SENG1050 Assessment 2

Web-Based Assignment 1 (15%)

Due Date: 11:59 pm on 06 Sep (Week 6)

Outline - Template - XML pages - Web Pages Development - Submission - Marking

The objectives of this assignment are to test your ability to create XML documents that contain well-defined information and to develop web pages using HTML with CSS. This assignment will be developed further in Assessment 3: Web-Based Assignment 2.

Warning - HTML, XML, CSS Generating Tools

*You are **NOT** allowed to use tools which generate HTML, XML, XSLT or CSS code, either from a visual representation or otherwise. This includes DreamWeaver, Microsoft Frontpage, ChatGPT, saving to HTML from MS Word, etc. You must write all code yourself in a **plain-text** editor such as Notepad++. Any assignments that use code generated from a tool will be given **ZERO**.*

Outline

The board of directors for a long-running Australian household hardware franchise wishes to establish a website to remain competitive in an increasingly demanding online shopping market. Called *DIY Paradise*, the business can list their household hardware products under a wide range of categories, such as kitchen, plumbing, building, gardening, and more.

With this in mind, the board has invited you, a promising student of the University of Newcastle with web development skills, to develop their web site, *DIY Paradise*. They would like you to construct a prototype of the web page to be launched. In particular, they would like you to choose TWO of the product categories as listed above and, for each, provide a page listing THREE products for purchase. This means that, if you choose kitchen as a category, your page should list THREE kitchen products for purchase.

Note: You can make up your own categories apart from the ones listed, for example painting, flooring, lighting, etc. Also, the listed products need not be real products - be creative!

Your task is to present these products as appealingly as possible on the web. This will involve,

1. Creating an XML template and DTD
2. Creating two XML documents
3. Developing several web pages for the site.

Details of all these requirements are given below.

XML template

The template will be a **valid** XML file. The important parts here are the tags (the organization of the data), **not the actual data** about any product. However, you may need to include some arbitrary data to allow your page to be validated. For example, if the *default* of an attribute is #REQUIRED, the attribute needs a value. Otherwise, it will not be validated.

XML Specification

As the industry develops and technology advances, new categories of products may be introduced. For example, further advancements in the smart home landscape are expected to lead to more smart household products. The site's IT team, responsible for page updating and maintenance, will need an XML template that they can copy and use for all new categories (i.e., do not hardcode allowed categories into the template). The template will be used for your example products as described below.

The document should be able to store as many products as required within a given category. You will decide the tags that you wish to use to describe the products, but, for each product, you should at least have element tags for the following items:

- Name (only one)
- Description (only one)
- Pricing
- Search tags (zero or more)
- Reviews (zero or more)
- The filename of a picture of the product (one or more, use a relative path)
- URL of a webpage related to the product (use an absolute path)

The 'Review' tag should have an attribute that indicates the score of the review (from 0 to 5 stars). If you include only the items mentioned above you will **NOT** get full marks for the assignment. Please add at least 3 more meaningful tags that make your template different from others. At least one other tag (aside from the 'Review' tag) should also have an attribute.

DTD

You will also need to write a DTD for your XML. Generally, it is your choice whether the DTD is internal or external, however, for this assignment, you should make the DTD internal because it is easier to validate your XML and DTD.

Your DTD will need to describe all the tags and attributes that you used in your XML. It will also be important to use optional modifiers (+, *, ?) to indicate how many instances of each tag are allowed.

In your DTD, define at least two entities (<!ENTITY>) and use them in the XML document.

XML documents

Once you have your XML template with DTD, you will need to make 2 separate XML documents.

