Swinburne University of Technology
Faculty of Science, Engineering and Technology

COS60007 Creating Web Applications and Databases

Assignment 1
HTML + CSS

Due Date	**10am on Monday in Week 8** (Late submission penalty: 10% per day)
Submission Method	Canvas Mercury Server (mercury.swin.edu.au)
Assessment Date	Demonstration in Your Tutorial in Week 8 (No detailed feedback will be provided to students who failed to demonstrate their assignments in the tutorial.)
Contribution to Final Assessment	10%

1. Purpose

This individual assignment will familiarise you with the techniques and skills involved in designing and creating static webpages, utilising fully validated HTML and CSS created with a standard text editor. You will deploy these webpages on a web server. This should be done in a way that keeps HTML content and CSS presentation separate, as discussed in the lectures.

In this assignment you will develop a website that will enable it to advertise vacant positions. These have a 'position description' that sets out the qualifications, skills and knowledge required. Potential applicants for the position will be able to submit an online test to apply for a position. Your website will have:

- demographic information about you (`index.html`)
- a detailed description of vacant position (`job.html`)
- an online test page related to this vacant position (`test.html`)
- a page listing CSS animations you have implemented (`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 requirements for this assignment are listed in the mark sheet. In general, the webpages must:

- have relevant contents
- include the HTML markups specified in the mark sheet
- validate to HTML5 fully without errors
- have a <head> with title, meta tags (including author information)
- be styled by a fully validated CSS3 file
- be linked to each other via a menu

2. Basic Requirements (80%)

All webpages 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 webpages must fully validate against the W3C HTML5 validator (http://validator.w3.org/nu).

2.1 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 website. This menu should appear on **every** webpage of your website.
- a header containing appropriate content. It 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.

2.2 Vacant position description page (`job.html`)

You should write at least 200 words on a vacant position which is allocated by your tutor, 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
- an <aside> with appropriate content
- at least one appropriate image related to this vacant job. This image should not be more than 100k so it does not take too long to load.
- a table containing some data related to this job.
- at least one ordered list
- at least one unordered list

2.3 Online test page (`test.html`)

This page consists of a form where the user can:

- enter their first name and last name
- enter student id
- answer at least 5 questions *related to the qualifications, skills and knowledge required*.

At least five different input types should be used for those questions:

- a text input question
- a multiple choice question with one correct answer (radio button group)
- a multiple choice question with multiple correct answers (check box group)
- a dropdown list with a single correct answer.
- 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.

Before online test 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 20 character should be able to be entered.
- The student number is either 7 or 10 digits.

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 http://mercury.ict.swin.edu.au/it000000/formtest.php. The server will then just echo back the name value pairs to the client. This will allow the form to be tested.

2.4 CSS Requirements

[IMPORTANT] No CSS code 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. Your webpages should follow basic usability / accessibility principles, e.g. distinguishable foreground and background colours, and font readability, etc.

You must create a single external CSS file that styles your webpages. This CSS 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 `job.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).

Note: CSS validators will validate against a particular version of CSS, e.g., CSS2.1 or CSS3. 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 use CSS3 and validate as CSS2.1 it will show errors. (Best to pre-set the version in the Web Developer tools).

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

3. Enhancements (20%)

Implement 2 different CSS animation. There are many types of CSS animations. Take your pick and make sure they are applied to appropriate HTML elements. Feel free to use Google or any source of information you deem appropriate to look up the code you need to implement the CSS animations.

Create a separate webpage called `enhancements.html`. List and describe each animation you have made and how you have **significantly** extended the basic HTML and CSS beyond what is covered in the lectures and tutorials. Hyperlink from this list to where the animation is implemented in your website. If it is a CSS animation, hyperlink to an example of the html that is selected by the CSS rule. For each CSS animation, include:

- a brief description.
- a hyperlink to where the CSS animation is implemented on your website (this is needed so the tutor can quickly locate and assess your CSS animations).

[IMPORTANT] All CSS animations must be able to run on the version of Firefox in the labs. Make

sure you check this.

The number of marks you receive for a CSS animation will be at the sole discretion of your tutor/marker. As a guide if the CSS animations has only taken a couple of lines of code it is likely to be trivial.

