

COS10011/60004 Creating Web Applications

Assignment Part 1

Develop a simple static web site

Important Dates:

Due Date	10am, Monday, Week 5 (Late submission penalty 10% of total available marks per day)
-----------------	---

Contribution to Final Assessment: 20%

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>.

Accessibility guidelines should be followed, especially for media, tables and forms.

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

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 be “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
- At least one image

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*, *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*, use a *placeholder*

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 you (*about.html*)

This page will contain your details.

The following information should be structured with a **definition list** and formatted appropriately. For example:

Name:	Amy Smith
Student ID:	s1234567
Course:	Bachelor of Science
Email:	1234567@student.swin.edu.au

The following information should also be included and formatted accordingly:

Information	Format requirement
-------------	--------------------

Photo of you.	Inside <figure>, Floating right, Double 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

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 in 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 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**. You can use another CSS file for enhancements.

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. 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 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).
- ☒ 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 include:

- *Effective, appropriate and innovative* use of a **number** of distinct HTML elements not covered in tutorials (e.g. Image maps) 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.
- 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 as 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:

<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>	<i>Folder for style.css and 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.

Assignment Submission (Mercury + Canvas)

Your assignment should be uploaded to Mercury on or before your deadline.

An electronic copy of your assignment should be submitted through Canvas on or before your deadline.

- Make sure all your 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 Canvas. 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 Canvas. Your last submission will be marked.
- Note that all deliverables must be submitted electronically. There is no need to submit an assignment cover sheet.

Make sure you complete your Canvas submission process.

Mark Sheet – Assessed by tutor in tutorial

Declaration:

Fill this in before you start

I hereby confirm that the assignment on the Mercury server is identical to that I submitted to Canvas

Student number Student name

Signature Date

Product / Service:

Tutorial Day Tutorial Time Tutor Name

Essential Requirements	Tick box <input checked="" type="checkbox"/> if requirement met	Y/N
index.html - validates to HTML5 <input type="checkbox"/> menu that links <input type="checkbox"/> , graphic (use) <input type="checkbox"/>		
product.html - validates to HTML5 <input type="checkbox"/> menu <input type="checkbox"/> , - appropriate use of headings <input type="checkbox"/> , meaningful content <input type="checkbox"/> table, lists <input type="checkbox"/>		
enquire.html - validates to HTML5 <input type="checkbox"/> menu <input type="checkbox"/> , - form input fields (text, email, radio, select, checkbox, textarea, submit) <input type="checkbox"/> - HTML5 used to check some form control elements <input type="checkbox"/> - all field values echoed back from the server <input type="checkbox"/>		
about.html - validates to HTML5 <input type="checkbox"/> menu <input type="checkbox"/> , - dl <input type="checkbox"/> , timetable <input type="checkbox"/> , figure, photo <input type="checkbox"/> , email link <input type="checkbox"/>		
CSS - CSS validates with no errors <input type="checkbox"/> - External CSS applied to all HTML pages <input type="checkbox"/>		
Deployed to Mercury <input type="checkbox"/>		
(all Y)		

