Swinburne University of Technology
Department of Foundation & Pathways (PAVE)

COS10020 Creating Web Applications

Assignment Part 1A

Develop a simple static website

Important Dates:

Due Date Canvas	**End of Week 5 – See Canvas** (Late submission penalty 10% of total available marks per day)
Demonstration	Your tutorial session 1: Week 6

Contribution to Final Assessment: 20%

Note: You must meet the Essential Requirements of this assignment to be eligible to submit Assignment 1B (and Assignment 2), and to be eligible to earn 50% for this subject. Assignment 1A = 20% and Assignment 1 B = 20%.

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).

Purpose of the assignment

This individual assignment will familiarise you with the techniques and skills involved in designing and creating static web pages 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 and tutorials/ Labs.

No JavaScript (JS) is to be used in this part of the assignment – we will use JS in Assignment 1B.

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

- have relevant content
- must include the HTML markup specified in the marking guide
- must validate to HTML5 without errors
- must be styled by a validated CSS3 file
- must be linked to each other via a menu
- must be deployed on Mercury.

Scenario:
Ceylon Textile Suppliers (CTS), an Australian owned organization based in Melbourne which supplies clothing related products to Australian and New Zealand retail outlets on a sale or return basis. It has an annual turnover of around $ 60M and plans to expand. CTC wants to 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 form to apply for a position.

In this assignment, you will develop a prototype of this website. The website you develop will consist of the following Web pages, accessible from a common menu on each page:

- Homepage with details of the company (index.html)
- A page of job descriptions (jobs.html)
- A job application page (apply.html)
- A page with your personal details (about.html)
- A page which lists any enhancements you have made (enhancements.html)

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

You will also include

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

Content and presentation of Web Pages

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. Homepage *(index.html)*

This page should contain appropriate title, a description and graphics related to the company. It is up to you to make up the details of the company that is advertising the jobs. It 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 with an email link to your student email.

2. Position Descriptions page *(jobs.html)*

You need to write a web page with at least 2 position descriptions. *For one of these, your tutor will allocate you a job title from the above scenario*. For the second position, the choice of job type is entirely up to you.

The HTML on this page must contain:

- Hierarchically structured headings of at least 2 levels
- More than one <section>
- An <aside> with appropriate content
- At least one ordered list
- At least one unordered list
- The page should also have an appropriate footer.

Your job descriptions should be concise but as a minimum include :

- Company's position description reference number (5 alphanumeric characters)
- Position title
- Brief description of the position
- Salary range
- The title of the position to whom the successful applicant will report
- Key responsibilities. A list of the specific tasks that are to be performed
- Required qualifications, skills, knowledge and attributes. These should be divided into "essential" and "preferable". These requirements should include such things as specific technical knowledge, number-of-years experience required, etc..

The content of the job description should be appropriately structured with headings, sections, subsections, lists etc. using the appropriate HTML elements.

Sources / References:

- You can use material from other websites 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.
- If you are unsure of what is contained in a position description there are many resources on the internet. This is one of the examples: https://recruitloop.com/blog/how-to-write-a-job-description/

3. Job application page*(apply.html)*

This page has a form that allows a potential candidate to register their interest in the advertised position. HTML5 data validation should be used to check the user's input.

The form will allow a potential applicant to fill in the following:

Field	Format requirement
Job reference number	exactly 5 alphanumeric characters
First name	max 20 alpha characters
Last name	max 20 alpha characters
Date of birth	dd/mm/yyyy
Gender	radio inputs grouped using a fieldset and legend
Street Address	max 40 characters
Suburb/town	max 40 characters
State	drop down selection from VIC,NSW,QLD,NT,WA,SA,TAS,ACT
Postcode	exactly 4 digits
Email address	validate format
Phone number	8 to 12 digits, or spaces
Skill list - the last item in list should read "Other skills..."	checkbox inputs
Other skills	Textarea

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

Data Submission to Server

The form should have a submit button labelled "Apply". When this 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 submission to be tested.

4. A page about you *(about.html)*

This page will contain information on the following:

Information	HTML element to be used
Your name	Definition list
Student number	Definition list
Your tutor's name	Definition list
Course you are doing	Definition list
Photo of you < 100k	HTML figure element
Your Swinburne timetable	HTML table
A **mailto** link to your student email.	

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 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.

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 styles 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.
5. **Position Descriptions Page:** Demonstrated the following specific CSS rules on the **jobs.html** page:
    - <h1> elements should have their font variant, size and family etc. set using the short-hand **font** property.
    - 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 the footer text should be in a small font and centred in the footer.
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 single 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 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 Canvas).

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 new techniques and applying them. When researching, look at the reliable websites such as the External Links provided on Canvas. 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 website 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 contributes to higher marks 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 your 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 Assignment 1 B.*

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:

```
assign1A/              You must have this folder – case sensitive!
  index.html
  jobs.html
  apply.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 "assign1A/" folder – not anywhere else.
- **All** images used for the **content** should be stored in the "assign1A/images/" folder.
- **All** images used for the style should be stored in the "assign1A/styles/images/" folder.
- There should be a "style.css" file in the "assign1A/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 CANVAS 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 Canvas submission link on or before your 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 "assign1A.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.
- Note that all deliverables must be submitted electronically.