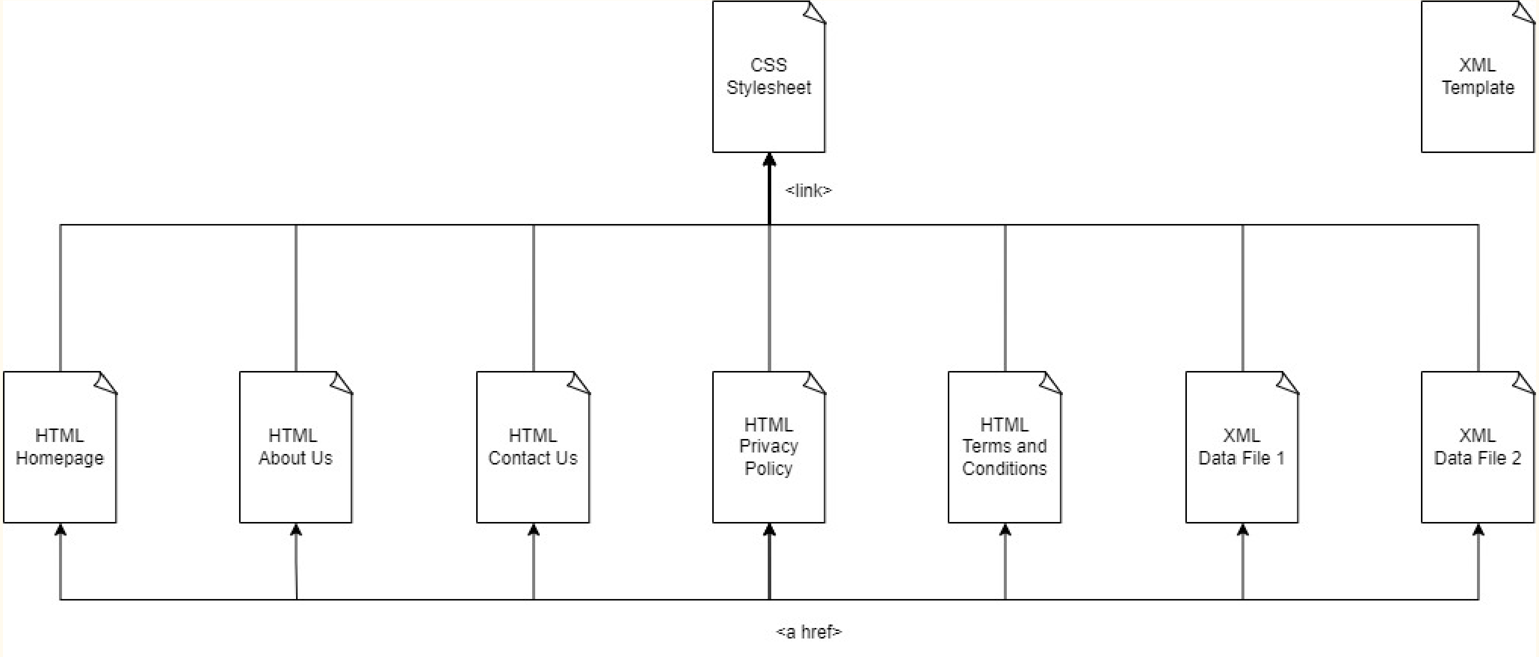
- Choose two categories and
- Write one XML document for each category.
- Each XML document should contain **data** for 3 products.
- Each XML document should contain the internal DTD at the top of the document.
- The DTDs in 2 XML documents must be the same.
- The 2 XML documents **MUST** have the same tags and DTD as the XML template.

Page development

You will develop web pages for the *DIY Paradise* website. You need to create the following files:

- Homepage (index.html)
- About Us (about.html) - Use a template such as from https://www.impactbnd.com/blog/best-about-us-pages
- Contact (contact.html) - Follow examples at https://www.ventureharbour.com/15-contact-form-examples-help-design-ultimate-contact-page/. Include options for mail and telephone - as well as a contact form for sending messages.
- Privacy Policy (privacy.html) - This is necessary as you are collecting data including personal identifying information (PII) from your guests. When creating this, you may use a template such as that at https://business.vic.gov.au/tools-and-templates/privacy-policy-template/
- Terms and Conditions (terms.html) - Use template at https://termly.io/resources/templates/terms-and-conditions-template/ and customise for your own site
- Cascading Style Sheet (style.css)

and your site map should look as below.



