



## **Assessment 4: Two pages Project (Your creative space) (25 points)**

**Due:** see the course outline for a specific date

In this project, you will practice how to build a website or product from scratch and learn steps that will help to turn an idea to an actual web page.

You can start working on your project from week 4 to week 10. Build your project part by part starting from top to bottom (depends on a page outline). Anyway, the project should not exceed two pages. Feel free to add more pages, but only two pages will be graded. The project simply is you come up with an idea and code it. However, you must make sure your project meets the project *requirements*.

### **✓ Requirements:**

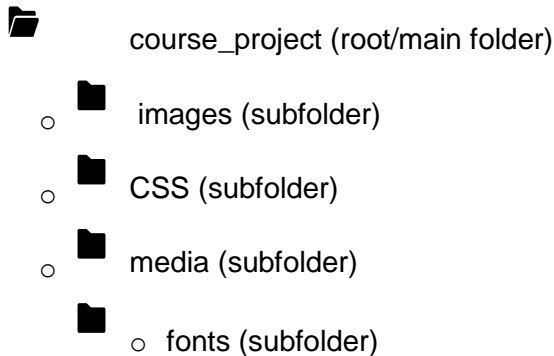
1. The design should be organized, and your code is clear. Crumble, or broken code will be rewarded with 0 (The broken system is a code that does not show a visual result or output).
2. The final product must have at least five sections (tag blocks) including header and footer.
3. At least one image, audio, and video included in the final project.
4. Fonts must be specified. You can use any fonts, but you must show it in your CSS code.
5. The website looks fine on two size screens. Here is the list you can choose from:
  - a. Mobile: minimum width is 600px.
  - b. Tables: minimum width is 768px.
  - c. Desktop: minimum width is 992px or 1200px.
6. Your code should work and look fine on Chrome and Firefox browsers.
7. You should design at least two pages that look consistent. That means they should have the same colour and fonts style.

### **✓ Instructions (Optional but recommended)**

#### **○ Step one: Set up a project folder:**

It is a good idea to create a folder for this project. The HTML document “.html” should be in the root folder (the main project folder). For extra resources such as images, create a subfolder called

“images,” and for the style sheet files called it “class” or “style\_sheet” ...etc. Your final project folder tree can look like this:



#### ○ **Step two: create a page layout using HTML:**

Now, you need to figure out how to structure the HTML page layout. Build the layout besides your design. Best way to start a layout is:

- First, write the main tags. Also separate your code using comments when creating a section, block or item/elements.
- Second, work in each section or item/element (header, middle, footer). Look closely at each element structure.

#### ○ **Step three: style and format HTML layout:**

Now, you need to use CSS to style your HTML layout. Go back to your design details and translate them to CSS code. Consider CSS properties and their values for each HTML elements.

#### ○ **Step four: check code errors and validate your code:**

Now, you need to check errors on your code.

- First, check syntax errors, if there is any, fix them.
- Second, check run-time errors, if there is any, fix them.
- Third, look if you have achieved the same result/output of this project, if not. Identify where is the problem for example, if the layout is not on the same order as required, check your HTML code first, then check your CSS code especially the position and display properties. If you have tried everything and still not getting a good result, please feel free to [email me](#).
- Validate your HTML code first, then Validate your CSS code. If you be able to get the validation icons, please include them on your final HTML product.

#### ○ **Step Five: Wrap up your project:**

Now, check all project folders and files, make sure they all open fine. Next, zip your main/root folder and only send that zip folder. Here are some links that may help with file formatting:

- How to zip a file in Windows 10: <https://www.laptopmag.com/articles/how-to-zip-files-windows-10>
- Free file compressor application for Windows: <http://www.7-zip.org/download.html>
- How to zip a file in Mac: <https://www.lifewire.com/how-to-zip-and-unzip-files-and-folders-on-amac-2260188>
- Free file compressor application for Mac: <https://theunarchiver.com/>
- How to save a pdf in MS Word document: <https://www.bettercloud.com/monitor/theacademy/save-word-doc-pdf/> .

