Principles of Web Development

Mini Assignment 6
Due: November 7, 2023, at 11:55 PM on myCourses

This assignment continues from Mini 5's Question 1 <form> problem asking you to implement the PHP response program running on a local XAMPP server.

QUESTION

The purpose of this question is to practice processing a REST POST call from a webpage to a backend PHP program using a <form> running from your local XAMPP server. All the code (HTML and PHP) will be stored in the HTDOCS folder under XAMPP.

Do not do the following:

- You are not permitted to use the **SOCS web server**
- You are not permitted to use Vue, or any other library
- You are not permitted to use Flash
- You are not permitted to use templates
- You are not permitted to use a database

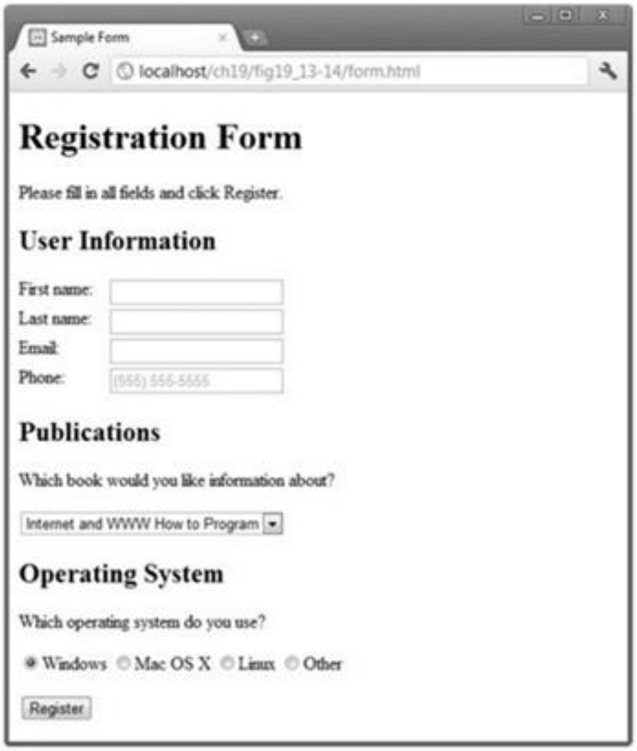
You are permitted to use:

- All of HTML
- All of CSS (including Flex)
- All of JavaScript
- All of PHP
- **XAMPP & localhost**
- **Interactive or responsive design** (optional)

Do the following:

1. Make sure your XAMPP server works. See the lecture and lab where we installed XAMPP.
2. You will create two pages: `mini6.html` and `mini6.php`. The HTML page will contain a <form> that will use a POST request to the PHP program. The PHP program will process the request and display a response to the browser. The browser will send information to the server using CGI. The PHP program on the server will send an HTML reply.
3. You can reuse the HTML code you used from Mini 5, but please included the necessary edits to the Mini 5 code to work with this assignment. The below picture shows you the webpage you are to create. Create the webpage as a <form> within the file `mini6.html`. **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:



Sample Form

localhost/ch19/fig19_13-14/form.html

Registration Form

Please fill in all fields and click Register.

User Information

First name:

Last name:

Email:

Phone: (555) 555-5555

Publications

Which book would you like information about?

Internet and WWW How to Program

Operating System

Which operating system do you use?

● Windows ○ Mac OS X ○ Linux ○ Other

Register

4. Your <form> must call `mini6.php` under your XAMPP server using localhost. **Use relative pathnames** so that it runs on the TA's computer with minimum issues.
5. Note: 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.
6. Your PHP program will do two things.
   a. Each time the PHP program is called it will **append** all the information from the form into a file called `mini6.csv`. Each "row" of the file will be formatted as a CSV (comma separated vector) record containing the input fields from the HTML <form>. **This CSV row does not need to follow all the CSV rules**. Simply add a comma between each field without any further processing. Assume there are no commas in the names, emails, phone numbers, book names, and OS names. A CSV row would look something like this:

      ```
      fname,lname,email,phone,book,os
      ```

   b. The PHP program will return a response to the browser listing **all** the CSV records from `mini6.csv`. The user's browser will render the records from the CSV file overwriting the original HTML <form> (this should happen naturally due to the nature of a synchronous call). Display the records in a <table> with each record of the CSV file as a table row <TR> and each field from the CSV as a table data cell <TD>. Print the table with alternating-colored rows. Add a table heading for the CSV column names. Make the output pretty.
7. Run your program a minimum of three times before submitting the assignment to myCourses. When you submit the HTML, PHP and CSV files to myCourses we want the CSV file to already contain multiple records from you. When the TA tests your program, they should see all your

previous appends and then the multiple appends they added when testing your code. This should produce a nice-looking table with a lot of data for the TA to see.

8. You must also handle the case of displaying an empty CSV file. This would be an important edge case. Your <table> should appear well formatted in this case as well.

fix row sizes

make table pretty

FOR THE GLORY

Redo this assignment using Node.JS and React. If you do the Glory question, you still need to complete the XAMPP / PHP version of the original question.

Remember, there are no bonus points for the glory question, but in this case especially, it would be great practice.

WHAT TO HAND IN

- The files `mini6.html`, `mini6.php` and `mini6.csv`. **ZIP all the files**.
- Also, the file `README.txt` stating:
  - Which browser you used.
  - The URL to your program running on XAMPP in case the TA needs to configure their computer.

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 (20 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)
  - +2 The HTML calls the SOCS backend PHP program properly
  - +3 PHP appends to CSV
  - +5 PHP replies to the browser as specified (pretty <table> alternating colors)
  - +3 PHP source code pretty (indentation, spacing, comments if needed, your name & ID)
  - +3 PHP output handles empty CSV edge case