General requirements

- Pages should be coded according to the HTML Living Standard.
- Pages should contain the full HTML document structure, including <!DOCTYPE>, <html>, <head>, <title> and <body>
- Pages should use Cascading Style Sheets (CSS) for visual formatting. You must use one external style file, some document-level styles, and a few (NOT many) inline styles.
- Your assignment directory should contain a sub-directory, named *images* with all the image files you are using in your web pages.
- There should be a nice, non-intrusive background image for all pages.
- There should be a left side margin of 1 cm.
- The pages must have good readability, consistency and navigation. Every page should link to every other page on the site.
- Using CSS
  - Use a class attribute and a CSS .xxx selector to style multiple elements with the same class
  - At least one <div>. Use CSS to provide some interesting styling for it (your choice of borders, background, floating alignment, etc.)
  - At least one <span>, used to provide some formatting (e.g., a colour or font change) for a few words of text
- Comments for your HTML code

Detailed description for the Homepage

Your submission should contain a homepage, called *index.html*. This page should contain the following elements (in addition to the General requirements):

- A descriptive title
    - The text that appears in the browser window's title bar.
- Navigation bar (Menu bar)
    - Each page should contain hyperlinks to all other pages.
    - These links should use *a relative* URL.
    - The text colours of a link before they are visited, after they are visited and during a user action must be different from each other. A link must appear at least 10% bigger when you move the cursor over it. (You may choose your own colours but they should be obviously different).
- Headings at more than one level, with <h1> formatted to make a nice looking title for the pages.
    - Level 1 heading appropriate for the homepage. Centre the heading.
    - The page must have a couple of level 2 headings, which should be underlined.
    - Meaningful headings for each section. All level 1 and 2 headings must be in uppercase and this must be controlled by the external stylesheet.
- Several paragraphs of text
    - A paragraph welcoming people to the Web site. This particular paragraph should use a font size larger than other paragraphs.
    - Use a CSS to control at least a few aspects of text rendering (e.g., colour, font-styles, text-align, etc.)
- List THREE featured products
    - Use an unordered list or an ordered list
    - Short descriptions for each product (brief without in-depth details)
    - One image for each featured product
        - Use an <img> tag
        - Float them to the left side of the webpage using CSS. Add box information (margin, padding and/or border) to make it look nice on the webpage.
        - Keep images in the *images* folder you created.
- The above mentioned welcoming paragraph and the short description must have some abbreviations.
- One hyperlink to an external website *(e.g., http://www.newcastle.edu.au)*, but related to the course in some way. This must be *an absolute URL*.
- A copyright statement, including information about you, like your name and email address. Your external stylesheet must place the copyright statement on the right bottom corner of the page.
- Choose some other tags, such as EM, STRONG, P, or BLOCKQUOTE. You can customize those tags in your external style sheet.

Contact Form

The contact page (`contact.html`) should allow visitors to send messages to *DIY Paradise*. This will be done as a HTML form, which sends the information using a POST request to https://jkorpela.fi/cgi-bin/echo.cgi when they click the "Submit" button. The following things need to be included on the form:

- An appropriate level 2 heading
- The name of the person sending a message
- A email address for replies
- The subject of the message
- The message itself
- Whether or not the message should be considered urgent (use two radio buttons for this).
- Include buttons to submit and clear the form
- All inputs are required, the form should not submit unless the entire form has been filled out.

External Style Sheet

- Once your webpages are written, create an external style sheet (`style.css`) to define a consistent style for each page. The external style sheet should specify:
    - The style of a header tag
    - The background image, and the font family, size, and colour for the text
    - The style of the tags you used
- Use a document-level style for some tags, AND an inline style for some other tags.
- Use at least one ID and one class selector in CSS.

Coding Style

Please note that your HTML code, XML code, and CSS code **must** be neat, indented, clear, and commented. Comments are at the top of the file, stating the name of the file, the name of the author and the date the file was created. Make sure these are comments, i.e. they don't show on the web page, but only when one looks at the source code. An indentation and a clean, organized code is also helpful for you as the designer.

Other than the specifications mentioned, the way you design your page is up to you. However it should use semantically-correct tags.

Your creation should be fully valid HTML and CSS.

All code must be your own work. The use of external libraries (such as bootstrap) is not permitted.

