CSci-4131 Internet Programing Spring 2020
Project Assignment for Final Project Options 3, 4, and 5

Due Sunday May 12th at 6:30 pm
No Late Submissions Accepted

For this final project assignment, you will do to server-side programming (scripting) with PHP. You can review all project options at the top of the class Canvas Website in the Resources Module. For this project option, you will implement PHP sessions to handle login/authentication, query the MySQL database using PHP, and parse XML in PHP. Your task is to dynamically create your login and contacts pages using PHP. You will need 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 *contacts.php*.
- *contacts.php*: This page is the PHP version of your "myContacts.html" page from previous assignments. You will basically display your contacts you have created.
- *style.css*: Your CSS file which is the same from previous assignments.

1. Preparation:

Unlike previous assignments, the PHP scripts (files) for this assignment reside on the server AND the Web pages displayed in your browser are dynamically updated by your PHP scripts when the are executed by the Apache Web Server running on the CSE Labs machines. That is, your PHP scripts (pages) receive information from your browser via the Apache Web Server, are executed by the Apache Server, create a Web Page, and the Apache server sends the resulting page to your browser. **So, for this assignment, you need to setup your account's personal web folder on the CSE Labs machines. Please create web pages (.html or .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`

explain how to create and set permissions on your .php files, and your ***.www*** directory. The links also 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 above and the additional files provided to your home directory (or. www directory/folder). ***Remember, you must change the permission of all your files that you use or create in your .www directory for this assignment to 644. You can do this as follows for the*** **login.php** ***and*** **contacts.php** ***pages:***

```
chmod 644 login.php
chmod 644 contacts.php
```

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

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

- In the files we provide for for this assignment, you will find 5 skeleton files:
  - `Create_accounts_table.php;`
  - `Create_contacts_table.php`
  - `Insert_accounts.php`
  - `Insert_contacts.php`
  - `dbconfig.xml`

First, you should edit the file `dbconfig.xml` to contain the information regarding your database configuration as follows:

- Carry out the following steps:
  - Edit the `dbconfig.xml` to provide the details of your own database.
  - You need create a file ***database.php*** which should load and parse the file `dbconfig.xml` using an xml parser (we use the `simplexml_load_file` PHP command - and create the following variables, each of which is assigned the corresponding data present in your `dbconfig.xml` file):
    - `$db_servername,`
    - `$db_username,`
    - `$db_password,`
    - `$db_name`

Please note that the names of the variables in your file must be exactly same as those specified above.

The PHP script in ***database.php*** will be the common code segment shared by four scripts in order to create the variables for data connection. You will use the PHP command:

```
include_once 'database.php';
```

on each PHP page/file which opens a connection to, queries, and closes the connection to your MySQL database

.

Note that the MySQL database for this assignment is the same as the node.js Homework 6 and Homework 7 assignments, and will use the same tables, and the data in those tables – if they still exist (that is, you haven't deleted them).

You can double check the status of your MySOL database tables and the data in them as follows:

- **First, recall, the Database in this assignment contains the same two tables as you created and used for Homework Assignment 6 and Homework Assignment 7*: tbl_accounts and tbl_contacts***

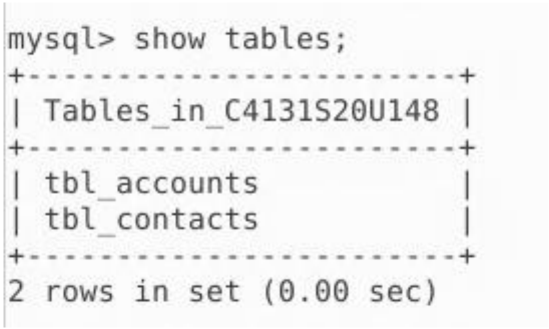
If the database tables in your MySQL database already exist and contain data , you can manually login to the database, login, and check to make sure your database tables and data is present using the mysql console.

