CSci-4131 Internet Programing
Spring 2019
Homework Programming Assignment 8

Due Date: Friday May 10th at 3pm

Submissions after the Due Date and Time will not be accepted

Version 2, Issued May 1st, updates are in red font.

This assignment is intended to introduce you to server-side programming with PHP. You will develop a full-stack application that employs session handling to facilitate a login capability, connect to and query the database same database we have been using in previous assignments, and, as with previous assignments, use the SHAH 256 hashing algorithm to both store a password for a login and verify the password a user enters against the password stored in the database when they log in. This assignment specification contains 8 pages.

Your task is to dynamically create functional login and events pages in PHP (you can create more pages should you require them to create a solution to the assignment). The pages will be served by the Web Server on the CSE Labs machines from your personal home page area.

***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 will use PHP, HTML, JavaScript to build the "events.html" page from previous assignments. When a browser displays this your events you have created and add functionality to query the database for information from the table.

***style.css*:** Your CSS file (same from previous assignments.)

1. Preparation:

Unlike previous assignments, the files for this assignment must reside in the .www directory you will set up for your homepages on the CSE Labs machines. Therefore, you need to setup your personal web folder in your CSELabs home directory (which is the directory you are in after logging into a CSELabs machine with your x.500 acount and password). Please create your .www directory and web pages (.html or .php files) based on the instructions for using CSELabs to create a homepage and php files available at the following links:

https://cseit.umn.edu/knowledge-help/homepages-cs-cselabs

The link above shows you how to create the directory for your personal webspace (in which you can create, store and access your personal web pages) on the CSE Labs machines.

The link:

https://cseit.umn.edu/knowledge-help/php-tutorial

illustrates and has a short tutorial on how to create and set permissions on your .php files in your .www directory on the CSE Labs machines.

For this assignment, you will need to create the **login.php** and **events.php** files specified above and the additional files we've provided to your .www folder. You can create any additional files you deem necessary as well (for example **logout.php** or others) In any case, REMEMBER TO CHANGE THE PERMISSONS ON FILES IN YOUR .www directory to .www to **644**.

This can be accomplished as follows (the % is the unix/linux command line prompt) for the **login.php** and **events.php** pages**:**

```
% chmod 644 login.php
% chmod 644 events.php
```

As specified in CSELabs PHP tutorial , you can test your programs via a web browser by entering the following URL's in the browsers address bar:

```
http://www-users.cselabs.umn.edu/~username/login.php
http://www-users.cselabs.umn.edu/~username/events.php
```

- In your .www directory, you should create a file ***database.php*** (with permissions set to 644) in which you set the following PHP variables to your database credentials, for example

```
          $db_servername = "cse-curly.cse.umn.edu";
          $db_username = "C4131S19GXXX";
          $db_password = "XXX";
          $db_name = "C4131S19GXXX";
          $db_port=3306;
```

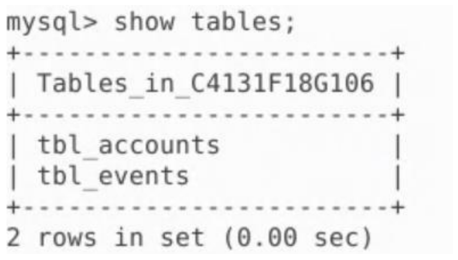
Please note that the names of the variables must be exactly same as specified above. The file ***database.php*** is a common code segment used by the four scripts we have provided, and the values of the variables are used to establish the database connection. You should also include it in your **login.php** and **events.php** pages (via the **`include`** or **`include-once`** PHP directive) and use the values stored in the variables to connect to your MySQL database.

