Swinburne University of Technology
School of Science, Computing and Engineering Technologies

COS10026 Computing Technology Inquiry

Project Part 1, Semester 2, 2024

Develop a Simple Static Web Site

Due Date	11 pm on Friday in Week 6 (Late submission penalty: 10% of total available marks per calendar day)
Contribution to Final Assessment:	40% (70% group tasks , 30% Individual tasks)

Important: 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).
The Group Agreement must have been uploaded before this assignment.

Web Development (Group Task)

Purpose

This 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 also learn about accessiblity guidelines for web in this assessment. 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.

Essential Requirements

Scenario:
An IT company 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:

- Home page with details of the company (index.html)
- A page of job descriptions (jobs.html)
- A job application page (apply.html)
- A page with your group 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 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
- 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.

All web pages in your website 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 Web pages must validate against the W3C HTML5 validator (http://validator.w3.org/nu) .

Pages should not contain any deprecated elements/attributes (e.g. <i> , <b>). Do not use iframe elements in your assignment.

Note: Generic structural elements like div or span should only be used where there is no more meaningful HTML5 element that is appropriate.

1. Home page *(index.html)*

This page should contain appropriate title, a description and graphic 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 IT industry*. For the second position, the choice of IT job type is entirely up to you. Why not write a position for the ideal job you would like to do? Be as imaginative as you like.

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 programming languages required, number-of-years of 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 web. Here is one: http://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 2) this will allow the form submission to be tested.

4. A page about your group *(about.html)*

This page will contain information on the following:

Information	HTML element to be used
Your group name	Definition list
Your group ID	Definition list
Your tutor's name	Definition list
Members Contribution to this project	Definition list (Name, Contribution)
Photo of your group < 100k	HTML figure element
Your timetable (assume you all have the same timetable)	HTML table
A **mailto** link to your group email.	

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

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

5. 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 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 (Group Task)

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 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 1 enhancement* will be assessed (*up to* 8 marks). You need to choose one from each following category:

- *Effective, appropriate and innovative* use of a **number** of distinct HTML or CSS proeprties and selectors (e.g. support for interactivity, animation) elements not covered in tutorials (e.g. Image maps, Canvas, etc) used in a way that improves the user experience of the website.
- *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.
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.*

Web Site Folder Structure and Deployment (Group Task)

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:

```
project1/              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 "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/styles/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 SUBMISSION.

Make sure you thoroughly test your website deployment on the mercury server.

Short Video (Group Task)

Create a short video to introduce and demonstrate your web application.

- Upload your video to YouTube
- Create a hyper link in the index.html page of your website, link it to your YouTube video
- Every team member must present in the demonstration video for a similar amount of time
- The total length of the video should be between 4 to 5 minutes.

Note: Failure to acknowledge the source of third party code or content is plagiarism and may result in zero marks for this assessment or other penalties in accord with Swinburne policy.

Report (Individual Task)

You need to provide a report on accessibility guideline and show the ways that the guideline have been implemented in your project. You need to run an accessibility checkers like WAVE which is a free browser extension to check your webpages and see if there is any issue and provide screenshot evidence. You need to also write a summary project report. **This is an individual task**. Your report must be professionally written (**within 1800 words**). Table 1 presents the suggested structure for the report and some sample content for each section of the report.

Table 1. Components of Report and Requirements

Component	Content
Title	• Report title • Your name & Student ID • Your Tutor's Name
Introduction	• Website introduction • Objective of the report • Outline of the report's structure
Accessibility Guideline	• Tips and roles to follow to make the web pages accessible • Example of following the guidelines • Screenshots of WAVE or other accessibility checkers
Website Content & Style	• A site map and purpose of each page and main content of each page • Technical details on how you use CSS markups • Screenshots to show the design/user interface
Your Contribution and reflection	• List and discuss your main contributions • Explain your challenges and highlights
Conclusion	• Recommendations for future improvements • Summary of the report
References	• (Optional) List of reference materials if used
Appendix	• (Optional) Information that supports but is not essential to the report

Peer Review & Feedback (Individual Task)

Every student needs to complete the peer and self-evaluation form. This task is crucial for providing constructive feedback to your peers and reflecting on your own contribution. The form can be found in the attachments of "Project Part 1 (Individual Submission)". The Peer & Self Evaluation results will be reported to each member anonymously to provide an opportunity for improvement. You must provide respectful and constructive feedback. This should be done individually.

Deliverables

The marks are allocated 70% for group tasks and 30% for individual tasks in this assignment.

- The website (group task)
- The short video (group task)
- Project report (individual task)
- Peer & Self Evaluation Form (individual task)

Assignment Submission (Canvas + Mercury)

Your website 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 website 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 “groupName_part1.zip”. Submit this to Canvas by your group leader into "Portfolio Part 1 (Group Submission)". 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 don’t need to submit the demonstration video. You only need to include a hyperlink in the index.html page pointing to your YouTube video. Ensure that the link is not private.
- You need to post the link to your website (Link to Mercury) on Canvas otherwise a penalty will be applied
- You need to demonstrate your work before the submission deadline to your tutor. Failing to do so will encounter 30% penalty.
- Every student needs to submit their individual project report to Canvas.
- Every student needs to submit their peer and self evaluation form.
- You can submit more than once through Canvas. Your last submission will be marked.
- Note that all deliverables must be submitted electronically.

Submission Summary:

- Demonstration of your website to your tutor one week before the due date.

Portfolio Part 1 (Group Submission):

- Zip file named groupName_part1.zip
- YouTube link to the video on the index.html page
- Mercury link to the website as a submission comment.

Portfolio Part 1 (Individual Submission):

- Report
- Peer & Self Evaluation Form

Good Practice / In-house Standard

(It will cause deductions if these good practice / in-house standards are not followed)

Requirement
Page design
- Well designed structure
- Appropriate contrast in colours
- Appropriate use of fonts
- Consist application of style across pages
- Appropriate application of styles
Content
- Job descriptions have sufficient quantity (200 words +)
- Job descriptions is sufficient quality
- About.html content meets spec
- Images (including portrait) present and appropriate file size
HTML
- Meta-data follows in-house standard
- HTML has no embedded Style markup CSS is 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)
- HTML follow usability standards (e.g. alt on images)
- Comments adequate
CSS
- No redundant CSS or unused selectors
- Responsive design
- Appropriate use of selectors (e.g. Class versus ID)
- Appropriate line comments
Web site
- Directory Structure as defined above
- Third party content inadequately acknowledged
- Accessibily Guideline is followed and implemented
Note: Failure to acknowledge the source of third party code or content is plagiarism and may result in zero marks for this assessment or other penalties in accord with Swinburne policy.