Swinburne University of Technology
Faculty of Science, Engineering and Technologies

COS10011 / COS60004 Creating Web Applications

Individual Assignment: Part 2
Semester 2, 2020
Develop an Interactive Website

Important Dates:

Due Date ESP	**8 am, Day of your tutorial, Week 8 (28 Sept – 2 Oct)** (Late submission penalty: 10% of total available marks per day)
Demonstration	Your allocated tutorial: Week 8

Contribution to Final Assessment: 24%

You must meet the Essential Requirements of this assignment to be eligible to submit Part 3 of the assignment. **See the making guide for the *essential requirements* for this part of the assignment.**

Note: Do *not* use JavaScript libraries or frameworks (e.g. jQuery, Angular) in the main part of this assignment. You may create an additional alternative implementation using a library/framework as an enhancement (see enhancements section below).

Note: The code that is assessed in your demonstration *must be identical* to the code you submit to ESP. This will be checked before the demonstration.

Prerequisite

If you failed to meet the Essential Requirements of Part 1, ***before*** this part of the assignment is submitted and marked, you need to demonstrate that you have fixed problems in your first part. Note that these fixes will **not** alter the mark you received for Part 1. It is advisable to get these fixes complete and signed off *well before* you hand in this assignment. The tutor will check the fixes and sign-off that they have been completed. Your tutor will be happy to advise you during labs or during consultation sessions if you need assistance fixing Assignment 1.

How to get your fixes signed off:

1. Arrange a time with your tutor to check your work during your allocated tute or during a consultation time.
2. Bring a copy of the ESP assessment printout from Part 1
3. Your tutor will check that your fixes to Part 1 address the issues identified on the mark sheet (fixes will not be required to the features html page).
4. If the fixes are successful, your tutor will record this and you will be eligible to have this assignment assessed. If there are issues that have not been fixed, your tutor will inform you of this and you will have *a further chance* to fix the assignment.

Purpose of the assignment

In this assignment you will further enhance the website you developed in Assignment Part 1 by using JavaScript to mark the quiz you created. You will:

- Use JavaScript to check data entered into HTML forms and provide user feed back

- Use client-side storage to transfer data between pages.

As in Part 1, there will be an opportunity to enhance your website beyond the basic requirements.

The `"use strict";` directive should be included at the start of all your JavaScript files.

Web Site Description

Check the quiz answers

In Part 1 of the assignment you created an HTML form **`quiz.html`** that consisted of at least five questions related to your topic. In this part of the assignment we will use JavaScript to mark those questions. JavaScript should be in a file called **`quiz.js`** located in a **`scripts`** folder.

Create a marking scheme for your questions so that each question is allocated one or more marks.

When the user clicks the Submit button on the form use Javascript to:

1. Check they have a selected an answer for any questions where a 'required' attribute was not set in HTML (e.g. check boxes)
2. compare their answers with the correct answers you have defined and calculate their score.

If the user gets zero for the quiz they should not be able to submit their answers. Provide the user with some feedback if this is the case.

Once a positive score is calculated, display the results of the attempt on a webpage called **`result.html`**. Use HTML5 local storage to do transfer the information between pages.

This web page will display:

- The user name and id.
- The score achieved for this attempt.
- The number of attempts the user has made doing the quiz using their browser.
- A hyperlink that allows them to have another attempt at the quiz (only if they have had less than 3 attempts).
- The maximum number of attempts a user (identified by their id) can make is three. You need to implement a way to prevent the user doing the quiz more than three times.

Hint: change the form you created in Part 1 of the Assignment to set the **`method`** attribute to **`result.html`**. While this file is just a static HTML page that does not do any processing of the form data, the web page will be returned from the server to the browser where you can initialize it with the local information you have stored.
We will further enhance this by adding server-side processing in Part 3 of the Assignment.

***Note**: There should be **no** JavaScript embedded in your HTML files. This precludes both event registration (e.g. `<form onsubmit="return validate()" . . .`) and function definitions, in the HTML.*

CSS

All pages should be styled appropriately using CSS as in Part 1, and should be valid CSS3.

If you wish to make alterations to the HTML and CSS in your Assignment Part 1 that is OK (but you must keep your assigned Web topic).
Remember: You need to implement your website in standard HTML5 that is also well-formed XML.

Enhancements using JavaScript

You should complete the above requirements before attempting any enhancements.

As with Part 1 you have an opportunity to implement enhancements to your Web site using techniques not covered in the tutorials. Each enhancement must be described on a page called **`enhancements2.html`**. The entries on this page should:

- briefly describe the interaction required to trigger the event ***and*** what a programmer has to do to implement the feature.
- provide a hyperlink to the page where the enhancement is implemented in your Web site.
- reference any 3rd party contribution to the enhancement

It is a good idea to discuss your proposed enhancements with your tutor before you implement them.

The JavaScript enhancements themselves should be in a separate **`enhancements.js`** file. Make sure there are adequate comments to explain the enhancement (including its source if applicable).
Examples of JavaScript and other enhancements you might make include (but are not limited to):

- Have your quiz questions written in JavaScript and dynamically display the questions on the quiz page (you can extend this enhancement in Part 3).
- Have multiple versions of quiz questions that are randomly displayed.
- Implement a timer so that the user only has a limited time to complete the quiz .
- *Re-implement* your JavaScript using a library such as jQuery or a framework like AngularJS. Add some enhancements the library/framework provides. **No** library code should be included in your **`quiz.js`** file. This alternative implementation should be in the file **`enhancements.js`** file. Explain the difference in approach using the library and using plain JavaScript.

