CO323 – Databases and the Web
Assessment 1 – Integrating HTML, CSS and JavaScript

Set on:	Monday, Week 17 (10/02/2020)
Due in:	Thursday, Week 19 (27/02/2020) at 23:55
Submission (individual):	via Moodle

Question 1 – 35 Marks

Create a web page that provides an input form for registering users to a gym membership. The form should ask for the user's name, age, height (via two drop downs for feet and inches between $4' - 6'11''$), and start date (via a date input). Provide two buttons, a "Reset" button, and a "Display" button with the following behaviour:

- Create an empty table below the form, where each column of the table corresponds to one of the input fields (i.e. you should create header cells for the first row of the table).
- When "Display" is clicked, add a new row to the table and clear the form – each cell of the row should correspond to one of the input fields from the form (for height, merge the input from the two drop downs into one cell). Each set of data entered by the user (via clicking "Display") should add a new row to the table. Rows of the table should have alternating background colours.
- Before adding a new row, input from the user should be validated. Age should be between 18–60 and the start date should be in 2020.
- "Reset" button should clear the form.
- Each added table row should have a cell containing a "Delete" button, that when clicked, removes the row from the table.
- You should also include a piece of text on the caption of the table that provides a counter of the number of data rows in the table. Whenever a new row is created, this number should be incremented. Whenever a row is deleted, this number should be decremented.

Question 2 – 20 Marks

Create a web page that generates a series a of small boxes, where the number of boxes is given by the user via a text input box. Incorporate the following behaviour:

- Set the background of each box to a random colour (you can use the "rgb" format of CSS colours).
- Number the boxes in consecutive integers, i.e. 1, 2, . . . , n.

- Set the margin-left of each box to be double the previous boxs margin, where the first box has *margin-left* 5px.
- Clear the old boxes when a new input is given by the user.

For reference, your code should produce something similar to the below figure for five boxes.

Question 3 – 45 Marks

The aim of this question is to build the *Tic-Tac-Toe* game (otherwise known as *Noughts-and-Crosses*). The following list describes the expected behaviour:

- Create the "game board" which comprises a 3x3 grid of boxes. Consider that later you will need to check for "winning conditions", e.g. a line of three Xs or Os. Think about how to structure the code so that this task is easier in the future. You will want an easy way to access the value of a box based on its row and column position. Therefore, you may want to generate the game board dynamically (e.g. using *createElement*). However, a static game board (i.e. not generated via JS) is acceptable.
- Provide the following behaviour: when a box is clicked, it is filled with either an X or O. When another box is clicked, it should then be filled with the alternate marker for the other player.

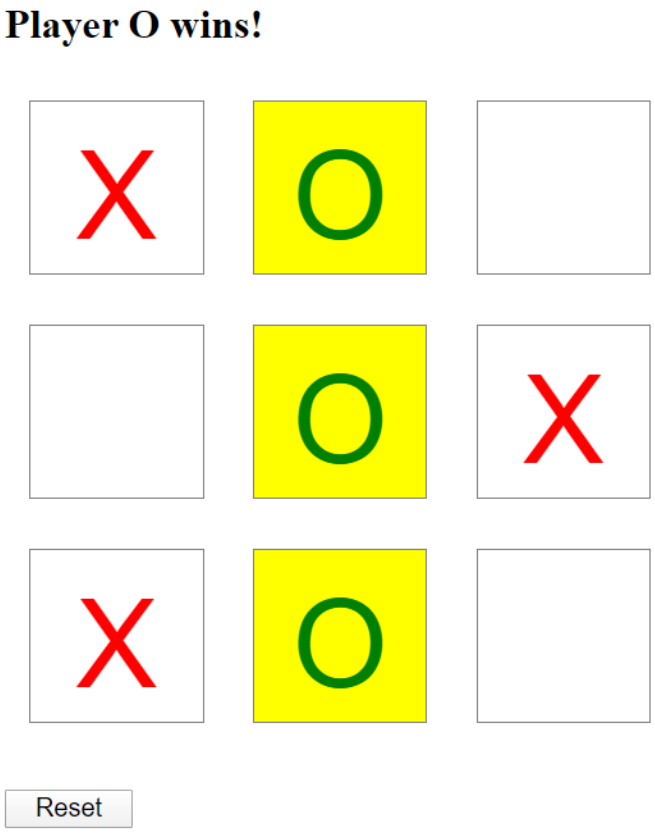
  (Hint: You will need a piece of state recording the current player. Each click event updates a particular box and switches the player.)

  Make sure that a box cannot be changed from an X or O once it has been clicked.
- After each user update of a box, provide a function that checks whether the winning condition has been met and provides a message if so. Consider the various possible winning conditions and how to write a function that concisely checks these.

  No further player moves should be allowed after the winning condition is reached.

  If there are no more moves and a player has not won, display an appropriate message.
- Add a "Reset" button to clear the game board and restart the game.
- Add some CSS styling to distinguish between Xs and Os, place Xs and Os in the centre of the boxes with big enough fonts, and clearly show the winning boxes.
- Pay attention to readability of your code: add comments, use intuitive variable names, use proper indentation, and make use of functions.

For reference, the end of the game should look similar to the below figure.



Submitting Your Work

Please create separate HTML files with .html extension, named "questionX.html", where X is the number of the question. If you happen to create any subsidiary CSS or JavaScript files then these should be named accordingly for the question, i.e. "questionX.css", where X is the question number.

Please submit a single ZIP file containing all your work.

You will gain marks for partial solutions, so please have a go at every question. Submit your answer files via the link on the module's Moodle page before the deadline above. You may re-submit your work as often as you like, but only the last submission will be kept and marked.

Late Submission and Plagiarism

Late or non submission of coursework

The penalty for late or non submission of coursework is normally that a mark of zero is awarded for the missing piece of work and the final mark for the module is calculated accordingly.

Plagiarism and Duplication of Material

Senate has agreed the following definition of plagiarism: "Plagiarism is the act of repeating the ideas or discoveries of another as one's own. To copy sentences, phrases or even striking expressions without acknowledgment in a manner that may deceive the reader as to the source is plagiarism; to paraphrase in a manner that may deceive the reader is likewise plagiarism. Where such copying or close paraphrase has occurred the mere mention of the source in a bibliography will not be deemed sufficient acknowledgment; in each such instance it must be referred specifically to its source. Verbatim quotations must be directly acknowledged either in inverted commas or by indenting." The work you submit must be your own, except where its original author is clearly referenced. We reserve the right to run checks on all submitted work in an effort to identify possible plagiarism, and take disciplinary action against anyone found to have committed plagiarism. When you use other peoples' material, you must clearly indicate the source of the material using the Harvard style (see https://www.kent.ac.uk/ai/styleguides.html).

In addition, substantial amounts of verbatim or near verbatim cut-and-paste from web-based sources, course material and other resources will not be considered as evidence of your own understanding of the topics being examined.

The School publishes an on-line Plagiarism and Collaboration Frequently Asked Questions (FAQ) which is available at: https://www.cs.kent.ac.uk/students/plagiarism-faq.html.

Work may be submitted to Turnitin for the identification of possible plagiarism. You can find out more about Turnitin at the following page: https://www.kent.ac.uk/ai/using-turnitin.html.