Principles of Web Development

Mini Assignment 4
Due: October 3, 2023, at 11:55 PM on myCourses

This assignment has two questions. Question 1 asks you to build a webpage implementing a <form> with JavaScript. Question 2 asks you to complete a website design plan for a specific case study. Your question 2 solution needs to be optimal.

QUESTION 1

The purpose of this question is to practice program using a <form>. Your form does not connect to a server. When the user presses submit, nothing happens. The TA will drag-drop (open the HTML file) from a local browser. The TA will be able to interact with the webpage's elements, but the page does not execute any backend code. In other words, the **action** attribute will not invoke a program.

Do not do the following:

- You are not permitted to use **Responsive design.**
- You are not permitted to use a **server.**
- You are not permitted to use Flex, Vue, or any other library.
- You are not permitted to use Bootstrap.
- You are not permitted to use Flash.
- You are not permitted to use templates.
- You are not permitted to use a database.
- You are not permitted to use backend languages, like PHP, Python, C, etc.

You are permitted to use:

- All of HTML (please use the <form> tag and <input> tags)
- All of CSS
- All of JavaScript
- Interactive elements (as described in the question)

Do the following:

1. You will create one page: `mini4.html`. The HTML page will contain a <form> that will use a POST request to a Python program.
2. The below picture shows you the webpage you are to create. Create the webpage as a <form> within the file `mini4.html`. Unlike the previous assignments**, you do not need to make it look exactly the same**, but it does need to have all the elements depicted.

   See the image below:

3. Your <form>'s **action** attribute must reference `mini4.py`. Assume the python file is in the same directory as the webpage. Do not create the Python program.
4. For the dropdown list of books, you can add any of your favourite books to the list. There should be a minimum of 3 books. The books do not need to be computer science texts.
5. Add the following interactive elements:
    a. A popup error message when the email does not have an @ symbol.
    b. A popup error message when the user presses register with one or more empty fields, then change the text color to red for the element(s) in error.
    c. If the user presses register without errors, reset the text color to black.

QUESTION 2

Read the following case study:

Super Duper University of Canada has asked you to create their student application portal. The university's website consists of a landing page describing the school and a menu with links to every portal the university offers. You have been asked to design the student application portal (http://sduc.org/registrationPortal). Students from around the world who want to enter this university will need to apply online using this portal. The portal consists of a landing page describing the registration and application process. Students must first register for a username and password, then students can apply to enter the university with their password. The portal's landing page has links to the password regisation page and to the student application form. The portal landing page is public, the link to the password registration page is public, but the link to the student application form is secure. Only people who have registered for a username/password can access the student application form. The password registration page asks for the student's full name, the username they would like to use, and the password they would like to use. A student correctly registers only when the student's full name and username are unique. If there is an error, they are returned to the registration page with a popup error message saying try again. If they register successfully, then they return to the portal landing page. At the top of the portal landing page the students sees their name and the message "Logged in". Only when their name appears and it says logged in will the link to the student application form work, otherwise it is greyed-out. If they have not logged but clicking on the student application link, a popup will display saying they need to login or register. The students can use the register page to login assuming their full name, username, and password match what is in the database. The student application page contains required and optional fields and a submit button is at the bottom of the page. The required fields are: username, password, email, department, major, high school transcript upload. The optional fields are: CV (resume) upload, recommendation letter upload. An error message popup is displayed when any of the required fields are empty. The student remains in the form and can supply the missing information. Otherwise the information is saved in a database and the student is returned to the portal's landing page. The portal landing page now displays the student's name, that they are logged in, and that they have registered.

Solve the above case study, for the portal only, by providing **an optimal solution** design. You are not writing any code. You are an engineer that has been asked to make a plan. Your plan must include the following:

- A story board with technology estimate. Remember, keep things as minimal as possible. By minimum I mean: less delay (fast), less memory (no bloat technology), inexpensive (spend only as much money as is needed and not more). Identify the ideal languages for this problem.
- Website page wireframes. There should be three wireframes.
- Website submit API signatures (URL + program).
- Website directory structure.

Write you answer in `mini4.pdf`.

FOR THE GLORY

The following GLORY challenge is for those of you brave souls who like doing extra work for no grades. BUT! you can brag to your friends about your programming skills!!

What are the design strategies behind Google's search engine? This question includes its landing page, the results page, the design elements on each page, and the search engine API + technology estimate.

WHAT TO HAND IN

- Question 1
  - The file `mini4.html`.
  - Also, the file `README.txt` stating which browser you used.
  - **Do not make the website responsive**.
  - You are permitted to use everything you have learned from class in your submissions.
- Question 2
  - The file `mini4.pdf`.

HOW IT WILL BE GRADED

This mini assignment is worth 20 points and part marks can be awarded.

Deduction points:

- -3 for not following instructions.
- Late penalty points.
- You will lose points for adding new material not covered during class.

Awarding points:

- Question 1 (10 points)
  - +2 <form> and <input> tags work correctly and as specified
  - +1 HTML source code is pretty (indentation, spacing, comments if needed, your name & ID)
  - +1 HTML page displays in a nice way (does not need to be very pretty)
  - +5 for the three interactive elements

- Question 2 (10 points)
  - +2 Story board
  - +2 Technology estimate
  - +3 Wireframes
  - +2 API Signatures
  - +1 Directory structure