Smaller enhancements (less than 5 marks) might include:

- Use the JavaScript methods `querySelector()` that take a CSS selector as an argument to manipulate the web page in response to user action.
- Create an extra client side JavaScript dynamic effect: e.g. Slideshow, random image displayed onload, etc. The code and structure of this is open, but must be documented and explained as clearly as possible.
- Use JavaScript to change the Menu display, to reflect the current page being viewed.
- …

Any enhancements that are not listed and linked on the page **`enhancements2.html`** ***and*** implemented in **`enhancements.js`** will not be assessed.

Up to 5 marks will be allocated to each enhancement. A maximum of 2 **enhancements will be assessed.**

Web Site Folder Structure and Deployment Requirements

Your website folder structure should follow a similar structure as Assignment 1.
All files should be under a folder `/assign2`. JavaScript should sit in an `assign2/scripts` folder.

```
assign2/              You must have this folder – case sensitive!
  index.html
  topic.html
  quiz.html
  result.html
  enhancements.html
  enhancements2.html
  ...other html pages

  scripts/            Folder for your JavaScript
  images/             Folder for images for your page content
  styles/             Folder for style.css other css files
  styles/images/      Folder for images referred to by your css files e.g. background
```

Notes:

- HTML files should only be in the base "assign2/" folder – not anywhere else.
- All links to your files (JavaScript, CSS or images) should be ***relative***. ***Do not use absolute links***, as these links will be broken when files are transferred for marking. No marks will be allocated if links are broken.

Assignment Submission

An electronic copy of your assignment should be submitted through ESP at https://esp.swin.edu.au on or before the due date.

- Make sure all your files are in the correct folders and compress your root folder with all your sub-folders with HTML, CSS, JavaScript, and image files into a zip file named "assign2.zip". Submit this to ESP. When the zip file is decompressed, the entire Web site should be able to be run from index.html without needing to move any files.
  (Hint: Check the zip file to ensure you have included everything.)
- You can submit more than once through ESP (Make sure you select the correct unit code!).
- Note that all deliverables must be submitted as softcopy. There is no need to submit an assignment cover sheet as ESP generates a receipt upon successful submission.

Mark Sheet

Marked by: ……

Declaration:
You should declare: I hereby confirm that the assignment to be demonstrated is identical to that I submitted to ESP.

Student number ……. Student name ……..

Signature ……. Date ……

Allocated topic ……..

Tutorial Day …… Tutorial Time …….. Tutor Name …….

Marker to Complete	
Prerequisite - Essential requirement errors in Assignment 1 fixed	☐
Mercury date file check ☐	Days late penalty if applicable (10%/day) ……..

Essential Requirements

Tick box ☑ if requirement met

To meet the essential requirement s
- **`quiz.html`** and **`result.html`** must be valid HTML5 running on mercury ☐
- using JavaScript a score is accurately calculated from the quiz submission ☐
- on quiz submission **`result.html`** displayed ☐
- name and score displayed on **`result.html`** ☐

All essential requirements met	**Y/N**

If your assignment fails to meet these essential requirements you will need to make it compliant before submitting Assignment Part 3.

Total marks = 50

Requirements	**Mark**

• All questions (e.g. check-boxes) have answers (2).
• The score correctly calculated using JavaScript (2 for each question counted = **10**).
• The following information *saved* on submit from **`quiz.html`** in local storage and *displayed* on **`result.html`.**
  ○ The user name and id. (**5**)
  ○ The score achieved for this attempt. (**5**)
  ○ The number of attempts the user has had at doing the quiz. (**5**)
• Maximum 3 attempts a user can have at doing the quiz (**5**)
• Quiz result not submitted if zero (**4**)
• A hyperlink on result.html to allow user to have another attempt at the quiz if attempts less than 3 (**4**)
	/40

Subtotal	**/40**

A maximum of 2 enhancements will be assessed if listed and linked from enhancements2.html. Up to 5 marks are available per feature. Poorly implemented or trivial enhancements may receive less or zero marks.

Enhancements **Name**	**Described**	**Linked to where implemented on your Web site**	**Source (if applicable)**	**Mark**
	Y/N	Y/N	Y/N/na	/5
	Y/N	Y/N	Y/N/na	/5
Sub-total				**/10**

*Deductions may be made during the demonstration or during code inspection **after** the demonstration.*

Requirement	**Deduction if not met**	**Deduct**
HTML5		
- Valid HTML5	-2 per error	
- Well-formed XML	-4	
- Meta-data follows in-house standard	-2	
- HTML has no Style information embedded	- 1 per instance	
- HTML form elements follow in-house standard	-2	
- No deprecated elements/attributes used	-2	
JavaScript		
- All JavaScript is in an external file	- 2 per instance	
- "use strict" directive present	**-5**	
- No 3rd party libraries/frameworks used in quiz.js	**up to -10**	
- Header comments as per in-house standard	-2	
- Line comments as appropriate	-2	
Web site		
- All third party content acknowledged properly*	up to -50	
- Directory Structure as defined above	-4	
Total Deductions		

* Note: Failure to acknowledge third party code or content *at all* is plagiarism and may result in zero marks for this assessment or other penalties in accord with Swinburne policy.

A final assignment mark will *not* be provided during the demonstration. All code is inspected after the demonstration by your tutor before a final mark is allocated.

Comments:

……