Specified Requirements	Place <input checked="" type="checkbox"/> or <input checked="" type="checkbox"/> in box - 2 marks each tick	Comment	Mark
index.html <i>HTML: (HTML5 validation errors - deduct 2 marks each, -12 max) []</i> - Meta info meets requirements (all pages, titles unique) <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> - Header with appropriate content and graphic <input type="checkbox"/> - Nav that links (consistent menu on all pages) <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <i>CSS: Menu appropriately formatted, with background colour <input type="checkbox"/></i> - Background graphic on page <input type="checkbox"/> Footer text small and centred <input type="checkbox"/>			/24
product.html <i>HTML: (HTML5 validation errors - deduct 2 marks each, -12 max) []</i> - Headings (at least contiguous 2 levels) <input type="checkbox"/> , 2+ Sections <input type="checkbox"/> , Aside <input type="checkbox"/> - Ordered list <input type="checkbox"/> , Unordered list <input type="checkbox"/> , - Table <input type="checkbox"/> Content graphic < 100KB <input type="checkbox"/> <i>CSS: <h1> font variant, size family set <input type="checkbox"/></i> - Aside 25% width <input type="checkbox"/> , floats right, margin, padding, coloured border <input type="checkbox"/> - Table different background colour for headings and data <input type="checkbox"/> - Footer full page width <input type="checkbox"/>			/24
enquire.html <i>HTML: (HTML5 validation errors - deduct 2 marks each, -12 max) []</i> - Form control elements and HTML5 validation as required: Firstname (max25 alpha) <input type="checkbox"/> , Lastname (max25 alpha) <input type="checkbox"/> , email <input type="checkbox"/> , street (max40) <input type="checkbox"/> , suburb (max20) <input type="checkbox"/> , state (select) <input type="checkbox"/> , postcode (4digits) <input type="checkbox"/> , radio <input type="checkbox"/> , phone (10 digits)+placeholder <input type="checkbox"/> , product (select) <input type="checkbox"/> , checkboxes <input type="checkbox"/> , textarea placeholder <input type="checkbox"/> , submit <input type="checkbox"/> , reset <input type="checkbox"/> - Labels linked with 'for' (test by 'clicking' on label) <input type="checkbox"/> , Fieldsets used <input type="checkbox"/> - Data for all inputs returned from server correctly <input type="checkbox"/>			/34

about.html HTML: (HTML5 validation errors - deduct 2 marks each, -12 max) [] - Student details: dl <input type="checkbox"/> , timetable <input type="checkbox"/> , figure, photo <input type="checkbox"/> , email link <input type="checkbox"/> CSS: Styled dl <input type="checkbox"/> photo double border, figure floats right <input type="checkbox"/> , - Table centred, table cell background hex <input type="checkbox"/>		/14
CSS (general): (CSS3 validation errors - deduct 2 marks each, -12 max) [] - Selectors: element <input type="checkbox"/> , #id <input type="checkbox"/> , .class <input type="checkbox"/> , grouping <input type="checkbox"/> , contextual <input type="checkbox"/> , - pseudo class/element <input type="checkbox"/> , - Fluid page flow when resized (uses relative dimensions) <input type="checkbox"/>		/14
Subtotal		/110

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 place in website)	Sourced (if applicable)	Mark
	Y/N	Y/N	Y/N/na	/10
	Y/N	Y/N	Y/N/na	/10
HTML: (deductions if invalid HTML5) []				
CSS: (deductions if invalid CSS3) []				
Total Additions				/20

Other Deductions

Requirement	Max Deduction if requirements not met	Deduct
Page design (deduct up to 2 marks each) - Poorly designed structure - Inappropriate contrast in colours - Inappropriate use of fonts - Inconsistent application of style across pages - Inappropriate application of styles (e.g. different styles for menu on each page)	-10	
Content (deduct up to 3 marks each) - Product content of insufficient quantity (<150 words) - Product content poor quality - Images inappropriate, or inappropriate file sizes	-9	
HTML (deduct up to 3 marks each) - Deprecated elements/attributes have been used - Inappropriate use of HTML semantics (e.g. use of <div> when <section> <article> should be used) - HTML usability does not follow standards (e.g. alt on images, label in forms, tables) - HTML Image height, width attributes missing or incorrect - HTML has embedded Style markup. CSS is not fully separated from HTML - Code comments inadequate to inform later code understanding/maintenance	-20	
CSS (deduct up to 3 marks each) - Redundant CSS included or unused selectors included - Inappropriate header comments - do not match in-house standard - Inappropriate use of selectors (e.g. Class versus ID) - Code comments inadequate to inform later code understanding/maintenance	-12	
Web site - Directory and file structure not as specified - Third party content inadequately acknowledged - <i>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.</i>	-4 -4 - All marks	
Total Deductions		

Comments: