CSC 170 Project 3: Multipage Website

Warning: Do not begin Project 3 unless your Project 2 is perfect (or as perfect as you can get it) because any points you lost in Project 2 will also be lost in Project 3.

Assigned: **April 7**
Due: **April 28** (three weeks)

Note: there is very little flexibility in the due date because it's so close to the end of the semester. Do not push it any further than April 28. Only projects submitted on-time are guaranteed to be graded. Also, do not expect to have time to fix errors and get your project re-graded after the due date.

General and Global Requirements

In this assignment, you will create a small website using your Project 2, and three other students' Project 1s.

- ❑ Each original HTML document will be its own webpage in your website. (4 pages of content)
- ❑ You must try to maintain the integrity of each HTML document as-is. I.e., do *not* change the HTML documents by adding content, or moving content around much.[1]
- ❑ Try to use as much of the CSS you created in Project 2 as a starting point for Project 3. Your goal is to use all the original CSS you started with, and only add or tweak CSS as necessary.
- ❑ You will also create a new homepage that will serve as an entry point to all the other pages. (total of 5 pages)
- ❑ There must be a common (consistent) multicolumn layout on all pages except the homepage.[2]

It is your responsibility to reach-out and obtain the original Project 1 documents (HTML and image files) from other students.

Validation

- ❑ Each HTML document must pass validation using the WC3 Markup Validation Service at: **http://validator.w3.org**
- ❑ The CSS must pass validation using the WC3 CSS Validation Service at: **http://jigsaw.w3.org/css-validator**

[1] There are places where you have to add content – described later in this document. Also, it's okay to move content around to maintain consistency between pages. You may also remove internal navigation (if any), and resize or crop images as necessary.
[2] You may use a multicolumn layout on the homepage, but it's not required.

~~Server-side Includes~~

- ❑ ~~You must use server-side Includes: common elements (headers, footer, et cetera) that appear on each page.~~[3]
    - ~~The content of the "includes" must be factored out of each page and placed in a separate file.~~[4]
    - ~~Accordingly, all filenames must end in **.php**~~ [5]
    - ~~BTW, remember: "server" side includes only work on the server (nothing will work on your computer). Suggestion: build your website as plain HTML files first. Then chop-up your website and create the *include* files as the last step before uploading your files to the server.~~

Responsive and/or Adaptive Design

- ❑ Every page on the website must display properly on all devices with no horizontal scrolling and no elements that overlap their borders:
    - Mobile portrait (320x480)
    - Small tablet portrait (600x800)
    - Tablet portrait (768x1024)
    - Small desktop (1024x768)
    - Large desktop (1200 and up)

Search Engine Optimized

- ❑ Every page must be optimized for search engines
    - Proper title tags (in the <head>)
    - Proper semantic use of HTML elements
    - Proper outline structure using H1, H2s, et cetera
    - Proper use descriptive text in all alt="" attributes in images
    - Descriptive text in all hyperlinks
- ❑ A sitemap.xml file must exist in the website's root (in the **/project3** folder)
- ❑ Google Analytics (the <script> from Google) must be installed on every webpage.

[3] ~~Because of the difficulty in applying the "you are here" class in the current page link on the navigation, you do NOT have to turn the <nav> element into a server-side include. (But it would be cool if you could do it successfully using a bit of JavaScript.)~~

[4] ~~How exactly you slice-up the common elements is up to you. You do not have to factor-out all common content perfectly – only enough to demonstrate your understanding of server-side includes.~~

[5] ~~You do not have to use server-side includes while you're building the website. It'll be easier to leave all the pages as HTML files (with an .html extension) until the last step when you transfer your files to the web server.~~

New Content That Goes on Each Webpage

Each page in your website must have the following items **in common**:[6]

- ❑ A banner (in a <header> element)
- ❑ A main menu (in a <nav> element)
- ❑ A footer (in a <footer> element)
- ❑ The overall layout using one of the approved layout techniques[7]: inline-block; table-cell; or flex
- ❑ A "container" to keep all content centered in the viewport and constrained in 960px or less.

