

**COS10011 / COS60004 / COS60007**  
**Creating Web Applications [and Databases]**

Assignment Part 1, Semester 1, 2017

**Develop a simple static Web site**

**Important Dates:**

<b>Due Date ESP</b>	<b>Week 5 – the day of your tutorial: 28 – 31 March 2017, 8am</b> (Late submission penalty 10% of total available marks per day)
<b>Demonstration</b>	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), 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 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). **For cos10011 students you must complete the 'Academic Integrity 101' course under 'My Courses' on Blackboard, achieve 100% on the MCQs and submit a completion certificate pdf, zipped-up with the assignment submission.**

**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 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.**

You will need to provide the details of the allocated product/ service and its features for the content.

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 all the pages of 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.

## 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 *Used in most pages*
- 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.

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 not a more appropriate or meaningful HTML5 element (e.g. `section` or `strong`).

Pages should **not contain any deprecated elements/attributes** (e.g. `<i>` , `<b>`).

Do not use `iframe` elements in your assignment.

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

This page should contain appropriate header and graphic and content related to your allocated product/service, and should contain a menu that links to the other pages on your Web site.

This same menu should be in every page of your website.

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

**Your tutor will allocate you a product/service type.** For example, this might be a “restaurant meal”; or “SLR Camera”; or a “university course”.

This 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/service.  
*Image should be less 100kb to minimise page download times.*
- A table containing some data related to your product/service.
- At least one ordered list
- At least one unordered list

This 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.

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 student websites are not live, and are not published.)

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

This page will have 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, alphabet spaces hyphens only*
- b. Last name: type *text*, *maximum of 25 characters, alphabet spaces hyphens 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. Preferred contact: (email, post, phone). use *radio*. (group with a fieldset and label)
- f. Phone number: type *text or tel*, *maximum of 10 digit*. Use a *placeholder*
- g. Product/Service from the range. use *select*.
- h. Product features/extras: use *checkboxes*
- i. Comment field, for example, allowing the reader to specify particular aspect they are interested in: use *textarea* Use a *placeholder*
- j. Submit button: type *submit*
- k. Reset button: type *reset*

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 controls will be sent to the server using **post** https method and the **action** address

<https://mercury.swin.edu.au/it000000/formtest.php>.

The server script will echo back the name-value pairs to the client, enabling the form to be tested.

Nothing will be stored on the server in this part of the assignment, (we will do this in Part 3.

#### 4. A page about your assignment (*about.html*)

This page will contain three sections:

- *Student details* (about you),
- *Assignment Requirements* (the HTML and CSS requirements you have completed, with internal website links to examples where appropriate),
- *Reflection* (Your notes for you, about how and what you have done, and what you might do differently next time. Perhaps with links to resources that you have used.)

##### **Student details:**

Information in this section should be structured using section, definition list and a table. This should be styled appropriately to demonstrate your CSS skills.

- **Name:** your name
  - **Student:** your SID
  - **Course:** what you are doing
  - **Photo:** of you within an HTML figure element.
  - **My timetable:** Your Swinburne timetable set in an HTML table
  - **Contact:** A **mailto** link to your student email.
- } definition list

##### **Assignment Requirements:**

Information in this section should be structured using section and lists.

This should be styled appropriately to demonstrate your CSS skills.

This section provides you with a place to have a check list of the essential requirements and track your progress

- HTML Requirements that you have completed with internal website links to example as appropriate.
- CSS requirements that you have completed with internal website links to example as appropriate.

##### **Reflection:**

This section contains your reflective learning. A free space where you can write some brief notes for yourself, to help you reflect on how, why, what you have done, and what you might do differently next time. You could include lists of links to useful resources that you have found, so you can find them later.

The content is just for you, and your benefit, now and later, not for anyone else.

## 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 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.

Create your own design and implement it using one **single external** stylesheet that applies to *all* your Web pages. This file should be named **style.css** and placed in a style folder on the server.

The stylesheet should style the common elements on *all* your web pages, and address the following specific style requirements.

1. **Comments:** The CSS should include comments at the beginning of the CSS file to identify author and purpose. Individual line comments should be used as necessary to explain particular styles and explain where they are applied.
2. **Selectors:** *All* the following CSS Selectors should be used *appropriately* at some point in this assignment:
  - element, #id, .class, grouping, contextual
  - pseudo class, pseudo element
3. **Menu:** The menu should have its own set of styles applied. Use a background colour.
4. **Index Page:** Demonstrated the following specific CSS rules on the **index.html** page:
  - display a background graphic.
  - the footer text should be in a small font and centred in the footer.
5. **Product Page:** Demonstrated the following specific CSS rules on the **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. **About Page:** Demonstrated the following specific CSS rules on the **about.html** page:
  - Style the definition list so that each <dt> is on the left and the <dd> on the right in a single line. Set the dt to have a common width.
  - The photo should be styled with a double border using the short-hand **border**-property, and the figure should be floated to the right of the definition list
  - <table> should be centred within the section, headings in bold, table cells with a background colour specified in hexadecimal format
  - The email should be style similarly to the definition list.
7. **All pages:** should have a fluid layout (the page should “Reflow” on page resize).

Other CSS selectors and properties can be used as necessary and appropriate for the presentation

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

**Hint:** CSS validators will validate against a particular version of CSS e.g. CSS2.1 or 3. This assignment should be valid CSS2.1 or 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).

## Enhancements to the Specified Requirements

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

*The technologies for developing Web applications are rapidly changing. One of the key skills you will need is finding out about these new techniques and applying them. When researching, look at the reliable websites such as the External Links provided on Blackboard. This assessment gives you an opportunity to demonstrate your ability to implement features/techniques that go beyond the specified requirements above. It also provides you with 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 required web pages (index.html, product.html, enquire.html, about.html). The extra feature needs to **enhance** your web site in a meaningful and relevant way.

**List and describe** each enhancement implemented on the separate **enhancements.html** page, and describe how you have significantly extended the basic HTML and CSS beyond the lecture and 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
- ☒ the references to any third party sources for the 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, animation) 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.

*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:

<code>assign1/</code>	<i>You must have this folder – case sensitive!</i>
<code>  index.html</code>	
<code>  product.html</code>	
<code>  enquire.html</code>	
<code>  about.html</code>	
<code>  enhancements.html</code>	
<code>  ...other html pages</code>	
<code>  images/</code>	<i>Folder for images for your page content</i>
<code>  styles/</code>	
<code>  style.css</code>	<i>other css files</i>
<code>  styles/images/</code>	<i>Folder for images referred to by your css files e.g. background</i>

### 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/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.

**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.**

## 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 your files are in the correct folders and compress your root folder with all your sub-folders with HTML, CSS, images, **(and for cos10011 students your Academic Integrity 101 completion certificate PDF)** 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 electronically. 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 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.**