**Web Programming and Design**
**CGS 3066: Spring 2023**
**Department of Computer Science**
**Florida State University**

**Assignment 4: 50 points**
**Due date: 04/04/2023 @ 2:30 AM**

**Objective:** Familiarization with HTML form validation using JavaScript.

In this assignment, you will implement HTML form validation using certain rules (i.e., certain specified requirements). You will implement this HTML form validation for an existing HTML form. If the data entered by the user does not satisfy any of the validation rules, the web page will bring appropriate error messages.

**Instructions:**

1. Download the contents of the Assignment 4 folder in the Canvas course website. You will notice that an easy way to download these contents of this folder is to download the entire folder as a single compressed file. Extract the compressed file in your machine.
2. The template will contain an index.html file, a CSS file and an **empty** JavaScript file. The following is a screenshot of the web site as soon as it is opened in a browser:


On the left side, the web page displays a form that the user will complete and submit using the *Submit* button. After the form is submitted, the web page will react by displaying content, in the *Result* column, in one of the two following ways (the *Result* column is the column on the right side of the web page shown above):

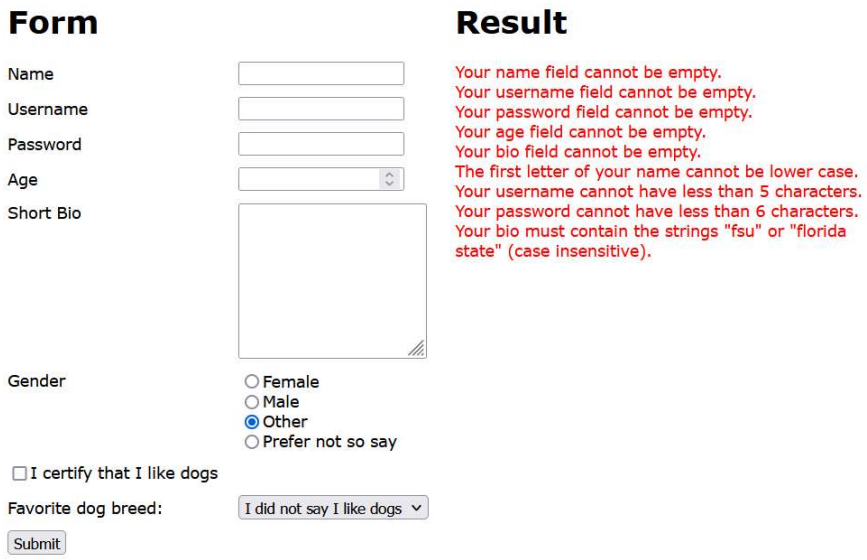
a) The *Result* column will display the information entered by the user. This is considered a *successful* form submission.
b) The *Result* column will display information, which notifies the user of the validation rule(s) or specified requirements that the data they provided violated. This is considered a *failed* form submission.

3. Implement, using JavaScript code, the following validation rules for the form:

   **Points: 35**

   a. All the fields that receive text must be completed (i.e., Name, Username, Password, Age, Short Bio) are required. **(5 points).**
   b. Name must have its first letter in uppercase. For thus rule, you may assume that a name comprises of only all alphabet letters alone, as there is no reasonable definition of lowercase or uppercase characters for non-alphabet characters. **(5 points).**
   c. Username must have at least 5 characters. **(5 points).**
   d. Password must have at least 6 characters. **(5 points).**
   e. Age must be a number greater than 0. **(5 points).**
   f. Bio must contain the strings "fsu" or "florida state." This rule is not case sensitive. **(5points).**
   g. If the user does not check the *I certify that I like dogs* checkbox, then she is not allowed to select a favorite dog breed (i.e., in that case, she can only select the *I did not say I likedogs* option). **(5 points).**

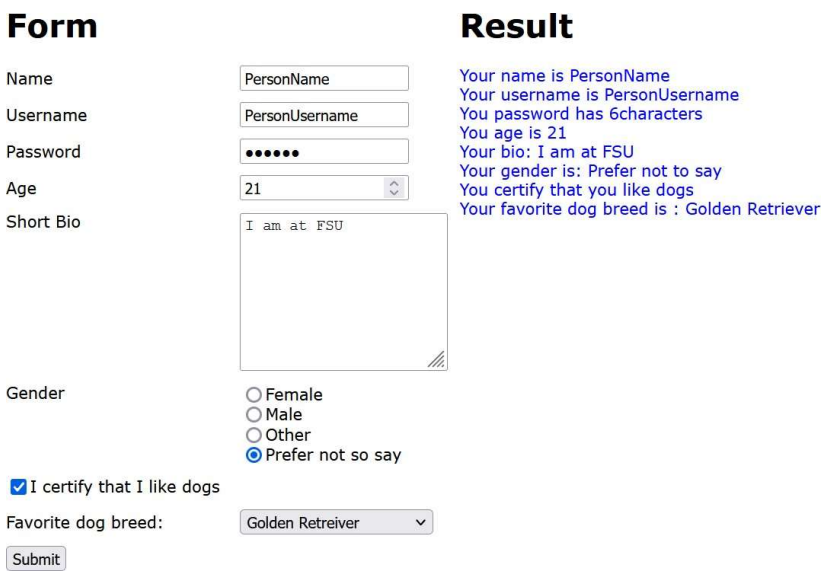
4. In case the data entered by the user does not satisfy any of the previous validation rules, **the web page must inform/alert the user with the specific rule that was not satisfied.** That is to say, generic error messages are not allowed. You can implement this feedback for the user using any strategy, but a neat way to do this is to display (in the *Result* column) information, which notifies the user of the validation rule(s) or specified requirements that the data they provided violated. The following is an example of the expected *Result* column after a user makes a failed submission: **(5 points).**




Notice that the displayed information, about the validation rule(s) or specified requirements that have been violated, is all in red font. Your solution should also display any violated validation rule(s) or specified requirements in red as well, and this applies whether you use the strategy of displaying this information about the violation in the *Result* column or you use any other strategy of your choosing (e.g., you may choose to display violation information next to a field into which invalid data has been entered, also in red, as a user enters data into the form or after a user has entered all data into the form and clicks the form submit button).

5. If the data entered by the user passed all the validation rules, the web page must display the entered information using the *Results* column. The following is an example of the expected

*Result* column after the user makes a successful submission: **(5 points).**




6. Use good programming practices **(5 points).**
    a. Your name at the beginning of each file you submit.
    b. Good indentation.
    c. Meaningful variable and method names.
    d. Code comments.

**Total points for this assignment: 50 points**

**Submission format:** Submit to Canvas a compressed file containing the modified version of the template. Make sure your submission includes all the code you wrote for this assignment.