

CSci-4131 Internet Programing Fall 2018
Programming Assignment 8
Due Thursday December 20th at 5pm

Note: Submission after the Due Date and Time will not be accepted

This assignment is designed to introduce you to server-side programming with PHP. You will learn how to do, connect to and query a database, use sessions, and read and parse an xml file using PHP. Your task is to dynamically create your events page using PHP. At a minimum you will need to create the following pages:

- *login.php*: This page is the PHP version of your "login.html" page from previous assignments. You should handle basic input validation and provide login functionality in this page. Once the user successfully logs in, the user should be redirected to *events.php*.
- *events.php*: This page is the PHP version of your "events.html" page from previous assignments. This page will display the events table in (ascending) sorted order by event name, even date, or event time according to the query issued.
- *database.php* – this page should be included (via the include-once PHP statement) by every php page that connects to and queries the MySQL database
- *logout.php*: This page is called when the user selects the logout icon on the events page.
- *style.css*: Your CSS file(s) from previous assignments.

1. Preparation:

All files for this assignment will reside on the server. Therefore, you need to setup your account's personal web folder. Please create your web pages (.html and .php files) based on the instructions for using CSELabs to create a homepage available at the following link:

<http://cseit.umn.edu/knowledge-help/create-homepage-cs-cselabs>

The links: <http://cseit.umn.edu/knowledge-help/set-htaccess-permissions>
 <http://cseit.umn.edu/knowledge-help/learn-php>

illustrate how to create and set permissions on your .php files and your .www directory, and have a tutorial that shows you how to use PHP on the CSELabs machines

For this assignment, you will need to add the files specified below to your .www directory . You should set the permissions on your .www directory as follows: `chmod 701 .www`

Next change the permissions on your **login.php**, **events.php**, **logout.php**, **database.php**, and **dbconfig.xml** pages to **644** as follows:

```
chmod 644 login.php
chmod 644 events.php
chmod 644 logout.php
chmod 644 dbconfig.xml
chmod 644 database.php
```

As noted in CSELabs instructions, you can test your programs via the web browser by accessing the pages as follows:

<http://www-users.cselabs.umn.edu/~username/login.php>
<http://www-users.cselabs.umn.edu/~username/events.php>
and so forth.

- In the Assignment files for this assignment, you will find 1 skeleton file:
 - dbconfig.xml (dbconfig.xml contains the information regarding your database configuration – note it is the same as the skeleton file given to you in HW 7)
- Carry out the following steps:
 - Edit the file dbconfig.xml and insert the information on your MySQL database.
 - Create a file **database.php** which should parse the dbconfig.xml and create and assign the proper values to the following PHP variables
 - \$db_servername,
 - \$db_username,
 - \$db_password,
 - \$db_name,
 - \$db_port

database.php should include the PHP code (that you will write) necessary to read the file dbconfig.xml and assign the values specified in the xml file to the PHP variables listed above. Note that the variables above should be used by the login.php and events.php pages to establish a connection to the database and query the database. You should use the php statement `include_once(database.php)` in each of the aforementioned files to ensure the variables specified above are assigned values and are available for use.

You can use following PHP code to initiate a database connection:

```
<?php
//...

include once 'database.php';

// Create connection
$conn=new mysqli($db_servername,$db_username,$db_password,$db_name,$db_port);
if ( $conn->connect_error ) {

    // report error

} else {

    // setup your query
}
//...
?>
```

Setting up the Database

You will use the same database and tables as you did for Honmework assignments 6 and 7, with the same table names and columns, specifically:

tbl_accounts with the fields (columns):

acc_id, acc_name, acc_login, and acc_passwd

and

tbl_events with the fields (columns):

event_id, event_name, event_location, and event_date

We have provided the following php script files that you should edit and execute.

- i. insert_accounts_hw8.php and,
- ii. insert_events_hw8.php

Edit the files specified in i. and ii. above and add your database account information and password that you used for homework assignments 6 and 7. After editing the files, you should ensure their permissions are set correctly (`chmod filename 644`), and execute them. Be sure to login to your MySQL database to ensure that the PHP script files we provided successfully inserted the information specified in them into your database (after you edit and execute them). We will use that information as part of the grading process for your homework.

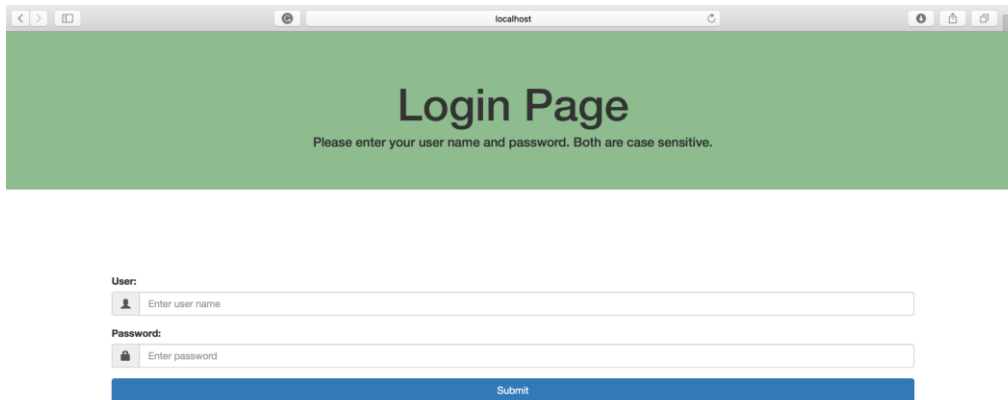
2. Functionality

Your website will have 3 pages:

