

Concordia University
Department of Computer Science and Software Engineering
SOEN287: Web Programming
Summer 2017
Programming assignment #3

Deadline:	Friday, Aug. 11, 2017 @ 23:59
Late submission:	Not accepted
Type of submission	<u>Electronic submission using EAS</u>
Evaluation:	10% of final mark

Important note: This is an individual or team assignment and must be submitted electronically using EAS submission system. Teams can be formed by maximum two (2) students.

Content: Server side programming with PHP

Objectives: To obtain a solid understanding of server-side scripting and how it can be used to build dynamic content and real world applications.

Coverage: Server side programming with PHP, cookies, session variables and file I/O.

Exercise 1

The following set of short questions will focus on getting you familiar with how to write PHP functions as well as how to make use of the pre-existing PHP functions. All functions and testing should appear in a single .php page.

All your PHP functions must be declared in the document `body` section and each functions name must be as specified below. To demonstrate the functionality of each method, you must

make function calls in the document body. Include a heading (h1 ... h6) that indicates which function is being tested before each function demonstration. The use of Global Variables is forbidden!

- A. **Function:** `factorial`
Parameter(s): Number
Given the number, find its factorial and return it. If the parameter is not a number, return false.
- B. **Function:** `mostFrequent`
Parameter(s): String Array
Outputs the string (element) that appears the most frequently within the array. Strings should be case insensitive.
- C. **Function:** `uppercaseFirst`
Parameter(s): String
For each word in the string, capitalize (uppercase) the first letter and return the modified string.
- D. **Function:** `splitSort`
Parameter(s): String
Given the string, place each word into an array index and sort it alphabetically before returning the filled array. Duplicated words, if any, are kept.
- E. **Function:** `dayAfterTomorrow`
Parameter(s): None
Return the date after tomorrow in the following format: DD/MM/YYYY
- F. **Function:** `findUnique`
Parameter(s): Array
Scan the entire array and print out the array excluding any duplicate elements. Example: Array a = 1,3,2,1 would output 1,3,2.
- G. **Function:** `formatHash`
Parameter(s): Associative Array
Given an associative array using full-names as keys and annual salaries as values; in a table format, print out the name and corresponding salary of each individual in increasing order (from lowest salary to the highest).

Exercise 2

Given the following URI (Uniform Resource Identifier):

`http://www.site.com/index.php?age=22&name=Obi+Wan+Kenobi&number=5435432156`

Answer the following short questions:

- 1- What is a query string and what purpose does it serve in server-side programming?
- 2- Is a query string characteristic of a GET or POST request method?
- 3- Which part of the URL is considered the query string?
- 4- How many different data parameters are being passed?

Create a single .php page (name it index.php) that, when passed the above query string as part of the URL processes the data using PHP and outputs it onto the screen in a nicely formatted table that resembles (CSS should be used in order to add some style):

Name	<i>Display Name data from query String</i>
Age	<i>Display Age data from query String</i>
Phone Number	<i>Display E-Mail data from query String</i>

The text in *italics* must be replaced with the appropriate values retrieved from the URI query string. If the query string is omitted from the URI, the above table should be replaced with the message: “No query string data found”.

Exercise 3

Create a new .php page (name it track.php) that uses cookies to track the number of times the page has been viewed. If the visitor is looking at the page for the first time, it should display the following message: *Welcome! You are new here*. If the page has been visited more than once, display: *Hello, this is your ## time here* (where ## is the number of visits count). Furthermore, if a page has been visited more than once, also display the date and time that the page was last visited on.

Exercise 4: Project

The last exercise of each assignment will be continuous from assignment to assignment and will have the goal of building an e-store website by the end of the course. (Note: since this assignment introduces PHP into the project, it is likely that some of your web pages will have to have their extensions changed from .html to .php).

Functional Registration

In the registration page you created in assignment 2, use PHP to conduct server-side validation on top of the JavaScript validation you performed in assignment 2. Validation should check that no fields are left blank, that the password fields match and that the password is at least 8 characters.

If validation on the server-side is successful, open a text file, members.txt and write all the user registration information into the file (the formatting is up to you). Multiple registrations should be possible and each new registration should be separated by a blank line.

A message should be displayed informing the user that the registration was successful or that there were errors while processing on the server.

In reality, sensitive information such as that provided by users during registration would typically be stored in some form of Database. However, Databases are beyond the scope of this course.

Functional Login & Logout

In the *side menu* area (see assignment 1) add a link labeled “Login”. When clicked, the link should direct the user to a new page: login.php. This page should include a form with two fields:

- Email Address
- Password

If both fields are filled out, PHP should be used to read the members.txt file and find a registration record that matches the email address and password entered by the user.

If the login is successful, a PHP Session should be created and used to persistently keep the user logged in throughout all pages of the website. When logged in, the “Login” link on the side menu should display “Logout” instead. Clicking the Logout link should destroy the Session and re-direct the user to the website home page.

Add to Cart

A link should be added either in the side menu or in the header that allows users who are logged in to access their shopping cart. The shopping cart page (cart.php) should display all items that have been added to the cart by the user during their session. Each items associated price should also be listed along with an overall total that should appear at the bottom of the cart page.

The “add to cart” link that you added to each of the store items in assignment 2 must now work to call upon PHP to store the item and its price in Session variable(s). The user should be brought to the cart page and given the option to either “continue shopping” – redirects the user to the page that contains the list of items, or “Checkout” – which should not do anything when clicked for the moment.

Submission Criteria

Your submission must include a README.txt file that includes the following information:

- Students Full Name
- Students ID
- Difficulties (outlining difficulties will help the TA’s focus on repairing weaknesses)
- Additional comments (anything deemed important for marking purposes)

Please give meaningful names to each html file, folder to make the feedback process easier. All files relating to each Exercise must be placed in its own folder named EX-# where # is the number corresponding to the exercise. All the exercise folders must then be placed into a single assignment folder named: *studentID-assignment3*

The assignment folder must be compressed into a zip file and submitted through EAS on the correct folder (Programming Assignment_3). For example, for the first assignment, student 123456 would submit a zip file containing all the files and folders related to this assignment named a2_123456.zip, and will upload it to EAS.

Please note that the markers reserve the right to deduct marks if the submission format is not followed as specified. Late submissions are not accepted; penalty for late submission will be 100% (Assignments submitted after the due date will receive a mark of 0). Also, email submission of assignments will not be accepted under any condition.

Academic integrity:

Students are encouraged to study and work in groups and discuss and share their knowledge with each other. However, copying is strictly prohibited and all assignments to be copied would not receive any marks. Also, those students who are found copying will face severe consequences. Students should be aware and observe the academic integrity & the university's code of conduct. For more information please refer to the course outline.

Marking schema for assignment no.3

Exercise	Description	Points
Ex. 1	2 points/each function (7 functions in total)	14
Ex.2	2 points/each question (4 questions in total) 2 points/each line in the table (3 lines in total)	14
Ex.3	“Welcome! You are new here” (2 points) “Hello, this is your ## time here” (2 points) Display the date and time (2 points)	6

Ex.4	Functional Registration	No fields are left blank (2 points), the password field match (2 points), the password is at least 8 characters (2 points), members.txt record registration info. (2 points), a messages (2 points)	10
	Login & logout	Login.php web page (2 points) Username and password validation (2 points) Persistent login status (2 points) Logout (2 points)	8
	Add to cart	Add to cart function (2 points) Cart.php display (2 points) “Continue shopping” or “Checkout” (2 points)	6
Readme file		0.5 points/each exercise	2
Total points			60

Wishing you success