Assignment 3 – JavaScript, PHP, & MySQL

# Purpose

* Solve some basic programming assignments using the JavaScript and PHP language.

# Due Date/Course Weight

This assignment must be submitted before **midnight on December 11, 2016.**

* NOTE: Late assignments will not be accepted.

This assignment is worth: **10% of your total course mark**

# Assessment

Please follow the Assignment 3 Assessment document on Blackboard for more information.

# Summary of Tasks

Listed below are the sections of the assignment which will need to be completed:

1. Develop the logic to display your web application.
2. Upload your websites for Task 1 and Task 2 to the webserver.
3. View your webpages for Task 1 and Task 2 using a web browser.
4. Submit Assignment Links on Blackboard for Task 1 and Task 2.
5. Create a ZIP file containing the entire Restaurant website (Task 2). Name the zip file **Restaurant.zip**.
6. Submit the ZIP file on Blackboard.

# Task 1- Implementing Temperature Converter using JavaScript

TemperatureConverter.html

Download ‘TemperatureConverter.html’ Template file from Blackboard (Course Content -> Assignments-> Assignment 3 -> TemperatureConverter.html) and save it to your computer. You need to write a JavaScript function inside the template file that

* Displays a prompt asking the user to select the temperature unit to be converted to (i.e., Celsius/Fahrenheit). [hint: you will need to use the prompt() Method]. The sample screenshot is as follows:

|  |
| --- |
|  |

* Displays another prompt asking the user to input the value of the temperature to be converted to Celsius/Fahrenheit. The sample screenshot is as follows:

|  |
| --- |
|  |

* The temperature value entered should be validated to ensure it is a number.
* Then, the appropriate formula should be applied to the value to convert the temperature to the selected temperature unit (i.e., From Celsius to Fahrenheit or vice versa).
* Finally, an alert should be displayed indicating the original temperature that the user entered and the converted temperature [hint: you will need to use the alert() Method to display your answer variable]. The sample screenshot is as follows:

|  |
| --- |
|  |

* The function should be called when the appropriate button (i.e., **Convert!**) is pressed.

# Task 2- Enhancing the functionalities of a Website using PHP and MySQL

# Sub-task1- Download ‘Restaurant’ Template file

Download the file ‘Assignment3-Task2.zip’ from Blackboard (Course Content -> Assignments-> Assignment 3 -> Assignment3-Task2.zip) which contains the Template folder (i.e., **Restaurant) for Task 2.**  The ‘Restaurant’ folder contains some PHP files, a css folder with the style.css stylesheet, and an image folder with no images.

# Sub-task2- Set up your database

Before we can get started using MySQL on the webserver we need a few pieces of information.

* Host
  + The host variable should contain the value “localhost”.
  + Example: $host = “localhost”;
  + NOTE: Localhost is a networking term meaning ‘this computer’
* Username
  + The Username to access YOUR database is: eatery
  + Example: $username = “eatery”;
* Password
  + The password is any legitimate password you choose cst@8238
* Database Name
  + The name of your database is: eatery
  + Example: $database = “eatery”

To create your database, username, password and tables on Hebergement Server, please review the following document on Blackboard:

Course Content -> Assignments -> Assignment 3 -> Assignment3-Task2.zip -> Instruction\_MySQL\_Hebergement.docx

To create your database, username, password and tables on EasyPHP, please review the following document on Blackboard (Only required for local testing):

Course Content -> Assignments -> Assignment 3 -> Assignment3-Task2.zip -> Instruction\_MySQL\_EasyPHP.docx

IMPORTANT: You need to modify ‘db\_config.php’ to include your own username, password, and database name.

# Sub-task3- Create a table

Now that we have a database we need to understand the tables inside it. The database must contain a table named: mailingList

Using phpMyAdmin (or the MySQL workbench, or the command line), research the MySql statement CREATE to create a new table in your ***eatery*** database.

The ‘***mailingList***’ table has the following columns:

* \_id, INT, Primary Key, NOT NULL, AUTO INCREMENT
* firstName, VARCHAR(50), NOT NULL
* lastName, VARCHAR(50), NOT NULL
* phoneNumber, VARCHAR(15), NOT NULL
* emailAddress, VARCHAR(100), NOT NULL
* userName, VARCHAR(50), NOT NULL
* referrer, VARCHAR(50), NOT NULL

**Very Important** - your table name and column names MUST match exactly or this assignment will not be marked!

# Sub-task4- Write PHP code to save the user's data to the database

Once you click ‘Contact’ navigation item from the side-menu of the website, a form will be displayed. You need to store user’s data to the ***mailingList*** table of the database entered through this form. The PHP code that saves the user's data must meet the minimum functionality requirements:

* All form fields must be validated to ensure a value is entered for each one. If a form field is left blank, no data is to be saved to the database. Duplicate entry of e-mail address into the database must be restricted. The user must be informed of which fields need to be updated.
* Once the data has been validated, an entry should be made in the database. The user's data must be put into the correct column (i.e, First Name in the firstName field etc).
* Once the data has been saved to the database, the user should be informed that the data was saved successfully. If there was an error saving the data, they should be informed of the error

# Sub-task5- Create a page that displays a table with all the customers on the ‘mailingList’ table

Create a php page called ***mailing\_list.php*** that, when called, displays an HTML table containing the customers’ data from the ***mailingList*** table.