- **Login** page
- **Events** page
- **Logout** page

The Events page will have a navigation bar with logout button and the logged in user display name


I. Login Page (`login.php`)



The screenshot shows a web browser window with a green header. The header contains the text "Login Page" in a large, bold font, and below it, in a smaller font, "Please enter your user name and password. Both are case sensitive." Below the header is a form with two input fields. The first field is labeled "User:" and has a placeholder text "Enter user name". The second field is labeled "Password:" and has a placeholder text "Enter password". Below these fields is a blue button labeled "Submit".

- This page should have a form with two fields: “User”, and “Password”
- Both these fields are mandatory (required – so make sure to set the required attribute)
- When the submit button is clicked, the values entered for “User” and “Password” should be sent to the server for validation before allowing further access to website.
- The server will validate the received values against the `acc_login`, and `acc_password` fields of `tbl_accounts`. Passwords is stored in the database in a SHA1 hashed format in `tbl_accounts`. Server should convert the received password string into a SHA1 hash format and compare it to the SHA1 hashed password stored in the database.
- Upon successful validation, server should
 - Create a user session (You can use `session_start()`)
 - Store user information in the session variable `$_SESSION`.
- If the validation fails, server should redirect the user back to the `login.php`: Note, your program does not need to indicate that any errors occurred in this case.
- If the login is successful, user should be routed to `events.php` page.


II. Event Page (events.php)

Events Page 

Event Name	Location	Date
CSCI 3123	Keller Hall 320	00:00:00
Meeting	Shepherd Labs 587	01:00:00
Bowling	Coffman	02:00:00
CSCI 2034	Keller Hall 322	03:00:00
Group Meeting	Walter Library	04:00:00


☒ event name ☐ event location ☐ event date

- If a user tries to access this page without a valid login, (the login information is not set, which you can check with the PHP function `isset`) the user should be routed to “**login.php**” page.
- The page should have a navigation bar with logout button.
- The page should display the logged in user name.
- The table in this page should be dynamically populated.
- To achieve this, the server should query from the following table: `tbl_events`.
- A group of radio buttons are provided for ordering events according to various columns, i.e., event name, location and date. Upon clicking the radio button options, the table should display the events in the order selected.

Events Page 

Event Name	Location	Date
Bowling	Coffman	02:00:00
CSCI 2034	Keller Hall 322	03:00:00
CSCI 3123	Keller Hall 320	00:00:00
Group Meeting	Walter Library	04:00:00
Meeting	Shepherd Labs 587	01:00:00

☒ event name ☐ event location ☐ event date

Events Page 

Event Name	Location	Date
Bowling	Coffman	02:00:00
CSCI 3123	Keller Hall 320	00:00:00
CSCI 2034	Keller Hall 322	03:00:00
Meeting	Shepherd Labs 587	01:00:00
Group Meeting	Walter Library	04:00:00

☐ event name ☒ event location ☐ event date

III. Logout Page (logout.php)

Upon clicking the logout button, the user session should be destroyed and user should be routed to the **Login** page (you can use the PHP statement: `header('Location:login.php');` to accomplish this.

3. Submission Instructions

PLEASE ENSURE TO TEST YOUR CODE ON CSE LAB MACHINES.

You will need to submit all the files used to develop your website. This includes all the HTML, CSS, JavaScript (if you used) and other files if any.

Towards this end, make a copy of your working directory and re-name it: **yourx500id_hw08**.

Compress the **yourx500id_hw08** directory and submit it.

We will use the `acc_login` and `acc_password` values (set by the script we provided for you to edit and run to insert values into `tbl_accounts` table) to login to your website. Ensure that these values are present in your `tbl_accounts` table and can be used to access your website.

IV. Evaluation

Your submission will be graded out of 100 points on the following items

- Submission instructions (including running the PHP provided in section on setting up the database) are not met. **(-10 points)**

15 points

- `database.php` file should parse the `dbconfig.xml` file and then create and assign values to the PHP variables specified used to establish a connection to the database and query the database. **(15 points)**

35 points (Login page)

- Events page of your website should redirect the user to Login page without authentication. **(5 points)**
- The "Login" page shows the form elements and submit button. **(5 points)**
- If wrong log in credentials are entered, the user should be redirected to the same login page. **(5 points)**
- After successful validation on server, the "Login" page redirects the user to "Events" page. **(10 points)**
- The logout functionality works correctly and redirects the user to "Login" page. **(10 points)**

35 points (Events page)

- "Events" page displays. **(5 points)**
- "Events" page displays the radio button group **(5 points)**
- "Events" page gets the list of events from server. The events are dynamically added to the table displayed by the page. **(15 points)**
- "Events" page displays the list of all events places in the correct order upon clicking option in the radio button. **(10 points)**

15 points (Logout Page)

- Logout page is called correctly by logout icon on the events page **(5 points)**
- Logout page ends (destroys the session) and redirects control to the login page, which is displayed correctly and functions correctly **(10 points)**