Note that the database tables you will use in this assignment are the same tables you created and used in your previous express/ node.js assignments. You can check the state of your database as follows:

- The database required for this assignment contains two tables*: tbl_accounts and tbl_events*
  You can check to ensure that the database tables are created and populated correctly using the *mysql* console as follows:.
  - Log in to your database via the unix command line (replacing the **xxx** with the credentials we provided for you for your node.js/express homework assignments) as shown below:

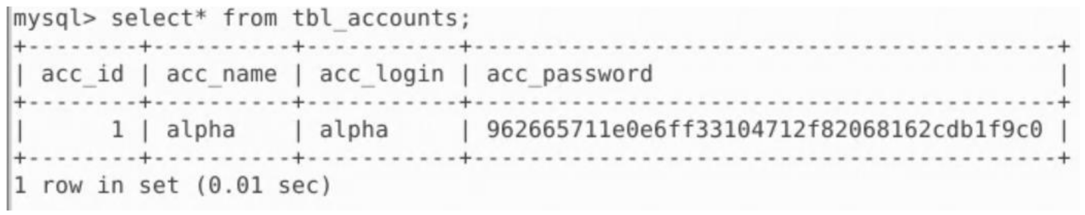
    ```
    % mysql -uC4131S19Gxxx -hcse-curly.cse.umn.edu -P3306 -p C4131S19Gxxx
    ```

  - You will be prompted for your password (use the numeric password we provided for you), enter it and press the enter key
  - Then execute the sequence of commands shown in the pictures below to ensure that your database tables exist and are populated with data you can use for this assignment. (Note, your tables may contain different data than the data displayed on the following page since the data displayed in the example is the data from our solution)

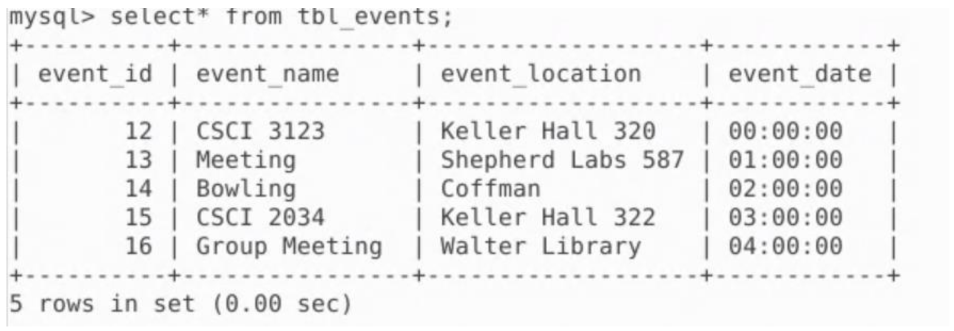


```
mysql> show tables;
+------------------------+
| Tables_in_C4131F18G106 |
+------------------------+
| tbl_accounts           |
| tbl_events             |
+------------------------+
2 rows in set (0.00 sec)
```

Display the elements in the tables using queries;



```
mysql> select* from tbl_accounts;
+--------+----------+-----------+------------------------------------------+
| acc_id | acc_name | acc_login | acc_password                             |
+--------+----------+-----------+------------------------------------------+
|      1 | alpha    | alpha     | 962665711e0e6ff33104712f82068162cdb1f9c0 |
+--------+----------+-----------+------------------------------------------+
1 row in set (0.01 sec)
```



```
mysql> select* from tbl_events;
+----------+---------------+-----------------+------------+
| event_id | event_name    | event_location  | event_date |
+----------+---------------+-----------------+------------+
|       12 | CSCI 3123     | Keller Hall 320 | 00:00:00   |
|       13 | Meeting       | Shepherd Labs 587 | 01:00:00 |
|       14 | Bowling       | Coffman         | 02:00:00   |
|       15 | CSCI 2034     | Keller Hall 322 | 03:00:00   |
|       16 | Group Meeting | Walter Library  | 04:00:00   |
+----------+---------------+-----------------+------------+
5 rows in set (0.00 sec)
```

If you need to recreate or add data to your tables, on Moodle, in the HW8 directory, you will find the following 4 skeleton files:

- Create_accounts_table.php;
- Create_events_table.php
- Insert_accounts.php
- Insert_events.php

- Download these files to your **.www** directory and set the permissions on each of the files to 644.

