Assign-02 : HTML Forms, Validation and Submission

Description:

This is an individual assignment. The idea behind this assignment is to get you comfortable in the creation of an HTML <form> without having to bother with the accompanying server-side stuff. The requirements are straight forward and the ASP page that you will submit your form to is being hosted on my office PC (http://pc-00224d503c23/testRequest.asp) or you can host it on your own IIS with the ASP page in the ZIP file included with this assignment.

Objectives:

- To practice using various input elements in a form element
- To practice input-level (client-side) data validation
- To practice and experiment with some basic web design and layout techniques

Requirements:

1. The name of your HTML form and page must be `register.html`
2. The title of the page is **SET Registration.**
   a. This title should be visible to the user on the page, and it should be in the heading of the HTML stream.
3. The User Interface should ask for the following information:
   a. Name   (in a textbox)
   b. Street   (in a textbox)
   c. City   (in a textbox)
   d. Province (in a dropdown list box containing all provinces and territories)
   e. Postal Code (in a textbox)
- All of these fields are required for the form to be submitted
- Check out this link for the required structure of the Canadian Postal Code. Please note that you only need to verify that *Civic Address* postal codes have been input.
4. In addition, there should be a set of option buttons (radio buttons) that allow you to choose between being Student or Faculty.
   a. If Faculty is chosen, a list of check boxes should appear that lets you select (all that apply) from
      i. Teach in Engineering
      ii. Drive Car to Work
      iii. Bring Lunch
   b. If Student is chosen, a list of check boxes should appear that lets you select (all that apply) from
      i. Study in Information Technology
      ii. Own a Laptop Computer
      iii. Take Bus to School
      iv. Own a Portable Music Player
- It is required that one of the 2 radio buttons (i.e. Student or Faculty) has been selected, but it is not required that any of the subsequent check boxes are selected

5. A *Submit* and *Cancel* button should be available
    a. If Submit is pressed, the data on the form is first validated for completeness and content format and then submitted to the response page
        i. Any errors should be identified on the form by way of user feedback and the form must not be submitted
    b. If Cancel is pressed, the form should clear and be restored to its original content.
6. A valid form should be submitted to an IIS server hosting the supplied response page, "**testrequest.asp**".
    a. To follow the practices of good web programming any and all client-side validation should be done before the form is actually submitted to the testRequest.asp page
    b. Use the *GET* method in the Form element. You will receive a response that the data was received
    c. As well, you can try the *POST* method to see what (if any) the differences are from your end.
7. The question labels and field value elements within your form must be aligned and presented in a professional manner. Spend some time ensuring that the web-form is designed in what would be considered an industry acceptable layout.
    a. There is no need to include or learn any styles (CSS or embedded) at this point
    b. You should be just ensuring that the questions within your form are structured and orderly
8. It is also expected that your HTML code and JavaScript code will be commented.

Hand in:

1. The final HTML page
2. Make sure you comment appropriately (HTML header comment, JavaScript function comment blocks as well as inline comments)
3. Make sure that your *SET Registration* application runs properly and consistently within the Internet Explorer and Chrome browsers
4. When submitting your HTML to the drop-box, please ZIP it up first