Swinburne University of Technology
Faculty of Science, Engineering and Technology

COS10011/ COS60004 Creating Web Applications

Assignment Part 1, Semester 1, 2018

Develop a simple static Web site

Important Dates:

Due Date ESP	**Week 5 – the day of your tutorial: 26 – 30 March 2018, 8am** (Late submission penalty 10% of total available marks per day)
Demonstration	Your tutorial: Week 5

Contribution to Final Assessment: 12%

Hurdle Note: You must meet the Essential Requirements of this assignment to be eligible to submit Assignment 2 (and 3).

This is an Individual Assignment. *All work must be your own.* Submissions are *automatically checked* for similarities. Unexplained/acknowledge similarities may constitute plagiarism. Carefully read the section on plagiarism in the Unit Outline before you proceed (including the section forbidding sharing your work with others).

Purpose of the assignment

This individual assignment will familiarise you with the techniques and skills involved in designing and creating static webpages, utilising validated HTML and CSS created with a standard text editor. You will deploy these Web pages on a Unix / Apache server. This should be done in a way that keeps HTML content and CSS presentation separate, as discussed in the lectures. ***No JavaScript is to be used in this assignment.*** It will be included in your next assignment.

In this assignment you will develop a Web site that describes a Web-related Technology. Each student will be assigned a *different topic.*
Your topic is available at https://mercury.swin.edu.au/it000000/assignment/topic.html
Your Web-site will have:

- An introduction to your topic (index.html)
- A more detailed description of the technology (topic.html)
- A quiz page related to your topic (quiz.html)
- A page which lists any enhancements you have made (enhancements.html)

You must call these files **exactly** by these names, otherwise the marking program will not know they exist!

You will also include

- A CSS file that styles your website (style.css).

The ***essential requirements*** for this assignment are ***listed in the marking guide***. In general the web pages must:

- have relevant content
- include the HTML markup specified in the marking guide
- validate to HTML5 without errors
- have a <head> with Title, Meta tags as specified in template (including author)
- be styled by a validated CSS3 file
- be linked to each other via a menu
- be deployed on Mercury.

Detailed marking criteria are contained on the last two pages of this document.

Content and presentation of Web Pages

All web pages in your website should have a consistent layout and navigation. Where “in-house” **templates** have been defined in this unit (e.g. for meta-data; tables; etc.) these should be followed. These include accessibility alternatives.

The HTML in your Web pages ***must*** validate against the W3C HTML5 validator (http://validator.w3.org/nu) . They ***should*** also be well-formed XML.

Hint: HTML5 validators do not necessarily check that the markup is well-formed XML. Check of your Web pages are well-formed xml by saving a copy of your **served** pages locally with an .xml extension (for example a locally saved *copy* of myfile.html would be renamed myfile.xml). If well-formed no errors will show when the xml file is loaded into a browser.

Pages should ***not contain any deprecated elements/attributes*** (e.g. <i> , <b>). *Do not* use iframe elements in your assignment.

Note: Generic structural elements like div *or* span *should only be used where there is no more meaningful HTML5 element (e.g.* section*) that is appropriate.*

1. Introductory home page *(index.html)*

This page should contain:

- An appropriate title.
- A *background* graphic (use CSS to do this).
- A menu that links to the other pages on your Web site. This menu should appear on every page of your website.
- A header containing appropriate content. This header should appear on every page of your website.
- A footer that includes an email hyperlink to your student email address. This footer should appear on every page of your website.

2. Topic Description page(s) *(topic.html)*

You should write at least 200 words on your allocated topic and the content should be appropriately marked-up with headings, paragraphs, sections, subsections etc.

The page must contain:

- Hierarchically structured headings of at least 2 levels. (e.g. make sure you H2 is below an H1)
- More than one <section>
- An <aside> with appropriate content
- At least one appropriate image related to your topic. This image should be less 100k so it does take too long to load.
- A table containing some data related to your topic.
- At least one ordered list
- At least one unordered list

Your content should briefly and concisely explain such as:

- What is the technology? Its purpose / function? Major points / features ?
- Who developed it? When? Why?
- What groups, if any, are responsible for managing it?
- Explain its growth or decline. Predict the future for the technology.
- What are related technologies? Compare / contrast with other technologies.

If you decide to have more than one topic page call them *topic1.html*, *topic2.html*, etc..

Note: You are not expected to create the "definitive reference" on your topic, but your information should be accurate. Incorrect information will reduce your mark.

Sources / References:

- Good resources for starting your research include Wikipedia, Google and online databases in the library. (You should use at least 3 sources of information. Do not just use Wikipedia.)
- Use primary sources for information, but then write the content in your own words.
- Please do NOT copy your text from other websites. You can quote / cite key text from another site, and there are appropriate ways to do this, eg. mark-up using a blockquote, inline quote, citation.
- Keep track of **all** your sources/references and cite them as footnotes.