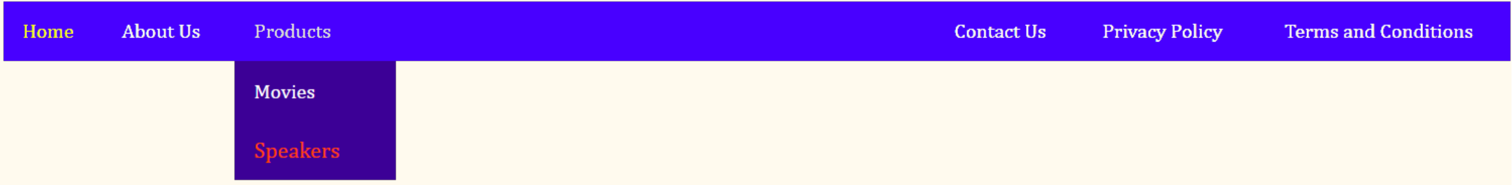
Bonus Task

Advanced Navigation Bar

- Implement the navigation bar as above, but using a CSS flexbox.

The items should appear horizontally, with links aligned on both the left and right sides of the bar.

- There should be a single item for the product categories which, when hovered over, displays a drop-down menu with links to each of the XML pages.
- None of the hover effects should cause visible changes in other parts of the navigation bar, for example, changing the position of other links or the size of other items.
- Refer to the sample below which illustrates what is required for this task (you don't have to use these specific colours and labels, the layout is what's important here):



- Students who complete this task will be awarded 5 bonus marks (does not go beyond the 100 marks cap).

Submission

Your submission should contain a template file (containing XML and DTD), and the 2 XML files (each XML file contains an internal DTD), html homepage and a CSS file. If there are any other files that are part of your assignment (e.g., graphics files), include those too. You should also include a `README.txt` file, containing your student number, a list of the files that are part of your submission, and references to any external content (such as images) that you have taken from other sources.

Finally, your submission should include a completed Assignment Cover Sheet. You can download an Assessment Item Cover Sheet for Individual Assessment from Canvas or from SUBMITTING AN ASSESSMENT, then fill in the details and include it with your submission. An Assignment Cover Sheet submitted electronically is deemed to have been signed. Your assignment cannot be marked until a completed cover sheet is received.

Warning - Plagiarism

Plagiarism is defined as presenting the work of others' as if it was your own. This includes the copying of content from other people's Web pages (and other sources) without "appropriate referencing" (indicating the parts of your assignment which have been copied and where they were copied from).

If you "quote" other people's work (or images) in your assignments, then you ***must*** *reference them appropriately (use a README.txt file to do this). Failure to do so will cause you to lose marks for the assignment. An assignment that consists largely of work copied from elsewhere will receive very low (zero) marks, and may be reported to the School for suspicion of plagiarism.*

The use of generative AI (such as ChatGPT) is forbidden for the writing of any code or for the research task. Generative AI may be used for purely creative aspects of the website (such as coming up with descriptions, welcome text, and images) as long as such content is clearly marked. AI detection tools will check each file, and any content with a sufficiently high detection score will be reported to the School for suspicion of plagiarism.

You can find more information on the University's Policy on Student Academic Integrity in the Course Outline and at Policy Library

Submission will be via the Modules page on Canvas. To submit your files, place them all into a single **`.zip`** archive containing every file and submit it using the Web form in the Assignment 1 area. The `.zip` archive should be named `cXXXXXXX.zip` archive where `XXXXXXX` is replaced by **your** student id.

Note that assignments submitted after the deadline will be penalized by 10% per day late (including weekend). For example, an assignment worth 20% marked at 78% may be graded as:

- On time: final mark = 0.78 * 1.00 * 20 = 15.6
- 1 day late: final mark = 0.78 * .9 * 20 = 14.04
- 3 days late: final mark = 0.78 * .7 * 20 = 10.92
- 5 days late: final mark = 0.78 * .5 * 20 = 7.8

Extensions will only be granted with a valid and approved Adverse Circumstances application for the assignment.

Marking Guide

- Total Marks: 100
  - XML and DTD Template Page: 25
  - XML pages: 30
  - HTML pages: 35
  - CSS: 10
  - You may be penalised for excessively poor aesthetic appearence, untidiness, spelling and grammar problems, and not correctly following instructions
- Remember, all XML, HTML and CSS pages must be valid. You should check your work before submitting at https://validator.w3.org for HTML and CSS, and https://www.xmlvalidation.com/ for XML.