- Should you need to run the scripts we have provided, enter the URL of the script you need to execute in your browsers URL (address bar). For example:

  http://www-users.cselabs.umn.edu/~liux4189/create_accounts_table.php
  http://www-users.cselabs.umn.edu/~liux4189/create_events_table.php
  http://www-users.cselabs.umn.edu/~liux4189/insert_accounts.php
  or
  http://www-users.cselabs.umn.edu/~liux4189/insert_events.php

Note that the scripts need to be executed by a Webserver via a request from the browser as shown in the example on the top of the next page in the browser URL (address) bar.

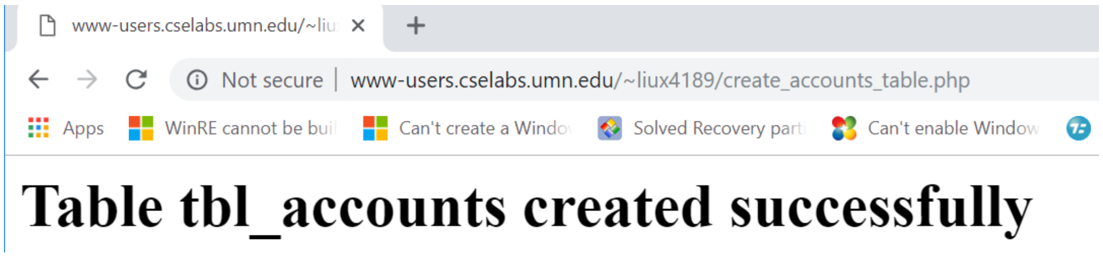

www-users.cselabs.umn.edu/~liu
Not secure | www-users.cselabs.umn.edu/~liux4189/create_accounts_table.php
Apps
WinRE cannot be bui
Can't create a Windo
Solved Recovery part
Can't enable Window

Table tbl_accounts created successfully

If the script executes successfully, a success message (similar to the one shown above) will be displayed by your browser.

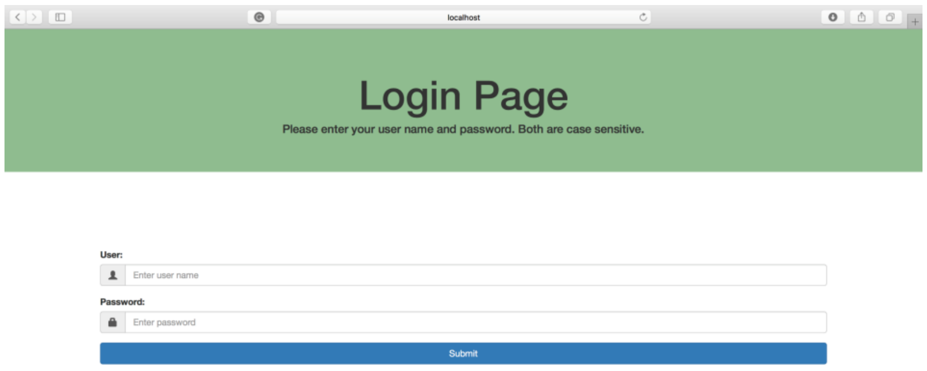
2. Functionality

Your website should have at least 2 .php pages:

I. **A Login** page (e.g., login.php)
II. **An Events** page (e.g, events.php), and
III. **Implement logout functionality**

The Events page will should have a navigation bar with a logout button and display the name of the user who has successfully logged in. The following pages specify the functionality REQUIRED in more detail

i. Login Page (login.php)