- Be relevant to the content of the website
- Be well listed and linked on `enhancements.html`
- Be non-trivial
- Be significantly *different* from other CSS discussed in the lectures and tutorials

4. Website Folder Structure and Deployment Requirements

The directory structure of your website is described below:

```
assign1/              You must have this folder – case sensitive!
  index.html
  job.html
  test.html
  enhancements.html

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

Notes:

- All your HTML files should be placed in the base "`assign1/`" folder.
- 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.
- Use only relative paths for your links to your files (CSS or images). Do not use absolute paths.
- Do not include video or other large (>1MB) media files in your submission.

5. Assignment Submission

An electronic copy of your assignment should be submitted through **Canvas** on or before the deadline. Another identical electronic copy of your assignment must also be uploaded onto the mercury server.

- 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 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 through **Canvas** for several times. Your last submission will be marked.
- Note that all deliverables must be submitted as softcopy. There is no need to submit an assignment cover sheet.

[IMPORTANT] Make sure you complete your Canvas submission process. Make sure to enter the correct unit when submitting your assignment on Canvas. Submitting to the wrong unit will not be grounds for granting an extension. Another identical electronic copy of your assignment must also be uploaded onto the mercury server.

6. Assignment Demonstration Procedure

Your tutor will download your submission from **Canvas**, examine how the website works and looks and how it is written**.**

1. Make sure you attend your allocated tutorial for the demonstration. You will demonstrate your assignment to a tutor in your allocated tutorial.

2. You will demonstrate how you have implemented the website according to the items listed on the mark sheet.
3. Your tutor will not mark your assignment in the tutorial. Final results are to be released later via email from **Canvas**.

Student ID: _____

Student Name: _____

Mark Sheet

Basic Requirements (*deduct 2 marks up to -10 for each HTML5 validation error*)

`index.html` (*tick box iff the requirement is **fully** met*, 2 marks each tick) **HTML:** - fully validates to HTML5 ☐ - <head> with appropriate meta tags ☐ - title and author ☐ - menu that links all webpages ☐ - 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 ☐
`job.html` (*tick box iff the requirement is **fully** met*, 2 marks each tick) **HTML:** - fully validates to HTML5 ☐ - meaningful content ☐ - at least 200 words ☐ -heading elements ☐ - ordered list ☐ - unordered list - ☐ graphics ☐ - Aside ☐ - headings (at least contiguous 2 levels) ☐ - table ☐ **CSS:** - different background colour for table headings ☐ and data ☐ - aside floats right ☐ - <h1> font variant, size family set ☐ - footer full page width ☐
`test.html` (*tick box iff the requirement is **fully** met*, 2 marks each tick) **HTML:** fully validates to HTML5 ☐ - five different types of input control used ☐ - form values echoed back from the server ☐ fieldsets properly used ☐ **HTML5 Data Validation**: - text radio input questions answered ☐ - names and student id fields not empty ☐ - first and last names checked ☐ - student ID digit 7 or 10 only ☐
`style.css` (*tick box iff the requirement is **fully** met*, 4 marks each tick) CSS fully validates with no errors ☐ - single file external CSS applied to all HTML pages ☐ - consistent style applied to all pages ☐ - proper CSS layout applied ☐

Enhancements

CSS Animation	Properly Implemented	Properly Described	Linked to Implementation on Website
1	☐	☐	☐
2	☐	☐	☐

Other (deduct from 2 to 5 marks for each item in the table)

Requirement	Fulfilled
- Directory structure as required	☐
- Fluid layout	☐
- Appropriate contrast in colours	☐
- Appropriate use of fonts	☐
- Consist application of style across webpages	☐
- Vacant position content has sufficient quantity (200+ words) and quality	☐
- Appropriate meta-data	☐
- CSS fully separated from HTML	☐
- No deprecated elements/attributes used	☐
- No inappropriate use of HTML semantics (e.g. use of when <section> <article> should be used)	☐
- Appropriate HTML header comments	☐
- Appropriate use of CSS selectors (e.g. Class versus ID)	☐
… other requirements not specified	☐