3. Quiz page *(quiz.html)*

This page consists of a form where the user can enter:

- their first name
- their last name
- student number
- answers to at least 5 questions *related to your topic*.

The questions should be use five different input types:

a. a text input question
b. a multiple choice question with one correct answer (radio button group)
c. a multiple choice question with multiple correct answers (check box group)
d. a drop list with a single correct answer.
e. an input type of your choice other than the above (e.g. number, range, text area, image map, etc.)

Fieldsets and legends should be used appropriately to group inputs into questions.

Labels and inputs should be linked with a **for** attribute.

Before quiz answers are submitted **HTML5 data validation** should be used to check the following:

- Text and radio input questions must be answered
- Name and student id fields are not empty.
- The first and last name data should be checked to ensure it only consists of alpha characters, hyphens or spaces. A maximum of 25 character should be able to be entered.
- The student number is either 7 or 10 digits.

Data Submission of Quiz answer to the Server

For this assignment all forms should have a Submit input. When the submit button is clicked the name-values from the associated form should be sent to the server using the **post** http **method**. The server **action** address is https://mercury.swin.edu.au/it000000/formtest.php. The server will then just echo back the name value pairs to the client. While nothing will be stored on the server in this part of the assignment (we will do this in Part 3) this will allow the form to be tested.

4. CSS Requirements

No style markup should be included in your HTML file.

The pages in your website must be styled with CSS and have a consistent ‘look and feel’, particularly common elements such as menus, headers and footers. While the emphasis is this assignment is on the appropriate application of techniques rather than graphic design, your pages should follow basic usability / accessibility principles, e.g. distinguishable foreground and background colours, and font readability, etc.

You are to create your own to implement your design. For this assignment you should create a single ***external*** CSS stylesheet that styles the common elements on *all* your Web pages. This file should be named ***style.css*** in an appropriate folder

1. **CSS should be commented at the beginning of the CSS file to identify author and purpose, and individual line comments should be used as necessary to explain particular styles and explain where they are applied**.
2. ***All*** the following CSS Selectors should be used *appropriately* at some point in your assignment:
   - element, #id, .class, grouping, contextual
   - pseudo class or element
3. Provide appropriate formatting to your menu with a background colour.
4. The following specific CSS rules should be demonstrated on your **index.html** page:
   - display a background graphic.
   - the footer text should be in a small font and centred in the footer.
5. The following specific CSS rules should be demonstrated on your **topic.html** page:
   - <h1> elements should have their font variant, size and family etc. set using the short-hand **font** property.
   - The table should have one a background colour for the headings and another background for the data cells
   - The <aside> should be 25% of the width of page and float to the right.
   - The <aside> should have a coloured border with an appropriate margin and padding.
   - The footer should cover the full width of the page.
6. All pages should have a fluid layout (the page should “Reflow” on page resize).

Other CSS selectors and properties should also be used as necessary and appropriate for the presentation.

Hint: CSS validators will validate against a particular version of CSS e.g. CSS2.1 or 3. This assignment should be valid CSS3. Make sure that you are checking your CSS using the correct version of the validator. For example, if you include CSS3 markup and validate as CSS2.1 it will show errors. (Best to pre-set the version in the Web Developer tools).

Do not include any proprietary CSS mark-up, such as –moz- or –webkit- etc.

5. Enhancements to the Specified Requirements

Note: Make sure you get all the basics working first before you attempt any enhancements. See the marking Guide below.

The technologies for developing Web applications are rapidly changing. One of the key skills you will need is finding out about these techniques (from the Web) and applying them. This assessment gives you need an opportunity to demonstrate your ability to implement features/techniques that go beyond the specified requirements above. This is an opportunity to demonstrate your ability to discover techniques from a range of sources and apply them in a standards compliant manner.

These enhancements need to be implemented within the Web pages (index.html, topic.html, quiz.html). The extra features needs to enhance your website in a relevant way.

On a separate Web page called *enhancements.html* **list** and **describe** each enhancement you have made and how you have significantly extended the basic HTML and CSS beyond what is covered in the Tutorials. Hyperlink from this list to where the feature is implemented in your website. If it is a CSS feature, hyperlink to an example of the html that is selected by the CSS rule. For each enhancement feature briefly explain:

- ☑ how it goes beyond the basic requirements of the assignment
- ☑ what code is needed to implement the feature
- ☑ if you have sourced your technique from a third party the source of this technique (e.g. URL) ***must be cited***.
- ☑ **a hyperlink *to where you have applied that extension in your website*** (this is needed so the tutor can quickly assess your enhancements during the demonstration).
- ☑ All enhancements must be able to run on the version of Firefox in the labs. Make sure you check this.