- Log in (replace placeholders i.e., xxx with the username and password provided.

mysql -uC4131S20Uxxx -hcse-larry.cse.umn.edu -P3306 -pxxx

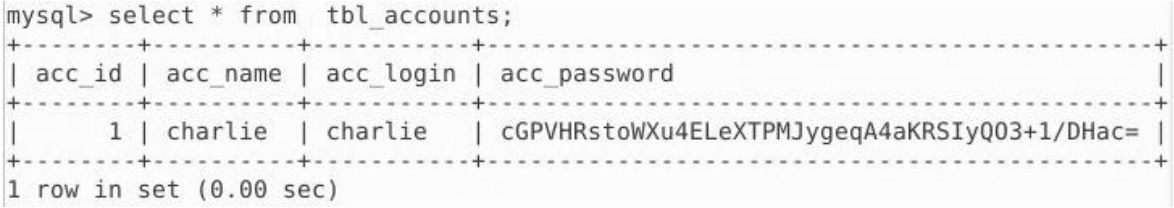
- Select database and check the success of the table creation using the following commands.

use C4131S20Uxxx;
show tables;



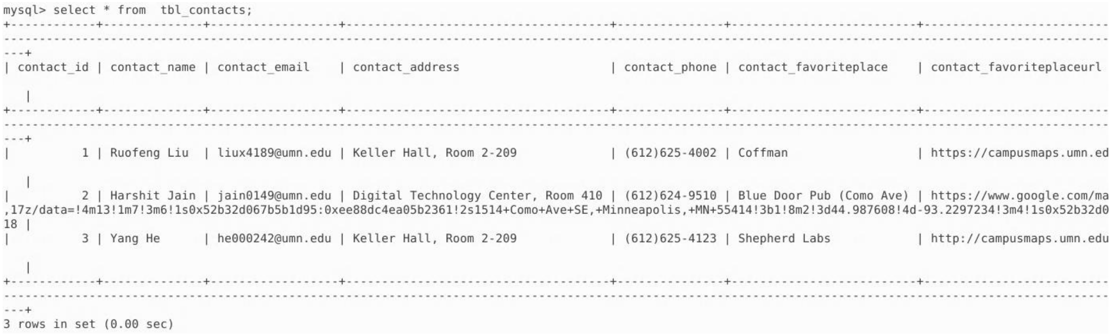
```
mysql> show tables;
+------------------------+
| Tables_in_C4131S20U148 |
+------------------------+
| tbl_accounts           |
| tbl_contacts           |
+------------------------+
2 rows in set (0.00 sec)
```

- To display the data in **`tbl_accounts`** table, execute the query: **select * from tbl_accounts (as in the EXAMPLE shown below – which displays the data in OUR tbl_accounts, table - *not yours!*):**



```
mysql> select * from  tbl_accounts;
+--------+----------+-----------+----------------------------------------------+
| acc_id | acc_name | acc_login | acc_password                                 |
+--------+----------+-----------+----------------------------------------------+
|      1 | charlie  | charlie   | cGPVHRstoWXu4ELeXTPMJygeqA4aKRSIyQO3+1/DHac= |
+--------+----------+-----------+----------------------------------------------+
1 row in set (0.00 sec)
```

- To display the data in **`tbl_contacts`** table, execute the query: **select * from tbl_contacts (as in the EXAMPLE shown below – which displays the data in OUR tbl_contacts table, - *not yours!*):**



```
mysql> select * from  tbl_contacts;
+------------+--------------+------------------+-------------------------------------+---------------+--------------------------+-------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
---+
| contact_id | contact_name | contact_email    | contact_address                     | contact_phone | contact_favoriteplace    | contact_favoriteplaceurl

   |
+------------+--------------+------------------+-------------------------------------+---------------+--------------------------+-------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
---+
|          1 | Ruofeng Liu  | liux4189@umn.edu | Keller Hall, Room 2-209             | (612)625-4002 | Coffman                  | https://campusmaps.umn.ed

   |
|          2 | Harshit Jain | jain0149@umn.edu | Digital Technology Center, Room 410 | (612)624-9510 | Blue Door Pub (Como Ave) | https://www.google.com/ma
,17z/data=!4m13!1m7!3m6!1s0x52b32d067b5b1d95:0xee88dc4ea05b2361!2s1514+Como+Ave+SE,+Minneapolis,+MN+55414!3b1!8m2!3d44.987608!4d-93.2297234!3m4!1s0x52b32d0
18 |
|          3 | Yang He      | he000242@umn.edu | Keller Hall, Room 2-209             | (612)625-4123 | Shepherd Labs            | http://campusmaps.umn.edu

   |
+------------+--------------+------------------+-------------------------------------+---------------+--------------------------+-------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
---+
3 rows in set (0.00 sec)
```

