takes a list of students with teams and imports it to Canvas.	To streamline the generation of and population of teams in Canvas. This was made prior to release of native Canvas .csv import functionality for team generation.	Google Sheet spreadsheet with student email, Canvas course ID, and team name.	This works as an excellent intermediary between CATME Team Generator and Canvas LMS.

Appendix B: Table of grading tool design needs and solutions

Need	**Solution**
Limit manual setup with new assignments	Script for automatic setup. Inputs: rubric manually entered into a template, access token for LMS. Script accesses LMS, downloads student and team information, and builds the user interface tab
Suitable for team or individual assignments	Toggle switch in script
Globally adjust point values and rubric line descriptions, without any re-entry for specific students	Binary entry for each student for each rubric line, with master columns containing description and point value for each rubric line
Review statistics and distributions of grades for each rubric line during grading	Column containing sum for each rubric line, using the binary entry by graders
Compare grading distributions between multiple graders working on the same sheet	Pre-grading assignment of students for each grader, sum columns for each pre-assigned range for each rubric line
Apply standardized comments and be able to modify the standard comment during grading according to observations	Set of rubric lines without point values, separated so that the script could tell when graded lines ended and comments began. Applied and modified like the graded rubric lines
Code responses for assessment or pre-research explorations, using rubric lines that are not shared with students	Rubric lines marked as internal placed below the comment rows with indication to the script to skip
Real-time validation, confirming multi-option rubrics were correctly applied	Tally totals with conditional formatting for visual cues to graders
Pre-upload validation of grading	Checks built in to the script, confirming that rubrics were complete
Automatic rubric generation	Script for automatic rubric generation created a file for each individual or team, either as a Google Sheet or a .pdf, according to instructor preference
Automatic rubric sharing	Script for sharing rubrics granted view permission, generating an automatic email, and posting a link as a comment on the submission in the LMS or uploading a .pdf, according to instructor preference
Automatic uploading of grades into LMS	Script for importing grades to LMS
Generate rubrics for individual students or teams in the event of late work or grade corrections	Script toggle for full class or specified individuals
Correct and remove rubrics with mistakes or necessary changes after posting	Script to batch remove automatically posted comments, links, and files from LMS in the event of an instructor error in grading, or a universal change to the rubric after posting

Appendix C: Example grading tool interface

	A	B	C	D	E	F
	Ethics Case Study			**Team/Student:**	**Last, FirstA**	**Last, FirstB**
				Topic:	Factory Farming	Drone spying
				Grader:	GraderA	GraderA
	Item	**Detail**	**Points**	**Count**		
	Question 1	[Identify an ethical dilemma or issue from th	1.00	130		
	Exemplary	Ethical dilemma is clearly and thoroughly identified and related to the given case study. Demonstrates evidence of careful reflection or additional research.	1.00	70		1
	Good	Ethical dilemma is clearly defined, with little to no extraneous or irrelevant information and/or somewhat unrelated to the given case study.	0.90	48		
	Some Notable Issues	Ethical dilemma is present, but difficult for the reader to understand and/or not clear in a few areas.	0.80	10	1	
	Significant Issues	Ethical dilemma is present, but is not clearly identified and/or seems to be very unrelated to the given case study.	0.70	2		
	Missing	Section not present.	0.00	0		
	Question 2	[Explain why the above answer is an ethica	1.50	130		
	Exemplary	Explanation includes solid reasoning for why the chosen issue or dilemma was chosen. Answer is clearly related to ethical considerations and backed up by information included in or inferred from the provided case study. Demonstrates evidence of careful reflection or additional research.	1.50	76		1
	Good	Explanation includes solid reasoning for why the chosen issue or dilemma was chosen. Answer is mostly related to ethical considerations and/or mostly backed up by information included in or inferred from the provided case study.	1.35	50	1	
	Some Notable Issues	Explanation is present, but difficult for the reader to understand and/or not clear in a few areas. Often not backed up by information from the case study.	1.20	1		
	Significant Issues	Explanation is present, but difficult to follow. Little to no information from the case study used to back up the claim.	1.05	3		
	Missing	Section not present.	0.00	0		
	[RUBRIC ABOVE]		**Max Points**	**Sections Graded:**	2	2
			2.5	**Days Late:**		
	Grader Input		**Average Grade**	**Total Points:**	2.15	2.50
	Enter value or text		95%	**Grade Percentage:**	86%	100%
	Flag/tally with a 1			**Comments:**	Impact on Stakeholc	
				[TALLY COMMENTS]		
				Please see comments on the assignment in [LMS]	1	1
				Ethical frameworks descr		1
				Some minor citation issue		1
				No citations included	1	
				No actual ethical framewc		
				Template not used proper		
	Assessment					
	Ethical theories discussed			Deontology	1	
				Consequentialism		1
				Virtue Ethics		
				Pragmatic Ethics		

Figure 1: Grader interface with some example entries.