A *maximum of 2 enhancements* will be assessed (*up to* 10 marks each). **Examples** of HTML/CSS enhancements you might make that will contribute a high distinction mark include:

- *Effective, appropriate and innovative* use of a **number** of distinct HTML elements not covered in tutorials (e.g. Image maps, Canvas, etc) used in a way that improves the user experience of the website.
- A **number** of additional CSS properties or selectors (e.g. support for interactivity) not covered in the tutorials. For example the use of a range CSS3 pseudo-elements and classes, child or siblings combinators, attribute selectors, etc. (e.g. use the CSS3 :target selector to help us see where you have applied your enhancements.)
- Implement Responsive Design with additional CSS that presents your website specifically for mobile phone / tablet sized displays.
- ***In addition*** to you standard CSS, create another CSS files that *re-implements* and *extends* the style with a library/preprocessor such a Bootstrap, LESS or Sass. Demonstrate and document a number of cool features that can be implemented using the library.

Discuss your proposed enhancements with you tutor before you implement them. The number of marks you receive for an enhancement will be at the ***sole*** discretion of your tutor/marker. As a guide if the enhancement has only taken a couple of lines of code it is likely to be trivial.

- Be relevant to / enhance the content of the website
- Be well described (as explained above)
- Be non-trivial.
- Be significantly *different* from other features you have implemented.

Note: Do not include JavaScript in this part of the assignment. This will be covered in the next part.

