Assignment 2

- Due Friday by 11:59p.m.
- Points 75
- Available after Mar 1 at 12a.m.

Assignment 2

This assignment is based on HTML, CSS and basics of Python Flask

Question 1(25 points)

For this question, you will need to get Python installed on your and your friend computer (assuming that you are working in groups of 2). Create a file called Question1.py that has a method called *user* that takes in as input a string and checks whether the string is in upper case or lower case. If it is in upper case, you must convert the string into lower case and if it is in lower case, you must convert the string into upper case. Your method *user* must generate back a HTML string that greets the person with the given name. Here is how we will test your code out, and you can do the same on your machine:

1. Make sure you have Flask installed on your computer.
2. Make sure that you have attended/watched the lecture on Flask.
3. Make sure that you have the appropriate app routing inside your Python code.
4. You can run your code in multiple ways, however, if you wish to run your code using the terminal on your computer (Mac or Linux) then type in the following:

env FLASK_APP=Question1.py flask run --host 0.0.0.0

5. Open your browser Chrome, and type in the following URL http://localhost:5000/dave

Note for the above example, you are passing in the name, dave, as lower case, and you should see in the browser a greeting for dave as 'Welcome, DAVE, to my CSCB20 website!'. You are free to edit the greeting in HTML in any way you like.

6. Open your browser Chrome, and type in the following URL http://localhost:5000/DAVE

Note for the above example, you are passing in the name, DAVE, as upper case, and you should see in the browser a greeting for DAVE as 'Welcome, dave, to my CSCB20 website!'. You are free to edit the greeting in HTML in any way you like.

7. Open your browser Chrome, and type in the following URL http://localhost:5000/ Dave

Note for the above example, you are passing in the name, ' Dave', with two spaces before the name, and you should see in the browser a greeting for DAVE as 'Welcome, Dave, to my CSCB20 website!'. You should remove the added spaces before Dave. You are free to edit the greeting in HTML in any way you like.

8. Open your browser Chrome, and type in the following URL http://localhost:5000/D1A2V3E123

Note for the above example, you are passing in the name, D1A2V3E123. Because this contains numbers, you do not care at this point whether the rest of the characters are upper case or lower case. You simply remove the numbers and return back ‘Welcome, DAVE, to my CSCB20 website!’.

9. Open your browser Chrome, and type in the following URL http://localhost:5000/dAve

Note for the above example, you are passing in the name, dAve, and you should see in the browser a greeting for DAVE as ‘Welcome, Dave, to my CSCB20 website!’.

10. You can assume that there will be no space between names, and that we will not deviate from the above syntax that is provided to you for testing your code.

Question 2 (50 points)

2.1 Introduction

In this assignment, you will develop a new course website for the course CSC B20 using HTML and CSS. You are **not allowed** to use any web framework such as BootStrap and JQuery etc. You may work individually or in groups of two or three. The webpage must be usable, responsive, and visually appealing. Below is an example course page, however it is up to you to create a design following the mandates listed in this assignment. I want you to have some fun with this assignment. There are some minimum requirements, but for the most part, there is a lot of creativity!

Think about what an ideal course website should look like? How should the user experience be like? What are the relevant contents that must be shown to the visitor of the website? What should the look and feel of the website be?

Does my website work on mobile phones? How should the CSS/HTML change to accommodate different mobile devices?

etc. etc. Here are some course websites that you can look at to get some ideas

- http://www.cs.toronto.edu/~ahchinaei/teaching/20165/csc148/
- https://www.teach.cs.toronto.edu/~csc148h/winter/
- http://www.utsc.utoronto.ca/~bretscher/b20/
- http://www.cs.toronto.edu/~mashiyat/csc309/

The course webpage that you create, **MUST** have content (i.e., name of instructor, course information, and course syllabus) similar to this course website of CSCB20 i.e., http://www.utsc.utoronto.ca/~bretscher/b20/.

However, it your job to redesign the look and functionality of the page, as well as making it responsive. You are redesigning and building a website for http://www.utsc.utoronto.ca/~bretscher/b20/, so make sure you use the name of the instructor, photo of the instructor and any other details that you require from http://www.utsc.utoronto.ca/~bretscher/b20/.

Your project will be marked based on the following criteria:

2.2 What is expected on your website at the very minimum?

1. All content and data on http://www.utsc.utoronto.ca/~bretscher/b20/ must be reflected in your website.
2. Links to the following must be included: Piazza, Markus, Assignments, Anon Feedback, Syllabus, Labs, CourseTeam, and Home. You can create a simple dummy pdf/ or a powerpoint file. And have this same file linked to any lecture material or course-related material on your website. For example, instead of 12 pdf or 12 ppt files for each of the 12 weeks on your course website, you can create one sample pdf or ppt file and have 12 different links on your course website refer to this single pdf or ppt. Make sure to have some meaningful name on the links by referring to the aforementioned CSCB20 website. Do not have external links referring to the aforementioned CSCB20 website.
3. Links should react to mouse hovers and clicks (Apply some style!)
4. Header must visibly stick to the top of the page as the user navigates the page
5. Footer is required:
    - Must stick to the bottom of the webpage, should not be visible until the user reach- es the bottom of the page
    - Must contain a link to the Faculty of Computer Science at UofT
    - Must contain a blurb about who created the website (Site design by Jane Doe)
6. You must use CSS grid layout or flex layout to design the header and footer section.
7. All navigation links should direct the user to the appropriate panel and sections of the panels.
8. Page should be responsive and adapt to tablet and mobile views
9. Use of sections and white space should be used to delimit content areas (e.g., Assignment section should be different than Syllabus section)
10. Your HTML and CSS code should be readable and maintainable.
11. You must have all your CSS on a separate .css file and have it referenced inside an HTML file by using the<link> attribute inside the header portion of the HTML file.