Title tags

- ❑ Each <title> tag in the five HTML documents must conform to a certain style.

  - The home page must follow this pattern:
    `<title>`*`yourname`*`, CSC 170 Project 3</title>`

  - The subpages must follow this pattern:
    `<title>`*`pagetopic`*`, `*`yourname`*`, CSC 170 Project 3</title>`
    (where *pagetopic* matches the H1 of the current page[8])
    (where *yourname* is your name)

The Banner

The banner (header) typically has three parts: (#1) a logo, (#2) some big text, and (#3) some subtext, like a catch phrase.

- ❑ The banner must contain some large text that acts as the name for the website.
- You can have the name for the website be simply: your name, or something else if you're feeling creative.
- You can choose whether or not to design a logo. (It would be awesome if you did.) The logo can contain the large text which acts as the name of the website, or it can be separate from the large text.
- Here's an example of a combo-logo which also contains big text that acts as the name of the website:

CHARLES PFEFFER
EXECUTIVE COACHING & LEADERSHIP CONSULTING

[6] "In common" means that these items must be identical on each page in the website.
[7] Except the homepage. The homepage (index) does not need to have the same layout as the sub-pages.
[8] If the H1 of the current page is very long, you can shorten it in the <title>; just try and capture the gist of the topic.

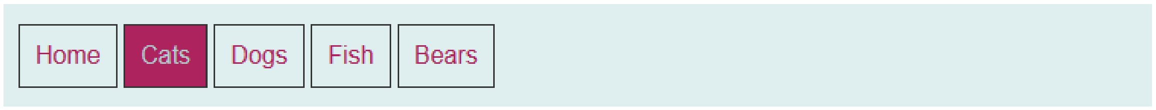
- ❑ You must include a catch phrase. If you're not feeling creative, at the very least, you can use this: *CSC 170 - Web Design and Development*

Banner Requirements

- ❑ The banner must be wrapped in a <header> tag.
- ❑ The name of the website (plus the logo if you use one) must be wrapped in an <h1>
- ❑ The <h1> (but not the catch phrase) must be wrapped in an anchor tag and linked to your homepage: **index.html**

Main Menu

- ❑ You must have a horizontal-style main menu (navigation), and links to each page in your website.
- ❑ The "current" page must be somehow styled differently to indicate that it's the current page
- Here is an example of a horizontal menu:



  Notice how the current page ("Cats" in this example) is highlighted compared to the other menu items.
- No matter what page the user is looking at, the main menu must answer the following two questions: (#1) where am I now? (#2) Where can I go?

Main Menu HTML Requirements

- ❑ The main menu must be wrapped in a <nav> tag.
- ❑ The main menu must be coded as an unordered list.
- ❑ Each list item must contain an anchor tag that links to one of the other webpages in the website.

Footer

- ❑ The footer on each page must contain your name and "CSC 170 Webpage Design and Development"
- ❑ On each of the four sub-pages (all the pages except for the homepage: index), the footer must contain a link to the location of the Project 1 from where the content came from.
- ❑ You can add other content to the footer if you're feeling creative, but whatever it is, keep in mind that everything in the footer must appear the same on each page.
- Ideas: in the footer, you can link to your Facebook page, Twitter account, whatever. You can also use a copyright like this: ©2015 *your name*. You can also insert a copy of the main menu in the footer.