Web Site Folder Structure and Deployment Requirements

The directory structure of your website is described below. You can create additional HTML files for your content (depending on what your content requires), but the following is needed:

```
assign1/             You must have this folder – case sensitive!
  index.html
  topic.html
  quiz.html
  enhancements.html
  ...other html pages
  images/              Folder for images for your page content
  styles/            Folder for style.css other css image files
    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 “assign1/” folder – not anywhere else.
- **All** images used for the **content** should be stored in the “assign1/images/” folder.
- **All** images used for the style should be stored in the “assign1/styles/images/” folder.
- There should be a “style.css” file in the “assign1/styles/” folder.
- All links to your files (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.

Note: DO NOT INCLUDE VIDEO OR OTHER LARGE (>5MB) MEDIA FILES IN YOUR ESP SUBMISSION.

Make sure you thoroughly test your website deployment on the mercury server using a Firefox browser.

Assignment Submission

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

- Make sure all you files are in the correct folders and compress your root folder with all your sub-folders with HTML, CSS, and images into a zip file named “assign1.zip”. Submit this to ESP. When the zip file is decompressed, the entire website should be able to be run from index.html without needing to move any files.
- You can submit more than once through ESP.
- 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.

Make sure you complete your ESP submission process.
You should get a PDF receipt if you have submitted successfully.

MAKE SURE YOUR ENTER **THE CORRECT UNIT CODE** WHEN YOU SUBMIT YOUR ASIGNMENT TO ESP.

Submitting to the wrong unit will not be grounds for granting an extension.

Assignment Demonstration Procedure

Note: When the tutor marks your demonstration they will not (just) be examining how the Web site works and looks but primarily how it is **written.** You need to have your source code displayed during the demonstration so you can discuss it with your assessor.

1. Make sure you attend your allocated lab. You will demonstrate your assignment to a tutor in your allocated Tutorial in week 5, 8 or 12 for Parts 1, 2, and 3 respectively. *You must attend this session to receive a mark for this assignment.* If you cannot attend your allocated tutorial due to illness you must provide a copy of the medical certificate to the convenor.

2. What you demonstrate in the tutorial ***must*** be the same as you submitted to ESP. The tutor will check the time stamps on your files in Mercury ensure this.

3. **Before** your demonstration starts
   a. Fill in and sign the Declaration on the Marking Sheet your tutor will give you.
   b. Make sure your web site is running **on the server** Mercury. (Your marker will check the URL).
   c. Load the Web site into **Firefox**. All demonstrations will be done on Firefox.
   d. In separate windows display the **source code** for ***all*** your HTML and CSS files.
   e. **Validate** ***all*** your HTML and CSS files in separate tabs in the browser so they can be quickly checked by your marker. (It might be helpful to load Firefox with the appropriate add-in for validating the HTML and CSS.)
   f. Ensure XML tools for Notepad++ is installed so your code can be quickly checked for XML compliance.
      OR
      Using WinSCP (or similar), copy the HTML files from Mercury to your local machine. Change the extension of the files to .xml and load them into your browser for viewing.
      (Do not save the served HTML files from browser using File | Save As …)

   g. Display a copy of your ESP receipt.

4. **During** the demonstration load *index.html* and your tutor will get you to run through the functionality of the Web site and check the validations. As you demonstrate your Web site your tutor ***will ask you to show the source code*** in your browser and explain how you have implemented various aspects of it.

5. **After** the demonstration your tutor will mark your source code. Your code will be checked in detail to ensure that it is your own work and that it meets standard (e.g. no style defined in the HTML). Results are released by email from ESP.

Mark Sheet – Assessed by demonstration in your tutorial

Declaration: **Fill this in before you start**

I hereby confirm that the assignment to be demonstrated is identical to that I submitted to

Student number ………………. Student name ……………………….

Signature ……………………. Date ……………..

Topic ………………………………………………………..

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

Marker check: ESP with receipt sighted □ *File date check:* □

Essential Requirements Place ☑ or ☒ in box	**Y/N**
index.html - validates to HTML5 □- <head> with appropriate meta tags incl. title and author □ - menu that links □ email link in footer □	
topic.html - validates to HTML5 □ - meaningful content □ - at least ~200 words □ -heading elements □ list □ table □ graphic □	
quiz.html - validates to HTML5 □ - At least two different types of input control used □ - specified field values echoed back from the server □	
style.css - CSS validates with no errors □ - single file external CSS applied to all HTML pages □ - consistent typological style applied to all pages □ - some CSS layout applied □	
Deployed to Mercury □	
Subtotal **(all Y)**	

Specified Requirements Place ☑ or ☒ in box - 2 marks each tick	**Comment**	**Mark**
index.html HTML *(deduct 2 marks up to -14 for each HTML5 validation error)* Menu that links (consistent menu on all pages) □ Header with appropriate context including title □ Footer with email hyperlink to your student email □ CSS: Background graphic □ Menu appropriately formatted with background colour □ Footer text small and centred □		/12
topic.html HTML *(deduct 2 marks up to -14 for each HTML5 validation error)* Headings (at least contiguous 2 levels) □ Ordered list □, Unordered list □, No. of Sections >=2 □, Aside □ Table □ Graphic < 100KB □ CSS: Table different background colour for headings □ and data □ Aside 25% viewport width □, floats right □, coloured border □ <h1> font variant, size family set □ Footer full page width □		/28
quiz.html *(deduct 2 marks up to -24 for each HTML5 validation error)* Text input for names, id and text box question □ Radio □ Checkbox □ Dropdown □ 5th input type question □ Labels linked with for □ Fieldsets used □ HTML5 data validation: Text radio input questions answered □ (6 marks) Names alpha/space/hyphen > 0, =< 25 □ ID digit 7 or 10 only□ Data for all inputs returned from server correctly □		/26
CSS (general) *(deduct 1 mark up to -14 for each CSS3 validation error)* Selectors: element □,#id □,.class □,grouping □,context □, pseudo □ Fluid page flow (relative dimensions) □		/14
Subtotal		**/80**

During the demonstration you will be asked about various aspects of your Web site and asked to explain how you have implemented it to your tutor/ assessor.

Enhancements to Specified Requirements *listed and linked* from enhancements.html
Maximum of 2 Enhancements will be assessed (put your best ones at the top of the list). Up to 10 marks are available per feature. Poorly implemented or trivial enhancements may receive less or zero marks.

Feature Name	**Described**	**Linked to where implemented on your Web site**	**Source (if applicable)**	**Mark**
	Y/N	Y/N	Y/N/na	/10
	Y/N	Y/N	Y/N/na	/10
Total Additions				**/20**

Other Deductions based on demonstration, documentation, code and file inspection

Requirement	**Max Deduction if not met**	**Deduct**
Page design		
- Well designed structure	-4	
- Appropriate contrast in colours	-2	
- Appropriate use of fonts	-2	
- Consistent application of style across pages	-2	
Content		
- Topic content has sufficient quantity (200 words +)	-4	
- Topic content is sufficient quality	-4	
HTML		
- Valid XML -	(-2 / error)**up to -10**	
- Meta-data follows in-house standard	-4	
- HTML has no embedded Style markup CSS is fully separated from HTML	-4	
- HTML elements follow in-house standard (e.g. alt on images)	-4 per element type	
- No deprecated elements/attributes used	-2	
- No inappropriate use of HTML semantics (e.g. use of <div> when <section> <article> should be used)	-2	
CSS		
- No redundant CSS or unused selectors	**up to -10**	
- Appropriate header comments (match in-house standard)	-4	
- Appropriate use of selectors (e.g. Class versus ID)	-4	
- Appropriate line comments	-2	
Web site		
- All third party content acknowledged properly*	4	
- 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: ……………………………………………………………………………………