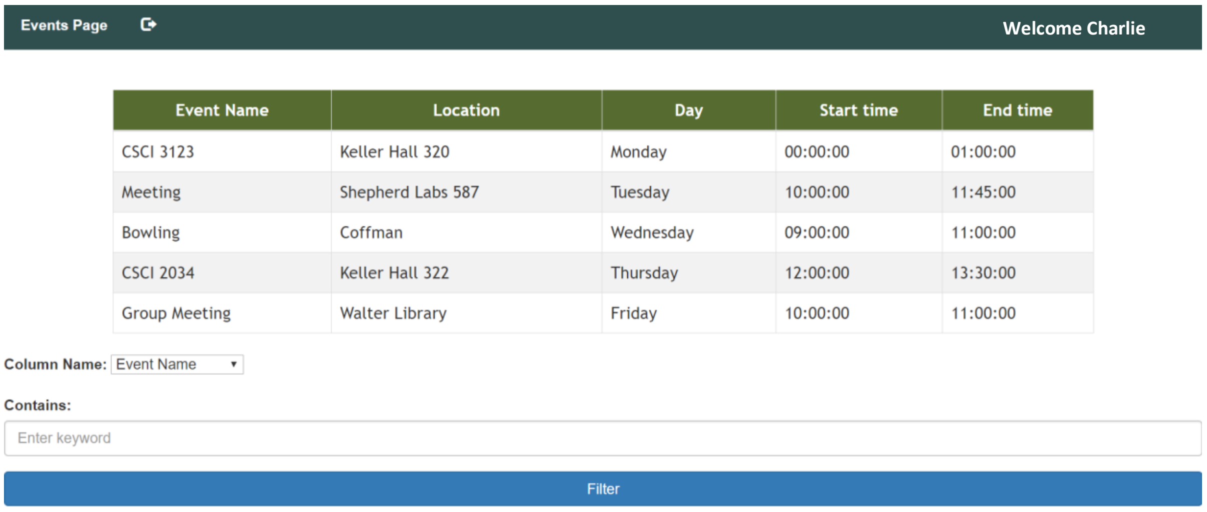
- This page should have a form with two fields: “User”, and “Password”
- Both these fields are mandatory.
- When the submit button is clicked, the values entered for “User” and “Password” should be sent to the server for validation before allowing further access to website.
- The server will validate the received values against the *acc_login*, and *acc_password* fields of *tbl_accounts*. Passwords should be stored in the database in a SHA256 hashed format in tbl_accounts. Your login page should convert the password string entered by the user into a SHA256 hashed password format and compare it to the SHA256 hashed password stored in the database.
- To match hashed passwords that you created and stored in your tbl_accounts with your node.js sha256 hashing function, you must encode in base 64. For example, the statement:

```
        $hashedPW = base64_encode(hash("sha256",$password,True));
```

  will create a sha256 hashed password encoded in base 64 using the value stored in the PHP variable **`$password`** and store it in the PHP variable **`$hashedPW`**.

- Upon successful validation, your login page should
  - Create a user session (You can use *session_start() at the top of the page*)
  - Store user information (login and password) in the session variable **$_SESSION**.
- If the user and password validation fails, your login page should redirect the user back to the login.php (that is, itself). Note, you do not need to display an error if validation fails
- If the login is successful, user should be routed to *events.php* page.

ii. Events Page (events.php)



Event Name	Location	Day	Start time	End time
CSCI 3123	Keller Hall 320	Monday	00:00:00	01:00:00
Meeting	Shepherd Labs 587	Tuesday	10:00:00	11:45:00
Bowling	Coffman	Wednesday	09:00:00	11:00:00
CSCI 2034	Keller Hall 322	Thursday	12:00:00	13:30:00
Group Meeting	Walter Library	Friday	10:00:00	11:00:00

- If a user tries to access the events page without a valid login or a valid password, the user should be routed to the "**login.php**" page.
- The page should have a navigation bar with a logout button.
- The page should display the name of the user who is logged in.
- The table in this page should be dynamically populated, then displayed.
- To achieve this, the page should query the database to obtain the information used to populate the table from the following database table: **tbl_events**.
- A Filter button should be provided to query the database by submitting the data with the user-entered selections for "Column Name" and "Contains" via a GET or POST request to the server when it is "clicked".
- Filter Criteria:
  - This form has two fields: **Column Name** and **Contains**.