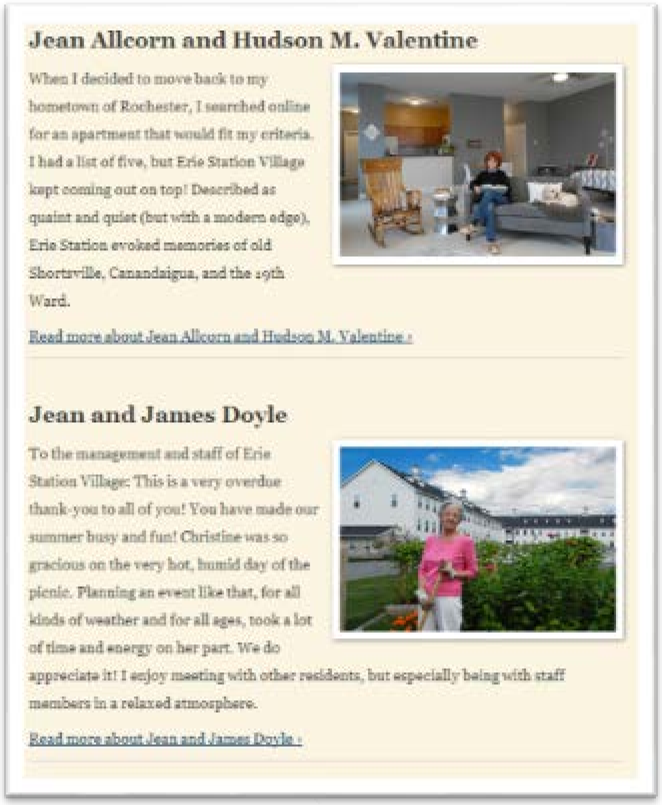
Footer HTML Requirements

- ❑ The footer must be wrapped in a <footer> tag.

New Homepage

- ❑ **Homepage layout:** You will create a new homepage (index) that will serve as an entry point to the other four pages. It will have all the same common elements (header, nav, footer and layout) as the other pages, except that it will have new content.
- ❑ **Homepage content:** The new content of the homepage must be a summary of each of the subpages: your original Project 1 document, plus the other Project 1 documents you are using. You just need to write an excerpt OR maybe just use the first paragraph of each of the subpages and make it part of your homepage. (You can use the first paragraph from any subpage only if it's a really good overview of the page it's from. Otherwise, you need to put together your own excerpts.)
- ❑ **Design:** Each excerpt must clearly indicate that it's referring to a subpage, for example, each should have its own heading and some other visual clues that tell the user that they're looking at group of excerpts that each refer to some other section in the website.
- In addition to headings for each excerpt, include embedded links that are redundant with the main nav, that also link to the subpages in your website.

1. Here's an example of excerpts that lead to subpages. (Notice the text that says "Read more…"):



Jean Allcorn and Hudson M. Valentine

When I decided to move back to my hometown of Rochester, I searched online for an apartment that would fit my criteria. I had a list of five, but Erie Station Village kept coming out on top! Described as quaint and quiet (but with a modern edge), Erie Station evoked memories of old Shortsville, Canandaigua, and the 19th Ward.

Read more about Jean Allcorn and Hudson M. Valentine ›

Jean and James Doyle

To the management and staff of Erie Station Village: This is a very overdue thank-you to all of you! You have made our summer busy and fun! Christine was so gracious on the very hot, humid day of the picnic. Planning an event like that, for all kinds of weather and for all ages, took a lot of time and energy on her part. We do appreciate it! I enjoy meeting with other residents, but especially being with staff members in a relaxed atmosphere.

Read more about Jean and James Doyle ›

Note: using embedded images as shown in this example would be awesome, but not necessary.

Design considerations

It will be difficult imposing the new common elements (header, nav, footer, layout) on the previously written Project 1 documents. You will have to make *some* changes to them.

- ❑ In each Project 1 document, you can add, remove or change elements, and add or change classes as needed. However, you must not make "content" changes to any original document. You must try to maintain the original intent of each document from a content standpoint.

For example: you need to impose a new <header> element with its own <h1> on each subpage in your website. But each subpage had its own <h1> (and maybe a <header>). Now that these pages are becoming part of a bigger, multi-page website, you will have to demote each heading:

- <h1> will become <h2>
- <h2> becomes <h3>
- ...and so on

You have to figure out how to make minimal changes to each original HTML document, but still make them all work within the structure of your multipage website.

Suggested Workflow

Prepare your Project 2 HTML file

1. Start by creating a new/empty folder named **project3**
2. Make *a copy* of your entire Project 2.
3. Rename the Project 2 HTML file from index.html to something that describes its content (something like it's H1), e.g. **mark-zuckerberg.html**
4. In your copy of your Project 2 page, add the "common elements" described earlier in this document, restructuring the HTML as you go, as necessary.

