

CS230[A] – Web Information Processing (2018-19:Semester 2)

Dashboard / CS230[A] (18-19:S2) / Labs / Lab 3 / Assignment 1

Lab 3 / Assignment 1

This is an open-book, graded assignment. You can use any resource available on the web, or in a textbook to help with the assignment. **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.

You should create a profile at <http://jsfiddle.net> to work on this assignment. You should submit the shared application link as part of the Moodle submission at the end of the class. Be sure that you share this link correctly. This assignment is to be completed using HTML, CSS and JavaScript only. jQuery should **not** be used in this assignment.

Before submitting your assignment students should check that their solution works in Chrome and Firefox.

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 (Shane McGarry). All work MUST be submitted by Friday, 1 March 2019 at 17:00. Late submissions will not be accepted. The assignment submission is the link to the code in JSFiddle.

1. You are required to develop an HTML/CSS/JS application that implements the following functionality. Use HTML/CSS to present hypothetical student assignment grades using an online styled table.
 1. 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 column that contains the arithmetic average of the five assignment grades in the preceding columns.
 2. The table should contain 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).
 3. Only table cells containing actual grades should be "content editable", and allow manual entry of the grades (every cell in all columns is editable except for the final aggregate column).
 4. The table style should utilise a sans-serif font for all data presentation. Apart from cells denoting column headings, all cells should have a white background with black text.
2. You should provide a styled button, centred below the table, with a background colour other than white or black, and should have text that says "Calculate Final Grades", that invokes a JavaScript function that calculates the average grade for each student, and writes the arithmetic average (mean) to the "Final Grade" column.
 1. 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.
 2. Final Grades that have a value below 40, should be styled so that the grade font is presented using a white colour on a red background.
3. **Extra Credit:** If you **complete** this assignment before the deadline, then try the following (if you wish). These can be completed in any order:
 1. Use JavaScript to automatically re-calculate all Final Grade averages whenever a table cell is updated by a user (*1 point*).
 2. Use JavaScript to count the total number of assignments that have not yet been submitted, and present in styled format, beneath the table. Furthermore, change the background of cells containing "-" (unsubmitted assignment) to yellow. These functionalities may be an automatic calculation, or achieved using a styled button (*1 point*).
 3. Use JavaScript to automatically validate cell data when manually updated by a user. Cells containing erroneous data should default to being "unsubmitted", i.e. contain a "-" (*1 point*).
 4. Use JavaScript to write a CSV representation of the table to a styled TEXTAREA below the table (this facilitates copying and pasting of the table data into another application, saving, etc.). This may be an automatic calculation, or achieved using a styled button (*2 points*).