

Pre-Proposal for a Capstone Project Intended to Satisfy the Requirements for an ALM in Digital Media Design at Harvard Extension School

Dave Morgan

July 30, 2019

This document is a pre-proposal for Dave Morgan's capstone project, intended to satisfy requirements for an ALM in Digital Media Design at Harvard Extension School. The proposed project will involve the refactoring of a website authored a decade or so ago, and visible at <http://qwizm.org>.

Qwizm contains a number of quizzes or assignments, each comprised of questions having multiple parts. Input values to these questions are pseudo-randomized, and derived from a student ID number so that each student receives 'unique' question inputs. Answers are numerical. To date, these questions have focused on problems in introductory engineering courses.

The aim of this refactoring is to make the site more accessible and user-friendly, and to take advantage of recent advances in technology allowed by HTML5, CSS3 and ES6. The proposed site will make it more convenient for instructors to create new questions and quizzes of their own and more convenient for students to use.

Candidate Information

First Name: Dave

Last Name: Morgan

ID: 10823943

Education:

BSc. Applied Mathematics (2003),
University of Calgary, Calgary, Alberta, Canada

BSc. Computer Science (2003),
University of Calgary, Calgary, Alberta, Canada

Current Employment: Retired

5-Year Deadline: May, 2020

Capstone Track

ALM in Digital Media Design

The Project

Background

In 2005, I began teaching introductory engineering courses at a polytechnic offering two-year diploma programs for prospective engineering technologists. The courses I taught were problem-based and examples, exercises and assignments typically required multiple-step mathematical solutions. The mathematics is not advanced, usually little more than high-school algebra and trigonometry, but many students find it challenging.

Attempts to ensure independent student work on assigned 'homework' are an ongoing battle. Students are able to find solution manuals online for most textbooks, or may subscribe to services that supply solutions. Instructor-created problem-sets tend to encourage group work, with the strongest member of the group providing direction to those less able — who may understand the process when it is presented to them but don't benefit from the exercise of thinking out a solution that builds from the same concepts yet is, perhaps only subtly, different from examples presented in class.

Many learning management systems (LMSes) offer question-types named 'arithmetical,' 'calculated' or equivalent, that provide pseudo-randomized question input values and marks the student submission based on that student's input values. For example, a simple question requiring the calculation of the area A of a rectangle with length L and height H might have inputs (and corresponding calculated answer) for each student, as shown below:

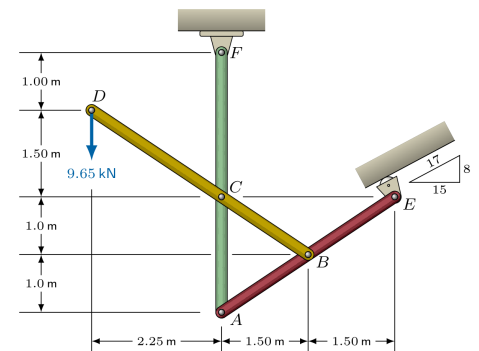
Student	L (m)	H (m)	A (m ²)
Adams	2.55	1.950	4.97 ✓
Brown	1.250	0.875	1.094 ✓
Clarke	4.20	1.600	6.72 ✓

Generally, questions in engineering courses are more complex and require several steps to reach the solution. In these cases, it is not often realistic to expect a student to arrive at a numerically accurate final answer without some feedback along the way. And, frequently, questions do not require a single final answer but several answers: such as calculating the tensile forces in bolts at the top and the bottom of an access hatch attached to a storage tank; or the determining of the reaction force in each of a number of connections in a rigid-member frame.¹

Table 1: Possible sample inputs for length L and height H , provided by an LMS for three different students, to find the area A of a rectangle, along with solutions that would be marked correct by the LMS.

¹ A typical engineering statics question requiring multiple numerical answers:

There is a frictionless roller at E and pinned connections at A , B , C and F .



Determine the magnitude of the reactions at all connections and the roller when a 9.65 kN force is applied vertically downwards at D .

LMSes don't generally offer question types that mark multiple answers although it does seem to be a much requested functionality (particularly on Canvas).^{2,3,4} Several years ago, the institution at which I worked was looking to switch to a new LMS; this multiple-answer functionality was strongly requested in the new LMS, not only by the engineering, and other, technology programs but also by several trade apprenticeship programs — but this functionality was not available in the LMS eventually chosen. And the chosen LMS stripped any user-generated JavaScript from files so there was no way for users to implement a hack as had been managed in our previous LMS.

Moodle⁵ is a popular open-source LMS that offers an impressive list of question-types⁶ but no mention of a random-input multiple-answer question-type. However, there is a Moodle plug-in (initiated in 2010 by Hon Wai Lau⁷) that provides the required functionality.⁸ I have built a handful of sample questions⁹ using this plug-in and consider it to be a realistic solution for those institutions that adopt Moodle as their LMS.