Each row should contain the following customer information in the order they appear below. The table should contain a header row outlining the data below.

* Full Name (combine the firstname and lastname)
* Email
* Username
* Phone

There is a navigation item under ‘Contact’ side-menu, called "List". This navigation item (i.e., “List”) will link to the new ***mailing\_list.php*** script you created.

The newly created page should have common look and feel (CLF) similar to other pages of the website.

### Sub-task6 - Create an adminusers table and users

The database must contain another table named: ***adminusers***

Create a new table (***adminusers***) in your ***eatery*** database.

The ‘***adminusers***’ table has the following columns:

* AdminID INT / NOT NULL / PRIMARY KEY / AUTO\_INCREMENT
* Username VARCHAR(50)
* Password VARCHAR(50)
* AdminLevel VARCHAR(50)
* Lastlogin DATE

Again (using phpMyAdmin or the MySQL workbench, or the command line) INSERT a new user into the ***adminusers*** table.

Details of the new user of the ***adminusers*** table are as follows:

* Username = admin
* Password = passme
* AdminLevel = Administrator

**Very Important:** The username/password you create MUST match the above or again your assignment WILL NOT be marked.

### Sub-task7 - Secure the Internal Page

On Blackboard, inside Restaurant folder, there is a file named ***internal.php***. Add it to the website structure. This script should only be accessible by admin users, meaning you will need to secure this page ensuring that only users that have successfully logged in and have a valid session can view this page.

Follow the instructions below:

* Add the ***internal.php*** to the navigation calling it "Internal" and listing it under the "List" link. From the left navigation, when a user clicks on "Internal" they will be linked to the new ***internal.php*** page.
* Secure this new ***internal.php*** page/script by ensuring that the user has a valid session variable. If no session variable exists then the user will be redirected to the new ***login.php*** page.
* Create a new page called ***login.php***. This page will include the same header.php and footer.php so that it looks like every other page.
  + The new ***login.php*** page should present the user with a login form where they will enter their username and password. First, the script must validate that the username and password values are not empty. If either field is missing, a message is displayed to the user letting them know which field is missing and allows them another attempt to login again.
  + The login script will also validate the username / password supplied against the new ***adminusers*** table.
  + If the username/password (admin/passme) validates, the session variable is set. After validation, the user is redirected to the ***internal.php*** script.
  + In addition, the date/time the user logged in is to be inserted into the Lastlogin column of the ***adminusers*** table.
  + If the user supplies the incorrect username/password they will be presented with a message letting them know their username and password did not authenticate.
* Once the user is validated, the session variable should be set and the user will be allowed to access the internal.php script.
  + At the top of the new ***internal.php*** script, you should insert/display the user's session variable's AdminID, AdminLevel, and LastLogin date.
* Create a new page called ***logout.php*** to end the current session for accessing internal.php page.
* Clicking on the **LOGOUT** link on the ***internal.php*** page will terminate the current session by executing ***logout.php*** script and redirect the user to the ***login.php*** page.

N.B.: **Demo-Restaurant-Subtask7.zip** (posted on Blackboard) file will give you guideline about how to proceed with Sub-task7 of Task 2.

# Bonus-Task- Add some colors/images so that the ‘Restaurant’ website looks attractive.

# Task 3

Upload Task 1 and Task2 to a webserver. Use an FTP client to connect to the webserver.

Once you connect to the webserver using an FTP client, create a directory called ‘**Assignment3’** under **‘/public\_html/CST8238’** directory. Once your personal directory has been created navigate to that new directory. Add your Task1 file (i.e., TemperatureConverter.html) and Task2 folder (i.e., **Restaurant)** to this location.

We recommend the FTP client FileZilla. The program is provided for free (and open source for those who are interested in such things) by the Mozilla Foundation; the makers of Firefox and Thunderbird.

# Task 4

View your website using a web browser. Open a web browser and navigate to the following web address:

**http://web-server\_domain\_name/CST8238/Assignment3/<filename>**

For example, the web address to my page is:

Task1: http://rejaul.com/CST8238/Assignment3/TemperatureConverter.html

Task2: http://rejaul.com/CST8238/Assignment3/Restaurant/index.php

Where ‘rejaul.com’ is the domain name of the Web server, ‘CST8238/Assignment3’ is the name of the directory I created in the Web server under ‘/public\_html’ directory using FTP client, ‘TemperatureConverter.html’ is the web page I created for the Task1 and ‘Restaurant/index.php’ is the main (index) page of the Restaurant website.

# Task 5

Once you have confirmed that your webpage is available online, you are ready to hand in your lab.  
  
To hand in your lab go to Blackboard and navigate to Course Content 🡪 Labs and click on ‘Assignment 3 – JavaScript, PHP, MySQL’ link.

Under “Assignment Submission”, in the Submission text box write out the following Information:

* Student Number
* First Name
* Last Name
* The URLs, or hyperlinks, prepared in Task 3

Under “Assignment Submission”, submit (attach) the sourceCode (Restaurant.zip) of Task 2 (i.e., Restaurant website).

Finally, once the Submission section is complete, click the ‘Submit’ button to send the lab to your professor.

**IMPORTANT NOTE**:

If the URL, or hyperlink, does not direct the professor to the lab you will receive a ZERO for the lab assignment.

**IMPORTANT NOTE:**

You may only submit a Lab ONE TIME. Be sure the lab is complete before clicking on the ‘Submit’ button.