    Column Name is a dropdown list consisting of names all the columns in the table (i.e., Event Name, Location, Day, Start time and End time.)

    Contains is a textbox for users to enter the keyword that they are searching for in the column.
  - Based on the column name the user selects from the dropdown list and the value the user enters in the textbox, the page should display the filtered content obtained from a query to the database as follows:
    - After entering text in the textbox and clicking the filter button, the page should display all the data from each row which contains text in the corresponding column.
    - If the user clicks the filter button without entering values in the textbox, the page should display all the content from each row stored in the database table.

See section 6 (The appendix) for an additional pictorial description functionality of the events page)

iii. Logout

Upon clicking the logout button, the user session should be destroyed, and user should be routed to the **login** page. (You can create a separate PHP file for this if you determine it is the best way for you to implement this (logout) functionality).

3. Additional Information

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
}
//…
?>
```

Also, you should use the PHP "`trim`" command to trim the data your php pages obtain from for fields

4. Submission Instructions

PLEASE ENSURE TO TEST YOUR CODE ON CSE LAB MACHINES.
You will need to submit all the files used to develop the 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: **yourx500id_hw08.** Rename the copied folder as **yourx500id_express**.

Create a README file inside yourx500id_express directory. This README file should include: Your x500id, acc_login, and acc_password values from insert_into_accounts_table.js file.

Compress the yourx**500id_express** directory and submit it**.**

NOTE: We will use the acc_login and acc_password values to login to your website. Ensure that these values are correct and can be used to access your website.

5 Evaluation Criteria

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

15 points

- Submission instructions are met. **(5 points)**
- Your solution should include and use the **database.php** file to load the necessary variables and establish the database connection **(10 points)**

25 points (Login page)

- Events page of your website should redirect the user to Login page if a user who attempts to access the page is not logged in. **(5 points)**