**Please refer to the rubric at the end of this document for evaluation details.**

## **Submit**

1. Name your folder yourname\_project.
  2. Right-click on the folder and choose 'Send to -> Compressed (zipped) folder.'
  3. Upload the zipped folder to the two-page project Dropbox
- 

## **Project design and development plan (optional):**

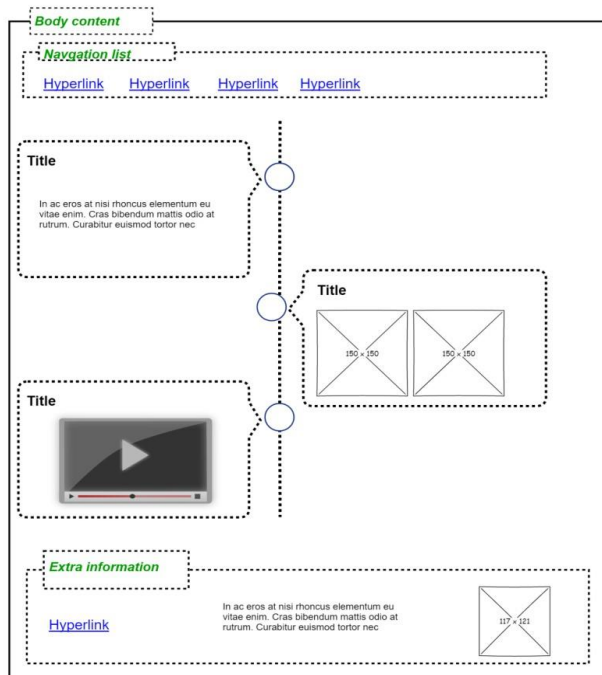
1. Week 4:
  - a. Brainstorm and build an idea or find a topic. If you need help, check the attached file called "Web design brief document.pdf." I would recommend that you start by writing everything come to your mind on a list. Then review the list and only keep things that essential to deliver your idea.
  - b. Sketch and collect content materials, e.g. images, media, icons... etc. Start by sketching your idea on paper before start designing. I would recommend that you start with a basic or general sketching structure. Keep on your mind to sketch a structure based on the materials you have collected. So, the content would fit into the structure well. I would advise you, never start coding without having a defined design of your idea. Once you finalized your sketching, you can design it in detail using HTML and CSS.
2. Week 5,6,7:
  - a. Start coding your layout for the main screen size you picked. Start with header and footer, then work on the body content.
  - b. Code the body content.
  - c. Check your code from errors and validate it
3. Week 8,9:
  - a. Make your layout responsive to fit the second screen size you picked.
  - b. Check on errors and validate your code. Then wrap up your project 😊.

## Evaluation

This assessment is graded out of 25 points and will be evaluated using the following rubric. Learners may receive partial scores or zero for unacceptable work.

	<b>Excellent: 5 point</b>	<b>Good: 3 point</b>	<b>Fair: 1 point</b>	<b>Poor: 0 point</b>	<b>Score</b>
<b>Layout/ Design/</b>	The web site has an exceptionally attractive and usable layout. It is easy to locate all of the important elements. White space, graphic elements and/or alignment are used effectively to organize material.	The web site has an attractive and usable layout. It is easy to locate all of the important elements.	The web site has a usable layout but may appear busy or boring. It is easy to locate most of the important elements.	The web site has a cluttered look and is confusing. It is often difficult to locate important elements.	<b>/5</b>
<b>Coding/ Markup Validation</b>	There are no errors in the XHTML, CSS or other coding errors on the site as found by me or an online validator.	There are 1-3 coding errors on the site as found by me or an online validator.	There are 4-5 coding errors on the site as found by me or an online validator.	There are more than 6 coding errors on the site as found by me or an online validator.	<b>/5</b>
<b>Cascading Style Sheet</b>	Student uses a style sheet to define attributes (along with some inline style), which makes all pages attractive looking and consistent.	Student uses a style sheet, however there are some inconsistencies in relation to the styles on each page in the web site.	Student uses a style sheet; however, it is not consistent on all pages, which disrupts the consistency of pages within the web site.	Student fails to use a style sheet within the pages of the web site.	<b>/5</b>
<b>Navigation</b>	Links for navigation are clearly labelled, consistently placed, allow the reader to easily move from a page to related pages and take reader where he expects to go. A user does not become lost.	Links for navigation are clearly labelled, allow the reader to easily move from page to page and internal links take the reader where he expects to go. A user rarely becomes lost.	Links for navigation are the reader where expected, but some needed links seem to be missing. A user sometimes gets lost.	Some links do not take the reader to the sites/pages described. A user typically feels lost.	<b>/5</b>
<b>Responsive</b>	Layout and content resize to fit the page width. There are no overflow issues and the page element stayed position as required. Images and fonts are resized as their containers.	Layout and content resize to fit the page width. There are overflow issues and some of the page element stayed position as required. Some of the images and fonts are resized as their containers.	Layout and content resize but does not fit the page width. There are a lot of overflow issues and a lot of the page element stayed position as required. Most of the images and fonts are resized as their containers.	Layout and content are not resizing and does not fit the page width. Elements overflow and page elements do not stay on position as required. Images and fonts are not resized as their containers.	<b>/5</b>
				<b>TOTAL</b>	<b>/25</b>

