Swinburne University of Technology
Faculty of Science, Engineering and Technology

COS10011
Creating Web Applications

Assignment Part 1, Semester 2, 2018

Develop a simple static Web site

Important Dates:

Due Date ESP	**Week 5 – the day of your tutorial: 27 – 31 August 2018, 8am** (Late submission penalty 10% of total available marks per day)
Demonstration	Your tutorial: Week 5

Contribution to Final Assessment: 12%

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

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 to be used in this part of the assignment – we will use JS in Part 2.

The Web site you will develop will describe a product/service for sale or hire.
Each student will be allocated a different type of product or service by their tutor.
It is up to you to provide details of the particular product or service and its features.
Your Web site will have the following web pages linked by a common menu:

- An introductory / home page (index.html)
- A product description page (product.html)
- A product enquiry page (enquire.html)
- A page about your assignment (about.html)
- A page that 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

Note: You should design the layout of your Web pages on paper first before implementing them.

HTML Elements

The website must be developed using HTML5 to describe the content and logical structure.
Web pages should ***not contain any deprecated elements/attributes*.**
The following HTML elements must be used in this assignment

- General
  - Comment, Head, Title, Meta, Body
    As appropriate to each page
- Structure
  - Header, Navigation, Footer, Section, Aside
- Content
  - Heading levels, Paragraph
  - Ordered list, Unordered list, Definition list, Table, Image and Anchors
  - Other elements as detailed in the page requirements shown below
  - A Form, with labelled and grouped form control elements which validate user input

Where "in-house" **templates** have been defined in this unit (e.g. for meta-data; tables; etc.) these should be followed.

All Web pages should have a consistent layout and navigation.
The HTML in your Web pages ***must validate*** against the W3C HTML5 validator http://validator.w3.org/nu and the web pages and markup ***should be well-formed XML***.
Accessibility guidelines should be followed, especially for media, tables and forms.

Hint: HTML5 validators do not necessarily check that the markup is well-formed XML. Check that 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>).

Elements such as block quotes, strong, emphasis, among others can be used if deemed necessary and appropriate for the content.

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.

Do not use iframe elements in your assignment.

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

This page should contain appropriate header and graphic related to your allocated product, and should contain a menu that links to the other pages on your Web site.
This common menu should appear on every page of your website.

2. Product range page *(product.html)*

Your tutor will allocate you a product type. For example, the product you are allocated might a "restaurant meal"; or "SLR Camera"; or a "university course".

The page must contain:

- Hierarchically structured headings of at least 2 levels
- More than one <section>
- An <aside> with appropriate content
- At least one appropriate image related to your product
  . This image should be less 100kb so it does not take too long to load.

- A table containing some data related to your product.
- At least one ordered list
- At least one unordered list

Your web page should describe a small range of products of this type (say 3 or 4). It is up to you to describe the details of the products. The products you describe can be real or imaginary. The products you describe *must* have a range of *optional features* from which the purchaser can select. For example, a restaurant meal could consist of a range of courses, with dine-in or take-away options etc. A camera might come in a number of models, with options of colours, lenses, etc.

You should write at least 150 words on the allocated product and its options. Appropriately structure the content with headings, paragraphs, lists etc. At least one product image must be included.

Sources / References:
In this assignment, you may use material from other web sites but the source of all material ***must be acknowledged***. This acknowledgement should be immediately after the material *and* include a ***hyperlinked* URL** to the original source. The text of the hyperlink reference can be a short name but the hyperlink must work. (Normally permission must be given before re-publishing content, but our websites are not live.)

3. Product enquiry page *(enquire.html)*

This page has a form where a user can lodge an enquiry about a particular product.
The form will have the following form controls:

a. First name: type *text , maximum of 25 characters, alphabetical only*
b. Last name: type *text, maximum of 25 characters, alphabetical only*
c. Email address: type *email*
d. Address (group these inputs with a fieldset and label)
    i. Street address: type *text, maximum of 40 characters*
    ii. Suburb/town: type *text, maximum of 20 characters*
    iii. State: use a select list with options VIC,NSW,QLD,NT,WA,SA,TAS,ACT
    iv. Postcode: exactly 4 digits
e. Phone number: type *text or tel, maximum of 10 digit. Use a placeholder*
f. Preferred contact*:* (email, post, phone). use *radio.*
g. Product - from the range the user wants to enquire about. use *select.*
h. Product features: use *checkboxes*
i. Comment field, for example, allowing the reader to specify particular aspect they are interested in: use *textarea*

All inputs should have labels. All form values, except the comment textarea are 'required' or have a default value (e.g. select, checkbox, radio). ***The user should not be able to submit the form if any of these required fields are blank.*** Use HTML5 validation to check the format.

Data Submission to Server
This form must have a Submit input. When this submit is activated, the name-values from the form will be sent to the server using the post http method. The server action address 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. A page about your assignment *(about.html)*

This will contain three sections:

- Student details (about you),
- Assignment Requirements (the requirements you have completed, perhaps with internal website links),
- Reflection (Your notes for you, about how and what you have done, and what you might do things differently next time. You might include links to resources that you have used.)

Student details:

Information in this section should be structured with a definition list, styled appropriately to demonstrate your skills in **must** be formatted in the following way and include:

Information	**Format requirement**
Your name	Centre aligned to the page, bold, font size 200%
Student number	Right aligned Arial font (default to San Serif if Arial not available)
Course you are doing	Colour specified as a RGB number
Photo of you.	Floating Groove border with 20px padding Size in bytes less than 100k
Your Swinburne timetable set in an HTML table	Centred Headings in bold Table cells have a background colour specified in hexadecimal number
A **mailto** link to your student email.	Centred at the bottom of page in the footer

It could also include personal profile, such as resume, interests, or information that is related to you. This extra information gives you an opportunity to extend the techniques you apply in your assignment, and could include:

- Demographic information about you
- Description of hometown
- A list of your favourite books, music, films etc.

CSS Requirements

No style markup should be included in your HTML file.

The pages in your web site must be styled with CSS and have a consistent 'look and feel', particularly with markup of 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 single ***external*** stylesheet to implement your design.

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 **product.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 can also be used if deemed necessary and appropriate for the presentation. *Your menu should have its own set of styles applied.* **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.

For this assignment you should create one **single** CSS file that applies to *all* your Web pages. This file should be named ***style.css***

Hint: CSS validators will validate against a particular version of CSS e.g. CSS2.1 or 3. This assignment should be valid CSS2.1 of 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 – see the note on Blackboard). Do not include any proprietary CSS mark-up, such as –moz- or –webkit- etc.

Enhancements to the Specified Requirements

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

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, product.html, enquire.html, about.html). The extra feature needs to enhance your Web site 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 Web site. 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 Web site*** (this is needed so the tutor can quickly assess your enhancements during the demonstration).
- ☑ All enhancements must be able to run on Firefox. 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 Part 2.*

Web Site Folder Structure and Deployment Requirements

Create a website structured as 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
  product.html
  enquire.html
  about.html
  enhancements.html
  ...other html pages
  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 "assign1/" folder – not anywhere else.
- **All** images used for the **content** should be stored in the "assign1/images/" folder.
- **All** images used used for the style should be stored in the "assign1/style/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.

Assignment Submission

An electronic copy of your assignment should be submitted through ESP at https://esp.ict.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, images, and wpLED output PDFs into a zip file named "assign1.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.
- 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.

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.

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

MAKE SURE YOU ENTER THE CORRECT UNIT CODE WHEN YOU SUBMIT YOUR ASSIGNMENT TO ESP
Submitting to the wrong unit will not be accepted as grounds for granting an extension.