CE212 ASSIGNMENT 1 2020

Set by: Mike Sanderson

Credit: 10% of total module mark

Deadline: 11.59.59, Wednesday 12 February

Introduction

The assignment comprises three independent exercises. They are not equally weighted.

You should refer to sections 5 and 7 of the Undergraduate Students' Handbook for details of the University policy regarding late submission and plagiarism; the work handed in must be entirely your own.

It is expected that marking of the assignments will be completed by the beginning of week 24.

You may, if you wish, use JQuery, but are not required to do so.

Submission and Marking Criteria

Copies of all code (including CSS, JavaScript and HTML files) must be submitted to FASER by the deadline in a single `.zip` or `.7z` file. (Marks may be deducted for files submitted in any other format.)

Other than where specific instructions have been given about coding (e.g. the use of an array of objects in exercise 2) the only criteria to be used for marking of this assignment are that the web pages have the correct functionality, and, where appropriate, the required style.

CE212 ASSIGNMENT 1 2020

Exercise 1 – Dynamic Data Table [35%]

The requirement for this exercise is to use JavaScript (together with CSS and HTML) to produce a dynamic table of data listing conversion between distances in miles and kilometres. The web page must obtain from the user the range (in miles or kilometres) of the distances to be displayed and produce a table showing each distance in the range alongside its equivalent in the other unit, which must be displayed to ***exactly two decimal places***. The user must be allowed to select whether the conversion is from miles to kilometres or from kilometres to miles. The JavaScript should be invoked by a button-click.

Table header cells must have a text colour that differ from data calls; alternative rows must have different background colours.

Text box input fields should be used for the entry values for the start and end of the range. Validation must be performed – the first value must be a positive integer and the other (if supplied) must be a non-negative integer. An alert or error message should be displayed if the validation fails. The user may enter the two values in either order so the program must be capable be displayed the table in both ascending order and descending order as requested. If only the first value is supplied you should use a default table size of 10 rows in ascending order.

To perform the conversion you should assume that 1 mile is equal to 1.609 kilometres.

Exercise 2 – Marking Form [30%]

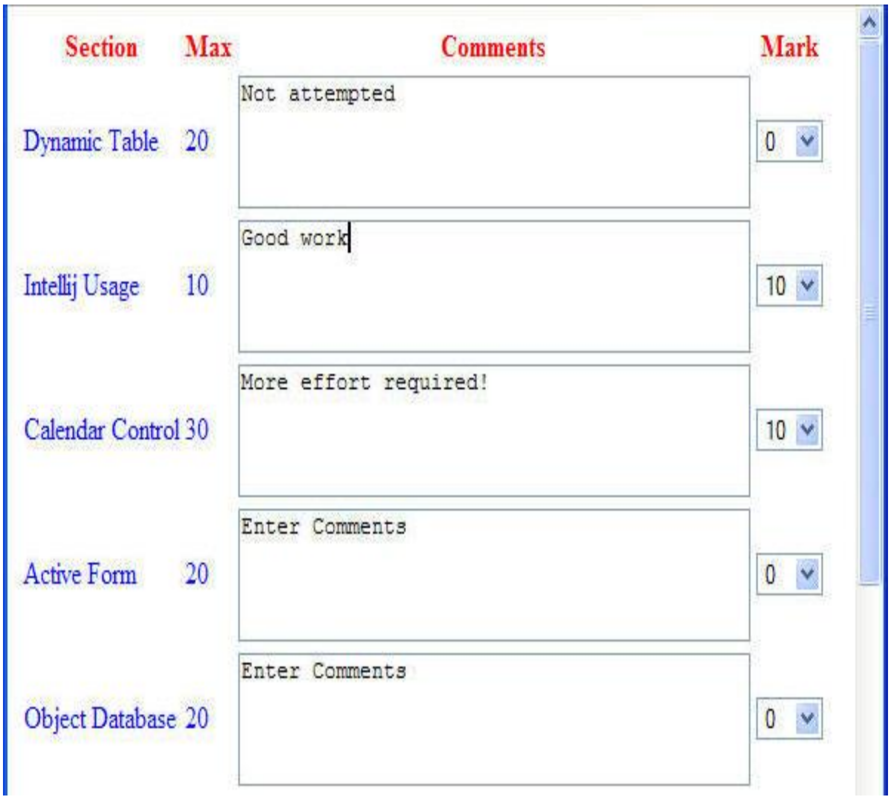
This exercise requires the creation of a marking form for an assignment.

Your solution should have components as shown below and also display the total mark, which should be updated as marks are entered.

The form must be created using JavaScript. The details of sections (i.e. their names and maximum marks) must be stored in an array of objects (which must not themselves be arrays) so that it is easy to change these details by changing the initialisation of the array in one place in the JavaScript file; all of the HTML-generation code must obtain the necessary data from the array. You should ***not*** assume that the array will always contain details of exactly five sections or that the maximum marks will always total 100.

Each drop-down menu should contain all of the integers from 0 to the maximum mark for the section.

The form should have a Clear button, which when clicked, should clear all of the comments and reset all marks to 0.



Section	Max	Comments	Mark
Dynamic Table	20	Not attempted	0
Intellij Usage	10	Good work	10
Calendar Control	30	More effort required!	10
Active Form	20	Enter Comments	0
Object Database	20	Enter Comments	0

CE212 ASSIGNMENT 1 2020

Exercise 2 – Word Count [35%]

The requirement for this exercise is to write JavaScript code to extract the text from an external HTML page and count the number of occurrences of each word in the text. The results should be presented in a table displaying each word and its number of occurrences. To gain more than 25 of the 35 available marks the output should be sorted, with the user being given the option of whether they should be displayed in alphabetical order or in order of number of occurrences.

You may, if you wish, assume that the web page is on the same file system as the web page written for the exercise, to allow the technique in lab 3 to be used to extract the contents of the page.

You may also assume that the web page comprises correctly-formed XHTML. A word is to be regarded as any sequence of non-white-space characters containing at least one letter within a text node within the body of the document so in the following simple web page the words are `Hello`, `World!`, `Click`, `here`, and `Goodbye` (with `World!` having two occurrences). Words are to be regarded as case-insensitive so `World!` and `world!` should be viewed as occurrences of the same word. (Words within tags, such as `body` and `href` in the example below, should not be included in the count.)

```
<html>
<body>
<p> Hello World! <br/>
    <a href = ”link.html”> Click <b> here </b> </a> <br/>
    Goodbye world!
</p>
</body>
</html>
```