Register your Project 3 website with Google Analytics

5. In Google Analytics (www.google.com/analytics), setup a new Google Analytics Account for your Project 3 site
    - The site will (eventually) reside at
      ***accountname*.rochestercs.org /project3/index.html**
      (where "accountname" is your Bluehost account name)
    - Find and copy the tracking code for your Project 3 Google Analytics Account.
6. Insert the tracking code for your Project 3 Google Analytics Account into the code of your HTML file.

Add the Other Project 1 files

7. When your copy of your Project 2 page is perfect make a copy of the HTML file. Name the file "start.html"
8. Hollow out the content the start.html file (not the common elements – just the content).
9. Make three copies of your start.html file. Name them names that describes the content that will go in them (the content from the other students), e.g. **alan-turning.html**, **tim-wu.html**, and **biz-stone.html**.
10. Fix the common navigation links so they all work correctly.
11. Carefully copy HTML content from the other students' HTML pages. Be sure to *copy the code* from the other HTML files – not the text from a web browser.
    - You *will* need to re-structure the other pages' outlines (the H1, H2s, et cetera) when you bring them over.
    - If you have a Project 1 that used its own internal navigation, you may remove it.
    - Other than those things, you need to use as much of the HTML from the other students without modification, as much as possible.

Build your Homepage

12. Make another copy of your start.html file. Name it **index.html** and fill it with the excerpts and links as described earlier in this document.

~~*Use of Server-side Includes*~~

~~Note: up to this point, all your HTML files should be ".html" files so you can work on your local computer and see how things look as you work.~~

13. ~~After the entire site is perfect, hack-up your html files separating the common elements into ".inc" files, and replacing them in each HTML file with PHP include statements.~~
    - ~~Example:~~
        i. ~~Cut out all H1 content in each HTML file~~
        ii. ~~Save the H1 content in a file named **inc/header.inc** …or something like that~~
        iii. ~~Replace all the cut-out H1 content in each HTML file with:~~
        ~~`<?php include "inc/header.inc"; ?>`~~
14. ~~Rename all **.html** files to **.php** files.~~
15. ~~Fix any internal link: all the links on the homepage; all the links in the navigation. Make sure they point to **.php** files, not **.html** files any longer.~~

~~Remember: after you do these steps (13-15) all your webpages will no longer work normally on your computer.~~

16. Upload the entire project to the web server and check it out. ~~If the server-side includes don't work, fix them now!~~

Generate a sitemap.xml *File*

17. Create a **sitemap.xml** file and insert it into the root of your website - in the **/project3** folder on the server.[9]

Location

- ❑ Your multipage web site with image(s) and CSS must be mounted on the class webserver, where you'll create a new folder that must be exactly this:

  project3

  …all lower case, spelled exactly like that.

- ❑ The name of the homepage must be exactly this:

  index.html

  …all lower case, spelled exactly like that.

- ❑ Subpage filenames can be whatever makes sense for the current page, so long as they follow standard naming conventions for webpages: all lowercase. No spaces. Use hyphens to separate words. E.g. for a webpage titled: "Mark Zuckerberg, CEO of Facebook", you would use "**mark-zuckerberg.html**"

- ❑ The image(s) you use must be in a subdirectory of the **project3** folder, and the subdirectory must be named **images**

- ❑ The external CSS file must be in a subdirectory of the **project3** folder, and the subdirectory must be named **css**

- ~~❑ The PHP includes must be in a subdirectory of the **project3** folder, and the subdirectory must be named **inc**~~

In a web browser (any), go to this address to check your handiwork:
***accountname*.rochestercs.org/project3/index.html**
(where "*accountname*" is your account name)

Also, create a ZIP file of your entire project3 folder and attach it as part of your project 3 submission in the Blackboard assignment. ← NOTICE THIS!!!

[9] You can use: **www.xml-sitemaps.com** to generate your sitemap.