- ✓ An example of a website:



- ✓ List of software you may need to complete the project:

- Sketching:

- ✓ Pencil (simple): <https://pencil.evolus.vn/>
- ✓ Gimp: <https://pencil.evolus.vn/>
- ✓ MS Office
- ✓ Google Draw (Online)

- Free text and code editor:

- ✓ Brackets: <http://brackets.io/>
- ✓ OpenOffice: <http://www.openoffice.org/>
- ✓ Google document (Online)

- Online file management:

- ✓ Google drive: <https://www.google.ca/drive/>
- ✓ One drive: <https://onedrive.live.com/about/en-ca/>
- ✓ Dropbox: <https://www.dropbox.com/>

- ✓ Free resources:

- Images:

- ✓ Unsplash: <https://unsplash.com/>
- ✓ Gratisography: <https://gratisography.com/>
- ✓ Pixabay: <https://pixabay.com/>

- ✓ Lifeofpix: <https://www.lifeofpix.com/>
- Videos:
  - ✓ Lifeofvids: <https://lifeofvids.com/>
  - ✓ Pexels: <https://videos.pexels.com/>
- Audio:
  - ✓ Videvo: <https://www.videvo.net/royalty-free-music/>
- Icons
  - ✓ icons8 (web friendly): <https://icons8.com/>
  - ✓ Font awesome (web friendly): <https://fontawesome.com/icons?from=io>
  - ✓ Flaticons: <https://flaticons.net/>
- Fake text:
  - ✓ Lipsum: <https://www.lipsum.com/>
- Fonts:
  - ✓ Google fonts: <https://fonts.google.com/>
- ✓ Tips: Tags cheat sheet. Feel free to use any tag that is not listed below:

Head	Sections	Text-level	Grouping	Forms	Embedded
1. doctype 2. html 3. head 4. title 5. base 6. link 7. meta 8. style 9. script 10. noscript	1. body 2. article 3. nav 4. aside 5. section 6. header 7. footer 8. h1-h6 9. main 10. address	1. a 2. em 3. strong 4. i, b 5. u 6. s 7. small 8. abbr 9. q 10. cite 11. dfn 12. sub, sup 13. time 14. code 15. kbd 16. samp 17. var 18. mark 19. bdi 20. bdo 21. ruby, rt, rp 22. span 23. br 24. wbr	1. p 2. hr 3. pre 4. blockquote 5. ol 6. ul 7. li 8. dl, dt, dd 9. figure 10. figcaption 11. div	1. form 2. fieldset 3. legend 4. label 5. input 6. button 7. select 8. datalist 9. optgroup 10. option 11. Textarea 12. keygen 13. output 14. progress 15. meter	1. img 2. iframe 3. embed 4. object 5. param 6. video 7. audio 8. source 9. canvas 10. track 11. map 12. area
Interactive	Tables		Edits		
1. details 2. summary 3. command 4. menu	1. table 2. caption 3. thead 4. tbody 5. tfoot 6. tr 7. th 8. td 9. col 10. colgroup		del, ins		

To learn more about any tag, search W3C website.  
<https://www.w3schools.com/tags/>

To learn more about any CSS property, search W3C website:  
<https://www.w3schools.com/cssref/>