

# CSC 4370/6370 WEB PROGRAMMING

## Assignment 3– PHP (2)

**Due date: 03/10/2020 (12:30 pm)**

### Instructions:

1. Create a folder named Assignment3 under your public\_html/Assignments and put your code in there and submit the link to icollege– “codd.cs.gsu.edu/~campusid/Assignment3/hw3.php”.
2. If there is no link to the codd server on icollege you would receive a “0”.
3. Your previous In class assignments, assignment 1, 2 ; project 1 and in class assignments link must be working from your “codd.cs.gsu.edu/~campusid/index.html” (Assignment 1). You will lose points if they don’t work.
4. Things to submit on icollege– Codd Link and your php code

### Question 1

- 1) Create a form that looks like image below

The image shows a web form titled "Form Validation" with the subtitle "HTML5 and server-side validation". The form contains the following fields and controls:

- Email:
- First name:
- Birthday:
- Age:
- State:
- Zip:
- 
- 
- [Reload page](#)

- 2) This exercise utilizes HTML5 client-side validation on Validate.php and PHP server-side validation on ValidateConfirm.php. The exercise uses three files:
  - a) validate.php - contains html input form and client-side validation
  - b) validateConfirm.php - php server-side validation
  - c) validationUtilities.php - validation functions used by validateConfirm.php
- 3) On clicking "Submit without HTML5 validation " all the details of the form must be displayed on a new page.
- 4) Test **Validate.php** to confirm that it works. Think of all the things you can add on client side validation. Example – Required fields etc
- 5) To validate birthday use the `IsValidDate()` function in `validationUtilities.php`.
- 6) To validate age, add a new function named `IsValidRange` in `ValidationUtilities.php`. The function will have three input parameters: `$value`, `$min`, `$max`. It should use php's `is_numeric()` function, a greater-than-or-equal test, and a less-than-or-equal test. It will be similar to the `IsValidLength()` function.
- 7) To validate zipcode, add a new function named `IsValidZipcode` to `ValidationUtilities.php`. It should use php's `is_numeric()` and `strlen()` functions (length should equal 5).
- 8) Modify `ValidationConfirm.php` to utilize the new functions.
- 9) Add a professional style to the page (Like a simple light background color and red color for validation errors). Don't add dark background colors.

## Question 2

- 1) Order01.php (<http://codd.cs.gsu.edu/~lhenry23/Testing/Order01.php>) - Order01.php –  
  
This three-page order form validates user inputs and uses cookies to store user information between pages. String length validation is used for all three parameters, which must be between 2 and 20 characters in length.
- 2) Order02.php uses both client and server validation to validate the two inputs (fname & model) from checkout01. If the inputs are not valid a message is displayed and the `exit()` statement is used to terminate execution (so that the form is not displayed). If the inputs are valid they are written to cookies and the form is displayed.
- 3) The final page, Order03.php, validates the color value from order02.php reads fname and model from the cookies, and displays the information in a message to the user. It also displays an image of the car they have selected.
- 4) There are nine car-color combinations (3 car models x 3 colors). Download images from google and use them in your code.
- 5) Provide a link back to order01.php from order03.php

## Question 3

- 1) In this part, you will create a password protected page named **protected.php**. The secured page checks for a session object to check if the user is authenticated and redirects them to the login page if they are not.
- 2) Create a form with two textboxes, a hidden field named 'postback' with value 'true', and a button. Passwords should not be passed in the query string.
- 3) The page will postback when the button is clicked. Retrieve the values for each of three form elements from step 1 and assign to variables
- 4) Repopulate the textboxes on page postback. Echo \$fname to the value field of the textbox. The syntax is something like:

```
value="<?php echo $fname ?>"
```

- 5) Validation: the sample page includes HTML5 validation but you need to add server side code as well. To prevent the server-side validation message from displaying when the page originally loads use a hidden field named 'postback' and set its value to 'true'. Below each textbox add php code similar to:

```
if ($postback && strlen($username) < 1) {  
echo "Please enter your name."; }
```

- 6) Validate inputs and create session: At the top of the page add an IF statement that checks that a username has been entered and that password is 'guest'. If both conditions are true, assign the user's name to a session object and redirect the user to protected.php. The syntax to check for a valid username and password looks something like
- 7) Hyperlink: Add a hyperlink near the bottom of the page that points to protected.php.
- 8) Finally, check that the page is using HTTPS (secure socket layers). Passwords should always use HTTPS. The first answer in Redirecting from HTTP to HTTPS with PHP (<https://stackoverflow.com/questions/5106313/redirecting-from-http-to-https-with-php>) shows how to redirect users to HTTPS. Cut-and-paste the first answer to the top of your login page.

### Steps for protected.php

1. Check if logged in: Add an if statement at the top protected.php checks for a session and if not found redirects the user to login.php. The syntax is something like:

if (!isset(\$\_SESSION["username"])) 2. Greeting: In the body of the page add a message that greets the user by name.

3. Logout button: Add a form button with the text "logout" and a hidden form field that sets "abandon" to "true."

4. Session\_unset(); Add another if statement at the top of the page that checks if the logout button has been clicked. If the posted variable "abandon" is true then use the session\_unset(); function to clear the session, then redirect the user back to the login page.

## ***Graduate Students***

In addition to the above questions, graduate students must complete the following questions as well as a part of this assignment

- 1) Write a function `countWords($str)` that takes any string of characters and finds the number of times each word occurs. You should ignore the distinction between capital and lowercase letters, and do not have to worry about dealing with characters that are not letters.  
Hint: Create an associative array mapping word keys to the number of times they occur. You will need to look at PHP's string functions to split a sentence into words. Hint 2: The `print_r($array_name)` function is useful for examining the contents of an array.
- 2) Write a function named `remove_all` that accepts a string and a character as parameters and removes all occurrences of the character. For example, the call of `remove_all("Summer is here!", 'e')` should return "Summr is hr!". Do not use the string replace function in your solution. Make this solution dynamic; i.e take the text from output through a text box and don't hard code it. Both the text and letter to be removed must be chosen via text boxes dynamically.