In 2008, I wrote a simple testing system,¹⁰ partially as an exercise for learning JavaScript. I had a specific set of requirements:

1. Users have access to a modern browser.
2. Questions are capable of 'marking' multiple parts (simply ✓ or X) to facilitate direction and progress through lengthy problem solutions.
3. Individual parts can be assigned an arbitrary number of marks, each part contributing to the question total.
4. Each question has randomized inputs, derived from the student's ID number. Logging in with the same ID number produces the same inputs.
5. Randomized inputs are critical to these questions but sometimes we need more than a random value, e.g., we may need to pick a pipe randomly from a set of commercially available pipe sizes. Qwizm can accommodate this requirement (by picking an index randomly from an array of pipe sizes).
6. All calculations are performed on the client side, in the browser. Nothing is stored between browser sessions. (Local storage in the browser was not then available.)

² <https://community.canvaslms.com/thread/5891>

³ <https://app-community.canvaslms.com/ideas/8172-multiple-numerical-blanks>

⁴ <https://community.canvaslms.com/thread/30523-multiple-part-formula-question>

⁵ <https://moodle.org>

⁶ https://docs.moodle.org/37/en/Question_types

⁷ <https://www.linkedin.com/in/hon-wai-lau-97508182>

⁸ https://github.com/jmvedrine/moodle-qtype_formulas

⁹ <http://eduk8r.org/moodle/mod/quiz/view.php?id=24>
User: learner, Password: Le@rner1

¹⁰ <http://qwizm.org>

7. Input values are super-imposed on the diagram. For example, a length would be marked 3.25 m on the diagram, rather than x_1 m marked on the diagram and $x_1 = 3.25$ m defined in the question statement. This functionality is not available in any LMS (to my knowledge) and is intended to remove one level of abstraction that can confuse the student.
8. Student answers must be correct to three significant digits although 5 or more should be carried for intermediate calculations. (In 2010, I updated the marking function to check for four significant figures in answers where the first non-zero digit is a 1, to match the practice in our course textbooks.)
9. There is a summary page, detailing all user entries, whether the entry is correct or not, and the marks earned for each part. Also, a total is calculated for the complete assignment.

Initial student response to these Qwizm assignments was mixed. Some embraced it immediately, viewing it as 'guided-learning' and enjoying the instant feedback. Others struggled with the intentionally demanding accuracy (three, or four, significant digits, with no tolerance) required for a correct answer — this followed the consistent pattern used in in-class examples. Initially, I had complaints of program 'bugs', i.e., Qwizm marking an answer wrong when the student was 'absolutely certain' that their answer was correct. For each class, I'd pick a particularly troublesome question and work through it on the board, demonstrating that it was indeed possible to get consistent ✓ marks the first time when sufficient attention was paid to significant figures.¹¹

Complaints diminished in number after these demonstrations, but students who aren't sufficiently accurate with their math often adopt a policy of trying adjacent final digits until they receive their own 'correct' checkmark.

(I have no remedy for this, due to other requirements I have chosen: to keep everything on the client side; and to allow students to reset the assignment — a button that will simply clear local storage — to start over, with a different ID number if desired.)

Motivation

Despite limitations, Qwizm has proved to be effective both as an assessment tool and as a learning tool. It continues to be used by the instructor who has taken over teaching my courses following my retirement from teaching. The refactoring proposed here is the first

¹¹ One topic, Forces Due to Static Fluids in an introductory course in Fluid Mechanics, can cause numerical problems. The formula

$$L_P - L_C = \frac{I_x}{L_C \cdot A}$$

is used to determine the very short distance, $L_P - L_C$. If the student uses $\frac{I_x}{L_C \cdot A}$ to calculate the value, the answer is given to the 5 or more required significant digits required for interim calculations. But, if the student calculates L_P and L_C first, to 5 significant digits, calculating $L_P - L_C$ may only have 2 or 3 significant digits.

For example, if $L_C = 1.2345$ m and $L_P = 1.2356$ m, then $L_P - L_C = 0.0011$ m which has only two significant digits.

This can cause problems in achieving the 'correct' result later. The question creator should be aware of these numerical issues which are not directly related to Qwizm.

part of a larger project (beyond the scope of this capstone) to build a desktop application that will enable instructors with no programming knowledge to develop their own questions and to combine questions to make quizzes.

Currently, question development is laborious and requires a knowledge of JavaScript, knowledge that instructors, in general, do not possess. A complex question can take several hours to produce and this lengthy development time is in large part responsible for Qwizm remaining positioned somewhere between a proof of concept and a complete course assessment resource.

Particularly time-consuming is the precise location of randomized inputs on top of a static diagram; this will be made much more efficient by the implementation of local storage, which can be used to remove the need to re-enter log in details each time that the question developer refreshes the application to view the effect of moving an input value just a few pixels in any direction to fine-tune the value location.

