HW 2

First Web Page – 2 PARTS

Date Due: **September 22, 2020**

Part 1: Creating Your First Web Page

What This Assignment Is About

In this assignment you will create your first web page. That may sound easy, but there are a number of pieces involved in getting yourself and your account ready for the more complex assignments to follow. To complete this assignment successfully, your first webpage must be visible to everyone on the cs.uml.edu server.

What You Are To Do

1. If this is your first time using ssh, I recommend using a software for ssh and managing your files such as Mobaxterm for windows. A software is not entirely necessary, you can still manage everything with simple putty or powershell application on your computer.

2. Verify that your file system is set up correctly. If the directories described below do not exist, create them.

    a. Log into either your cs.uml.edu account. (ssh to cs.uml.edu)

    b. Under your main directory you should find a subdirectory named **`public_html`**. If it does not exist, create it.

    c. Check that the protection on directory `public_html` is **`drwx--x--x`** (711). If it is not, use the command `chmod 711 public_html` to set it to this protection.
        - *Do not change this protection! Doing so will cause you to be suspected of cheating by allowing other students to copy your files.*

    d. Check that your `public_html` directory has a subdirectory named `WEB-INF`. If it does not, create it.

    e. Check that your `WEB-INF` directory contains two subdirectories named `lib` and `classes`. If it does not, create them.

    f. Check that the protection on your `WEB-INF` directory and its `lib` and `classes` subdirectories is **`drwx--x--x`** (711). If it is not, use the `chmod` command to set this protection.

3. Create an HTML5 web page in your `public_html` directory that introduces yourself and can be used as an index to the future assignments in this course. Your web page need not be fancy, but it should show that you have put in effort to apply the techniques and use at least the tags that we have discussed in class so far. Try to design the page so that it can be used as an index to all of the future assignments you will do for this course. But again, your page need not be fancy at this point. You can continue to improve it throughout the semester as you learn new techniques.

Make sure your page have the basic html5 structure discussed in class. It can also be found at: https://www.sitepoint.com/a-minimal-html-document-html5-edition/

4. Set the protection of your HTML file to `644` (`-rw-r--r--`) using the command `chmod 644` *your-HTML-file-name*

5. Make sure that your file contains the following documentation within `<!-- ... -->` tags at the top of the file:

    a. your full name
    b. your e-mail address
    c. your affiliation as a student at **UMass Lowell** in course **91.61 GUI Programming I**
    d. the date on which you created the file
    e. a short description of what your Web page does (this can be very short for this simple page)

    This documentation is very important and will be a major part of your grade for this assignment. Most students who do poorly on this assignment simply don't put enough effort into the documentation.

    Here is an example of proper documentation at the top of an HTML file. Documentation in this format appears at the top of virtually all the Web pages that I create by hand. You can see the documentation below in this file by selecting View->Source in your browser.

```
<!--
  File:  /~heines/91.461/91.461-2013-14f/461-assn/FirstWebPage-v03.jsp
  91.461 Assignment:  Creating Your First Web Page
  Jesse M. Heines, UMass Lowell Computer Science, heines@cs.uml.edu
  Copyright (c) 2013 by Jesse M. Heines.  All rights reserved.  May be
freely
    copied or excerpted for educational purposes with credit to the
author.
  updated by JMH on September 2, 2013 at 9:23 AM
-->
```

    Note: The extension for this page is `.jsp` because this is a JavaServer™ Page, not an HTML5 page. Your file's extension should be `.html`.

6. Make sure you properly document your code (see grading rubrics for details)

7. Verify that you can view your web page over the Internet.
    a. Open a browser and enter: `http://cs.uml.edu/~`*your_login_name*`/`*your_html_file_name*`.html`
        - *If you are sure all the steps for file permission is correct and your page is still not showing, contact our system administrator Tuyen Nguyen (tuyen@cs.uml.edu). There could be an issue on your account.*

    b. You are expected to use index.html file as your home page, this way when you enter: `http://cs.uml.edu/~`*your_login_name*`/` → this will bring up the same page as: `http:// cs.uml.edu/~`*your_login_name*`/index.html`

    c. note that you must ***not*** include `public_html` in the URL to your page
        - when the server sees *~your_login_name*, it automatically goes to the `public_html` directory in your account
        - this is a feature of the Apache HTTP server running on a Linux system

8. Test your page using either:

    a. the W3C Markup Validation Service (http://validator.w3.org/)
    b. the HTML5 Living Validator (http://validator.nu/)

To receive full credit on this assignment, your page must validate against one of these two validators with no errors. You must then indicate which validator you used (one or the other or both) on the Assignment Submission Form (see below).

How You Will Be Graded

This part of the assignment will be graded on a 20-point system with points awarded as follows. Please note that the lists of features provided below are not meant to be exhaustive. They are merely representative of the types of things we are looking for in each grading category. Also note that 5 of the 20 points, or 25% of your grade, is awarded for documentation.