- The "Login" page shows the form elements and submit button. **(5 points)**
- If incorrect log in credentials are entered, the user should be redirected to the same login page. **(5 points)**
- After successful validation of a login and password entered by a user, the "Login" page should redirect the user to "Events" page. **(10 points)**

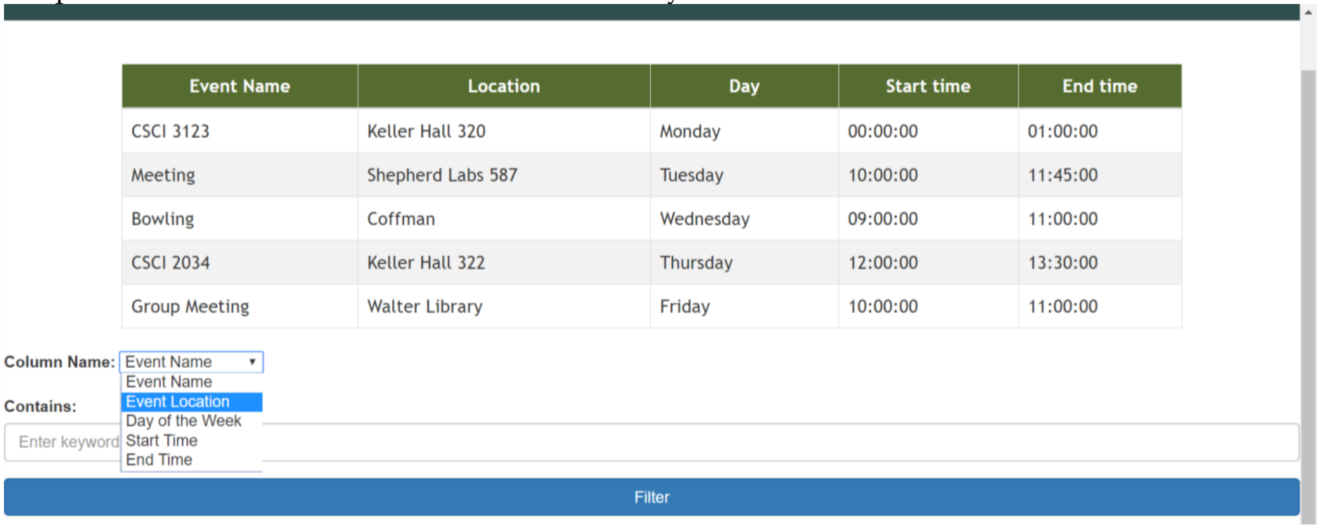
60 points (Events page and logout functionality)

- The logout functionality works correctly and redirects the user to "Login" page. **(10 points)**
- "Events" page correctly displays the events in the table and the navigation capability functions correctly. **(5 points)**
- “Events” page correctly displays the user name of the user who has logged in **(5 points)**
- “Events” page correctly displays the dropdown list and text box used to get selection criteria (i.e., fields to enter values to “filter” the data stored in the tbl_accounts table to obtain the information that the user wants to display) **(5 points)**
- "Events" page gets the list of events from the database. The events are dynamically added to a table and correctly displayed by the page. **(20 points)**
- "Events" page correctly displays the list of all events that match the filtering criteria upon clicking the filter button. **(15 points)**

ALSO – 10 bonus points for Submitting your final version of the assignment by Saturday, May 4th at 11:55pm

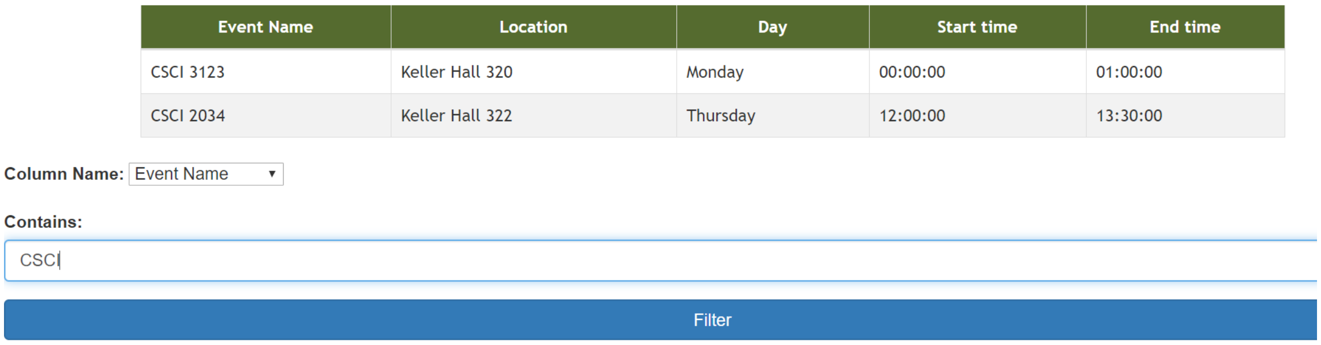
6 Appendix

Dropdown list – should enable the user to select any field from the table



Event Name	Location	Day	Start time	End time
CSCI 3123	Keller Hall 320	Monday	00:00:00	01:00:00
Meeting	Shepherd Labs 587	Tuesday	10:00:00	11:45:00
Bowling	Coffman	Wednesday	09:00:00	11:00:00
CSCI 2034	Keller Hall 322	Thursday	12:00:00	13:30:00
Group Meeting	Walter Library	Friday	10:00:00	11:00:00

Filter by Event Name – displays the result set of ALL ELEMENTS IN EACH ROW that contain the keyword entered in the text box (rows are selected using the field name from the drop-down list box and the string entered in the text box labelled "Contains".
NOTE, if no keyword is entered in the text box, ALL THE DATA FROM EACH ROW STORED IN THE TABLE **`tbl_events`** SHOULD BE DISPLAYED.



Event Name	Location	Day	Start time	End time
CSCI 3123	Keller Hall 320	Monday	00:00:00	01:00:00
CSCI 2034	Keller Hall 322	Thursday	12:00:00	13:30:00

Column Name: Event Name

Contains: CSCI