There are several motivations for this proposed refactoring of Qwizm: to speed up development time for new questions and quizzes, thereby encouraging further development of question banks; and to make the application more user-friendly from a student perspective, providing a responsive site that they may conveniently access on different devices and that retains previously entered values between browser sessions.

The primary motivation, though, is to realise an efficient codebase that could be generated by a future desktop application. This application would enable instructors from many disciplines to package and distribute a collection of numerical questions as a single html file. (For example, high-school physics teachers might find such a tool useful.) No programming knowledge would be required though the ability to build a formula, as in a standard spreadsheet, would be necessary.

Prior Experience with Qwizm

I accept the reality that there is no way to guarantee that all students work independently on assignments. Individualised question inputs do not solve the problem of unwarranted collaboration but they do help with learning. Even when the next step in a problem has been discussed in a group, each student does, at least, perform their own calculations — and to write out their own solution which is submitted together with Qwizm's printed summary sheet.

Although I have used these Qwizms almost exclusively for assignments that count towards a final course grade, some students choose to re-do them, possibly with a different ID number to generate different inputs, as review before mid-term and final exams. With a more comprehensive question library, they could also be used as an alternative to 'static' in-class examples or exercises.

For many instructors, the attraction of online testing is the associated reduction in time spent marking student work. This comes with some cost; not actually seeing student work until exam-time can be a time-saver for instructors but successful steering of students towards a logical and professional presentation of material is more difficult to assess.

Qwizm, with its instant feedback and traditional paper-based submission, does have some advantages:

1. The instant feedback from question parts encourages students to persist until they have the right process and the correct answer for that part before continuing. Indeed, students realise that there is little point in continuing since future results usually depend on the successful completion of earlier steps.

Prior experience with existing commercial LMS assignments had less successful outcomes as far as student motivation was concerned: students would complete an assignment online, submit it for marking, find they had scored, say, 70% but felt frustrated by having to retake the whole assignment, with fresh inputs to questions they had already completed successfully, to improve their mark. In contrast, Qwizm, with its continual feedback, encourages students to persist towards completion; it is common for students to get 100% on Qwizm assignments.

(Although completion of these assignments can be quite lengthy, in my courses they only count for 10% of the total course mark; that is substantial effort for a somewhat minor return in marks but students recognise that these are a good way to learn material and the time spent will serve them well at exam time.)

Other attempts at online assessment using offerings from textbook publishers were also a source of frustration; for the text that we tried this with, answers were frequently incorrect with no way for the instructor to correct them, most questions did not have randomized inputs but were simply the same as questions from the text, and students wondered what they were obtaining in exchange for their required subscription to the publisher's testing site.

2. Compared to traditional paper submissions, the marking burden on the instructor is dramatically reduced: marks are already calculated, ready for entry into an LMS or spreadsheet, and the written work can be quickly viewed for completeness, professionalism, etc. It is not unreasonable to process a student assignment in less than one minute.

Surprisingly, over the ten years that I used Qwizm in the classroom, I was not aware of any student using browser developer tools to retrieve the answer values stored in variables.

Technologies To Be Used

A combination of modern frontend technologies (HTML5, CSS3 and ES6) are all that will be required. A subset of jQuery and jQueryUI may be included to improve the look and style of the site. The UI will be relatively simple so there is little advantage in using a JavaScript library such as ReactJS.

Some Implementation Details

The following is a list of currently identified enhancements to the original Qwizm application.

1. Upon loading a quiz, the browser will check local storage for the existence of an object with the same numerical quiz identifier. If this object does not yet exist, a login screen will be presented to the user, prompting for a username and ID number. If the quiz object does exist, then the user has visited this quiz previously; the user will be logged in automatically and previous progress in the quiz will be loaded into memory.
2. The app is a single page with a row of tabs to navigate between questions. (jQuery may be used here.) There will be a tab to view the summary page and also a tab to clear the quiz completely, removing the quiz details from local storage and requiring a fresh log in to start the quiz again (possibly with a fresh ID).
3. The quiz object in local storage will be kept synchronized with a quiz object in memory, local storage being updated from memory whenever a new result is entered.
4. The quiz object will have properties for the subject (or course) and topic of the quiz, an optional due date and an array of question objects that define the questions in the quiz. Other properties will

specify the tolerance (if any) in the marking scheme and an array that maintains all question entries, marks earned for each question part, and a mark total.

5. Just as a quiz is a collection of questions, a question is a collection of question parts. Each question object will have a question statement, a path to an image, an array of question parts, declarations and constraints for the creation of input variables, equations for calculating the correct results from the input variables and marks assigned for each part.
6. Utilities for marking an entry right or wrong will be improved. If, as usual, an answer requires 3 significant digits, then an entry of 3.20 should be marked correct and numerically identical 3.2 should be marked incorrect.
7. The site will be responsive and provide layouts suitable for phones, tablets and laptop screensizes.

This list will expand as more opportunities for refinement or improvement are discovered; these are just initial considerations.