I reiterate that your page need not be fancy. I am very aware that some students enter this class with significant prior knowledge of HTML, CSS, and JavaScript. Those students are encourage to use — and more importantly to expand upon — their existing knowledge, but no extra credit will be given ***at this time*** for things that we have not yet covered in class. (In the future I ***will*** expect you to go "above and beyond" what we have covered in class.) All students should be able to score all 20 points on this assignment, regardless of what you knew before you took this course.

Criteria	*Possible Points*
Access • web page is visible by our teaching assistant and me over the Internet from `http://cs.uml.edu/~`*your_login_name* (see Step 7 above)	5
Content • web page must be set up so that it can serve as an index to all future assignments in this course (see Step 3 above) • content is non-trivial, that is, your page contains more than I would expect from a non-major taking 91.113 Introduction to the Internet • all content is written in proper English with correct spelling	5
Structure • properly structured as an HTML5 document • passes validation (see Step 7 above)	5
Source Code Documentation and Formatting (see Step 5 above) • user name and pertinent contact information appear in each source file • all files contain adequate explanatory documentation • all files have adequate white space for readability • all files are properly indented and formatted • all documentation is meaningful and does not merely echo code • ***any sources used are cited in comments embedded within code*** *Additional Notes:* • documentation is critically important in this course and is where many students lose points, especially on their first few assignments • be sure to include all the necessary information and document your work thoroughly	5

• *note that in this assignment, documentation is 25% of your grade!*	

Part 2: Styling Your First Web Page With CSS

What This Assignment Is About

This part 2 builds on part 1. You are to enhance that page with additional HTML techniques and apply CSS to your HTML to give your page some basic styling. In this homework, please refrain from using bootstrap.

Information relevant to this assignment appears in *CSS* chapters of our book. We obviously can't cover all that material in class before the assignment's due date, but by reading these chapters and looking at examples on the W3Schools website, specifically the tutorials at www.w3schools.com/css and www.w3schools.com/css3 you should be able to fulfill the assignments requirements.

What You Are To Do

1. Make sure you have read substantial amount of information on CSS through W3Schools, online tutorials, or textbooks.
2. Redo the page that you created for Part 1 or create a totally new page from scratch. The page you choose must be "rich" enough to provide you with ample opportunities to style elements. You may even wish to use a page created by someone else and add styling to that page. Of course, if you choose a page that you did not initially create yourself, identify its source and author (or company if you can find an author named).
3. Using the techniques discussed in the textbook and in class, style the text on the page you have chosen as you see fit. Explore the various CSS text properties and their possible values discussed and listed at http://www.w3schools.com/cssref/default.asp#text. If you have a friend who is artistic, you are welcome to ask him or her for help, of course naming that person in your code documentation to give him or her credit for his or her input.

Code Criteria

- Your CSS code must be in a separate file from your HTML file, in a subdirectory name **css**. This is the industry-standard way to organize websites. You might use inline CSS and/or an internal style sheet during development, but you must move all your CSS to a separate, external file before you submit your assignment. That external file should have an extension of `.css`.
- Your CSS code must be documented to the same standard we have used for HTML. That is, major sections must be documented to identify what they are styling. You should ***not*** document each individual CSS rule. Assume that that reader of your documentation knows CSS, but that he or she is not familiar with the way you're using it.

- Test both your web page and your CSS code using the W3C Validators:
  - HTML5 Markup Validation Service: http://validator.w3.org or http://validator.nu
  - W3C CSS Validation Service: http://jigsaw.w3.org/css-validator

  Your code must pass both validators. When it does, add the W3C validation icons to your page.

Code Review

- You can learn an awful lot by just having someone else look at your page and try it out. If you do this assignment before the due date, there is no reason why you can't ask a classmate to review your work for you. I strongly suggest that you do this, and that you do so early enough to allow time to make the changes that he or she recommends that you also agree with.

Submitting Your Assignment for Grading

Please follow these directions carefully. Incorrectly submitted assignments will not be given appropriate credit.

To submit this assignment you must go to our course blackboard and submit prior to the deadline. Please submit all related files and also a link to your styled site in a readme file (all in a zip).

How You Will Be Graded

This part will be graded on a 20-point system with points awarded as follows. Please note that the lists of features provided below are not meant to be exhaustive. They are merely representative of the types of things we are looking for in each grading category. Also note that 4 of the 20 points, or 20% of your grade, is awarded for documentation.

Criteria	*Possible Points*
Program Integrity / Design • page displays properly as intended from the weblab.cs.uml.edu server (1 point) • source code for all files is visible on the web and/or accessible via links on the assignment or index page (1 point) • all CSS resides in a separate file with an extension of `.css` in a subdirectory named `css` (2 points) • page exhibits creativity and effort, not just trivial implementations (8 points) • page validates using the W3C Markup Validation Service (2 points) • page validates using the W3C CSS Validation Service (2 points)	16
Source Code Documentation and Formatting • user name and pertinent contact information appear in all source files • code is logically organized • all files contain adequate explanatory documentation • all files have adequate white space for readability • all files are properly indented and formatted • all documentation is meaningful and does not merely echo code • ***any sources used are cited in comments embedded within code***	4