CS230 - Web Information Processing

Assignment 3

Assignment Release Date:	08-03-2020
Submission Due Date:	14-03-2020 (note: Extension from 12-03-2020)
Feedback Due Date (estimated):	30-03-2020 (for assignments that make Due Date)
Support Laboratories	Lab 05 (One Week)
Total Mark:	5%

This Assignment is worth 5% of the Web Information Processing CA Component.

This is an open-book, graded assignment. You may use online resources for reference purposes only to help with the assignment. Please cite all references as comments in your submissions. You cannot directly reuse HTML/CSS/JS **solution code** from online sources. **You must not engage with another student, in person or electronically (phone, social media, etc.) to secure assistance with this assignment. If you do so you will receive an automatic fail (0%).** We will perform similarity checks on submitted assignments to check for collaborative efforts. A reasonable attempt at this assignment will gain you 5% of your continual assignment marks. It is possible to gain extra credit (up to a maximum of 5%) for this assignment. Extensions, for any reasons

Assignment 03 - Working with Tabular Content

You are required to develop a **Student Grades Table** - an HTML/CSS/JS application that implements that manages hypothetical student assignment grades using an online styled table similar that shown in Figure 1, below.

The table should contain columns for the students' names (Name), ID numbers (ID), results (a grade between 0-100) for five assignments (Assignment 1, Assignment 2, etc.), together with a final grade (Average [%]) column that contains the arithmetic average of the five assignment grades in the preceding columns.

The table should contain initial default data (-) for ten hypothetical students; "-" indicates that a particular assignment has not yet been submitted. The table should be created using HTML, and styled using CSS (names and IDs should be left-aligned, headings should be centre aligned, and numbers should be right-aligned). All interactivity should be implemented using JS.

Only table cells containing grades should be "content editable", thereby allowing manual entry of the grades (every cell in all columns is editable except for the final aggregate column).

The table style (shown in Figure 1 below) should utilise a sans-serif font for all data presentation. Apart from cells denoting column headings (which have a dark grey background), all cells should have an alternating row-colour background with black text. Student names and IDs should be left-aligned, headings should be centre-aligned, and numbers should be right-aligned). Assignment cells not containing numbers (i.e. with a “-“) should be centre-aligned.

The following section details specific interaction requirements for the Student Grades Table.

Assignment 03 - Interactive Requirements

You are required to develop an HTML/CSS/JS application that implements the following functionality:

1. Your table should have the following dynamic functionality.

    (i) Automatically recalculate all final grade averages whenever a table cell is updated by a user.

        (a) The value should be rounded, not include decimal places, and be expressed as a percentage, for example, "55%" and not "55.3", "61.7%", etc.

        (b) Final grade averages that have a value below 60%, should be styled so that the grade font is presented using a white colour on a red background.

    (ii) Provide a count of the total number of assignments that have not yet been submitted, presented in a styled format, beneath the table. Furthermore, change the background of cells containing "-" (un-submitted assignment) to yellow. These functionalities should be automatically calculated.

    (iii) Automatically validate cell data when manually updated by a user. Cells containing erroneous data should default to being “un-submitted”, i.e. any input other that number between 0-100 should be replaced by a “-".

2. Clicking the “final grade” (Shown in Figure 1 as “Average[%]” column of the table should invoke a JavaScript function that toggles the presentation of the average grade for each student between the Percent Grade, American Letter Grade, or American 4.0 Grade, in accordance with the following rules:

    1. The title shown for the different representations will be “Average [%]” for the Percent Grade, “Average [Letter]” for the American Letter Grade, and “Average [4.0]” for the American 4.0 Grade.

    2. The conversion table, for the toggle, will be as follows:

Percent Grade	Letter Grade	4.0 Scale
93-100	A	4.0
90-92	A-	3.7
87-89	B+	3.3
83-86	B	3.0
80-82	B-	2.7
77-79	C+	2.3
73-76	C	2.0
70-72	C-	1.7
67-69	D+	1.3
63-66	D	1.0
60-62	D-	0.7
< 60	F	0.0

3. You should provide additional buttons either, on or below, the table, with the following functionality once parts 1-2 above have been completed:

    (i) A CSS styled button that inserts a new table row suitable for recording new student data. You may insert new rows after the last row of the table.