2.3 Requirements

- Before starting to code in HTML and CSS, you are required first to generate some basic mock-ups (full-size model of a design that demonstrates what the website would look like) of your website. These mock-ups will show how the user will interact with your website.

– Each mock-up will refer to a single user story. Following are some user stories that may help you in generating mock-ups for your website:

1. “As a student user, the student must be able to see all the lecture slides when clicking on the ‘lectures’ button/link.”
2. “As a student user, the student must be able to see all the Office Hours for this week on Google Calendar or Google Maps (maybe you want to show where these office hours are happening?), when clicking on the ‘Office Hours’ button/link.”
3. “As a student user, when I click on the lab’s page on the CSCB20 website, a grid layout of all the labs must be shown with information of all the TA.”
4. .
5. .

- Each group may have any number of user stories. But you should have **at least 10 user stories**. For each user story, you should mock up what the web page should look like. You can generate your mock-up using many online free tools. One of them happens to be this https://gomockingbird.com/home. You are free to use any other website of your choice for mock ups.
- Make sure each mock-up is saved as a PNG file. And that there is a direct mapping between the mock-up and the user stories.
- Complete this assignment using exclusively only HTML and CSS. You must use **table-less design** (no table element is allowed).
- You cannot use any predefined or 3rd party CSS framework or an HTML template. Everything should be written in HTML and CSS in scratch and by you.
- The main entry page for your site should be called **html**. Your CSS styles must be saved separately inside its CSS file. For example, the style of index.html must be inside index.css.
- Have your course webpage work well with mobile devices. What changes are required to make it visually appealing on mobile devices?

Deliverable

Your assignment should be submitted as a zip (assignment2.zip) file on Markus. When we unzip your zip file, it should contain a single directory called ‘Assignment2’ which in turn contains the following:

- A directory called **MockUpAndUserStories**: This directory will contain two sets of items i.e.
  - 1) MockUp and 2) UserStories. Your Mockup png should be named as ‘mockup1.png’, ‘mockup2.png’, etc. Your user stories should be named as ‘userstory1.txt’, ‘userstory2.txt’ etc. The ‘userstoryX.txt’ must correspond with the file ‘mockupX.png’ where X is any number.
- A directory called **src**: In this directory, you will place all your HTML and CSS files. One of your .html files must be called index.html that we will open in Google Chrome browser to mark your assignment. Your ‘index.html’ will be the starting point of your entire CSCB20 webpage. Each HTML file that you create must have a corresponding CSS file. All your HTML and CSS must reside in this directory. If your page makes use of any images or other resources, make appropriate subdirectories inside ‘src’ and have your HTML and CSS reference them.
- A file called **HonorCode.txt**: You must enter the following blurb in your honor code signed (by entering your and your partner first name, last name and student #).
  - **If this file is absent, we will not mark your assignment, and you will receive zero on the entire assignment.**
  - **Honor Code:** We (**your and your parters’ name**) pledge that this program represents my own program code and that we have coded on my own. We received help from no one in designing and debugging my program. We have also read the plagiarism section in the course info sheet of CSC B20 and understand the consequences.
- A file called **Report.pdf**. In this report (1-2 pages) you must try and answer these questions:
  - What issues and problem did you notice with some of the previous websites?
  - How did you address these issues in your redesigned course website?
  - What are some challenges that you and your team member faced? How did you go about addressing these challenges?

- A file called **Question1.py** that contains your Python code for question 1. Please make sure that this code runs, otherwise, we will straight away give you zero if the code does not run on our computer. You can safely assume, that we will run your code on a machine that has Python 3.10.x installed.

Resources that may help:

- CSS flex: **https://internetingishard.netlify.app/html-and-css/flexbox/ (https://internetingishard.netlify.app/html-and-css/flexbox/)**
- CSS grid layout: **https://learncssgrid.com/ (https://learncssgrid.com/)**
- Online sandbox for trying out basic ideas: **https://codepen.io/LandonSchropp/pen/KpzzGo (https://codepen.io/LandonSchropp/pen/KpzzGo)**
- HTML wireframe mockup: **https://gomockingbird.com/home (https://gomockingbird.com/home)**
- Story mapping: **https://www.eecg.utoronto.ca/~shuruiz/teaching/ECE444-2020F/slides/Lec7.1-StoryMapping&Risk&Prototype.pdf (https://www.eecg.utoronto.ca/~shuruiz/teaching/ECE444-2020F/slides/Lec7.1-StoryMapping&Risk&Prototype.pdf)**