	A	B	C	D
1		**Input Values**	**Example**	**Notes**
2	**Rubric Sheet Name:**		Super Assignment	Sheet name for the assignment rubric
3	**Max Points:**		5	Manually adjust in grade sheet if this changes for some reason after populating the rubric
4	**Course Info:**			Will be written to the final graded rubrics along with the 'Topic' in the grade sheet
5	**Course ID:**		112233	Can be found in the Canvas course URL
6	**Assignment ID:**		998877	Can be found in the Canvas assignment URL (only used for uploading grades)
7	**Canvas Token:**			You can use the same token throughout the semester (Canvas --> Account --> Settings --> Approved Integrations --> New Access Token)
8	**Group Set Name:**		Teams M1	The unique name assigned to the Group Set you created in Canvas for your Course (leave blank for individual assignment)
9	**Individual Re-Grade:**		Smith, John Jacob	Copy-Pasta student/team name from grade sheet or score sheet (optional for regrading individual student or team)
10	**Late Calculated in Canvas:**	☑		If checked, the graded rubrics will not show 'Days Late' (optional for instructors who don't want to use this tool for LATE calculation at 10%/day)
11				
12		**Progress (%)**	**EST total time (minutes)**	**Notes**
13	**Populate Students/Teams:**	100	[2] students, [10] teams	
14	**Build Graded Rubrics:**	0	[60] students, [15] teams	If the rubric building times out, run again until all students in the score sheet have scores. It will automatically pick up where it timed out.
15	**Upload Grades:**	0	[10] students, [10] teams	Scores and comments are uploaded for each student, regardless if a team assignment
16	**Add Viewer Permissions:**	0	[15] students, [5] teams	
17				
18		**Count**		
19	**Students:**	145		** **Do not modify rows and columns on this sheet, only change the 'Input Values' above to set up tool** **
20	**Teams:**	0		
21				
22		**How To Use**		
23	**Populate Students/Teams:**	Grading Tools --> Populate Students: This will get students (or teams) from Canvas and populate the Grade Sheet and Score Sheet (must already be cleared of student info)		
24	**Populate Rubric:**	Grading Tools --> Populate Rubric: (limited functionality) Will set the rubric name as the header of the Grade Sheet, and set the Max Points on the Grade Sheet. Does not currently replace rubrics for you.		
25	**Build Graded Rubrics:**	Grading Tools --> Build Graded Rubrics: After grading is complete, this will generate individual graded rubrics and fill in the students' scores and rubric URLs on the Score Sheet		
26	**Upload Grades to Canvas:**	Grading Tools --> Upload Grades to Canvas: ALWAYS read the prompt to verify course and assignment information which is pulled from Canvas using the ID values above. Uploads students' scores from the Score Sheet to Canvas for the assignment ID above, URLs for rubrics are uploaded as a comment on their assignments		
27	**Give Viewer Permissions:**	Grading Tools --> Give Viewer Permissions: After uploading grades, but before posting them, this will give students viewer permissions for their individual graded rubrics		
28	**Build Individual Rubric:**	Grading Tools --> Build Individual Rubric: Optional, will use the 'Individual Re-Grade' input value above to rebuild a new rubric for an individual student or team		
29	**Upload Individual Grade:**	Grading Tools --> Upload Individual Grade: Optional, will use the 'Individual Re-Grade' input value above to upload grade and comment with rubric URL for an individual student or team		
30				
31	**Suggestions for Use:**	Create three copies of this workbook. One so you have an original, one for student assignments, and one for team assignments. For subsequent assignments, create new copies of ones already graded.		
32		Populate students/teams, so the score sheet is set up. You should probably do this for each new assignment before add/drop just in case. This will also clear out the grading section of the Grade Sheet.		
33		Fill in the 'Input Values' above. Run the 'populate rubric' so the name and max points values are set in the Grade Sheet.		
34		Replace the 'Example Assignment' sheet contents with your own rubric, and rename the sheet to be consistent with the 'Rubric Sheet Name' input value. You should also put this name in the header cell on the Grade Sheet.		
35		Adjust the rubric on the Grade Sheet with the same rubric you are using on the dedicated sheet (important that they are the same). You may have to tweak some formatting in the grading section.		
36		Grade the assignment, using the comments row for unique comments, and add custom tally-style comments as needed (can be really handy for repeat comments or codings).		
37		VERIFY that the section counts for the individual columns are all the same, and that the counts above each rubric item are all the same. This will help you catch missing sections.		
38		Add any of your own metrics at the bottom of the Grade Sheet as desired (e.g. average grade by grader =AVERAGEIF(E54:54,"=John"), count missing assignments =SUMIF(E54:54,"=0"), etc.).		
39		Build the final rubrics. For student assignments, this can take awhile and you might have to run the command multiple times when it TIMES OUT after 30 minutes. The code will handle picking up where it left off.		
40		Spot check a few rubrics at random (just a good habit), then upload grades to Canvas. DONT POST grades yet, it's best to give viewer permissions first so students aren't emailing you requests for access.		
41		Give viewer permissions, then post grades. Make a copy for the next assignment your team will be grading (also a good habit), clearing out student info if desired.		
42		Let me know if you need any help! My email is [redacted]. I also have a long list of desired features, and I would appreciate your feedback and ideas for future development.		

Figure 2: Automated setup interface with instructions and suggestions.