    (ii) A CSS styled button that inserts a new table column suitable for recording new Assignment grade data. This column requires a title. You can decide how you wish to accomplish the title allocation (automatic, content-edit, etc.). There should be another button that then retrieves that data and fills it back to the table in the state that it previously held. If extra rows or columns have been added, the table should revert back to its previous state when the table was saved (5 rows and 6 cells). All additional grade columns bus be inserted after the “Student ID” column, and before the final grade (Average) column.

Student Grades Table

Student Name	Student ID	Assignment 1	...	Assignment 5	Average (%)
Joe Bloggs	123456789	85	...	85	85
Jim Bloggs	987564321	-	...	-	0
			...		
			...		

Figure 1

Sample Table Styles (Assignment columns 2, 3 and 5 have been replaced by ellipses)

Assignment 03 - Extra Credit Functionality

4. If all of the functionality in 1-3 has been completed, you may attempt any the following for extra credit (there is a maximum of 5% extra credit obtainable):

    (i) Use JavaScript, to highlight a complete table data row when a student's name is selected. Clicking again deselects the row. Note you should not be able to select and highlight the title row! (1%).

    (ii) Use JavaScript, to highlight a complete table Assignment Grade column when a column's title is selected. Clicking again deselects the column. Note: that you should not be able to select and highlight the final (average) grade column! (1%).

    (iii) Use JavaScript, and any method of your choosing, to delete a data row selected by a user. Use JavaScript, and any method of your choosing, to delete an assignment column selected by a user. The function should ensure that the Final Grade column totals are updated following this deletion (2%).

(iv) Use JavaScript to provide additional functionality to insert a new row (as in 3(i) above). Users should be able to right click on any cell in a row, and be presented with a menu option to insert a row above or below the row. This is really only for students that want a significant challenge. Do not waste time on this unless you've done everything else quickly and have time (2%).

(v) Use JavaScript, to undelete the last deleted row or column. You should consider that once the row/column is undeleted, the button "loses focus", i.e. cannot be clicked again until another row/column is deleted. This is really only for students that want a significant challenge. Do not waste time on this unless you've done everything else quickly and have time (2%).

Assignment 03 - Development Notes

Please adhere to the following development requirements:

1. You may not use a CSS framework, such as Bootstrap, for this assignment. You may, if you wish, use the jQuery Javascript framework. If you use TypeScript, or similar, and translate to JavaScript, you need to provide all sources. Your app only needs to run on a desktop browser such as Chrome - you do not need to ensure it works on every browser (mobile browsers, for example).
2. You must comment your code, clearly indicating, how your code implements the solution described above in the "Assignment 03 - Requirements" section (and "Assignment 03 - Extra Credit" section if you are attempting additional functionality).

Please note that there are many sample (HTML/CSS/JS) solutions for implementing tabular (spreadsheet) functionality available online. While it is fine to consult these, and accompanying articles, for references, you may not re-use code from these projects. Please cite your reference sources in your codebase. We will search and identify online coding solutions to similar problems for the purposes of checking against submitted solutions in instances where we have concerns about code originality.

IMPORTANT SUBMISSION DETAILS

Before submitting your assignment students should check that their solution works in Chrome and/or Firefox. Please indicate the Browser, Lab Operating System (Linux/Windows) and Browser version used for testing (as a comment in your submitted code).

All work must be submitted via Moodle (see "Assignments" section for submission). Work submitted via other means will not be accepted unless you have prior arrangements with the Head Demonstrator (Behnam Faghih). All work MUST be submitted by the due-date deadline. Late submissions will not be accepted.

*The assignment submission is a zip file named "**assignment-03-xxxxxxxxx.zip**" (where "xxxxxxxxx" is your student id) containing a solution file named "**assignment-03.html**" together with any other resources used in the assignment solution. External CSS and Javascript files should be named "**assignment-03.css**" and "**assignment-03.js**", respectively. Please ensure that all external files use relative directory referencing, rather than hard-coding the files' location.*