If, for some reason, your database (and data) from Homework 6 and 7 no longer exists, you can edit and run the scripts we have provided to re-create and re-populate them as follows:

Similar to the .js files provided (e.g., create_accounts_table.js) in the HW6 and HW7, the PHP scripts we provided with this project option can be used to create and populate your MySQL database.

- Download these files to your .www directory
- Carry out the following steps
  - Edit each of the 4 PHP script files include the details of your own MySQL database.
  - Then run each of the 4 scripts in order to create required tables and to populate some entries in the tables.

    http://www-users.cselabs.umn.edu/~username/create_accounts_table.php

    http://www-users.cselabs.umn.edu/~username/create_contacts_table.php

    http://www-users.cselabs.umn.edu/~username/insert_accounts.php

    http://www-users.cselabs.umn.edu/~username/insert_contacts.php

Please Note that the scripts need to be triggered via the request from the browser in a manner similar to the example on the next page, and a success message will show up if the database operations are executed successfully.

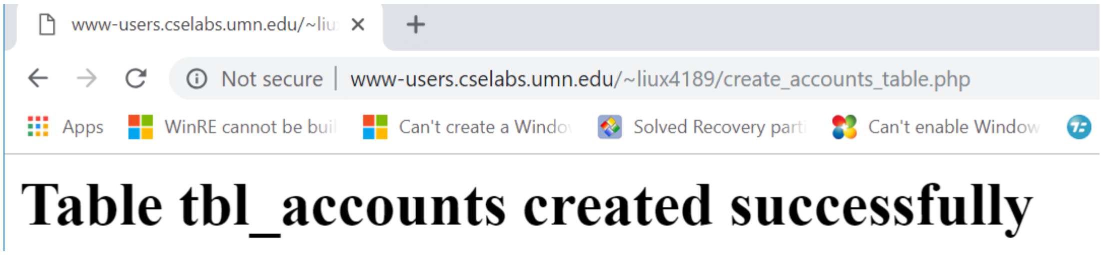


Table tbl_accounts created successfully

For this assignment, in your `login.php` and `contacts.php` pages, you can use following PHP code to initiate a database connection:

```
<?php
//… PHP script that preceeds PHP script to include your MySQL database info

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
}
//… PHP script that follows the PHP script used to connect to your MySQL database
```

The following pages describe the functionality required for this project assignment.

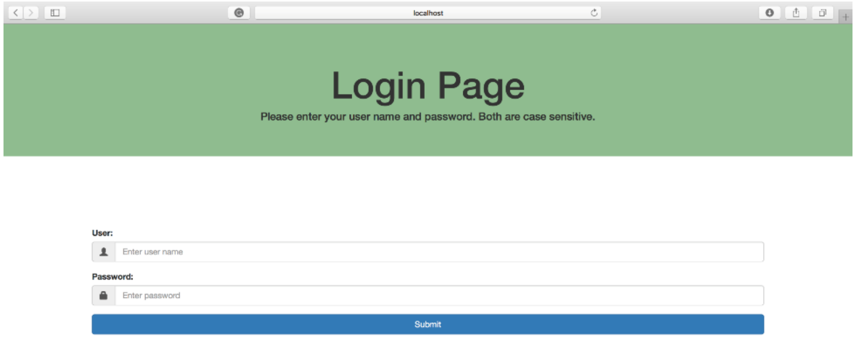
2. Functionality

Your PHP website should have the following 2 pages:

- **Login** page
- **Contacts** page

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

1. Login Page (name: login.php)



Login Page

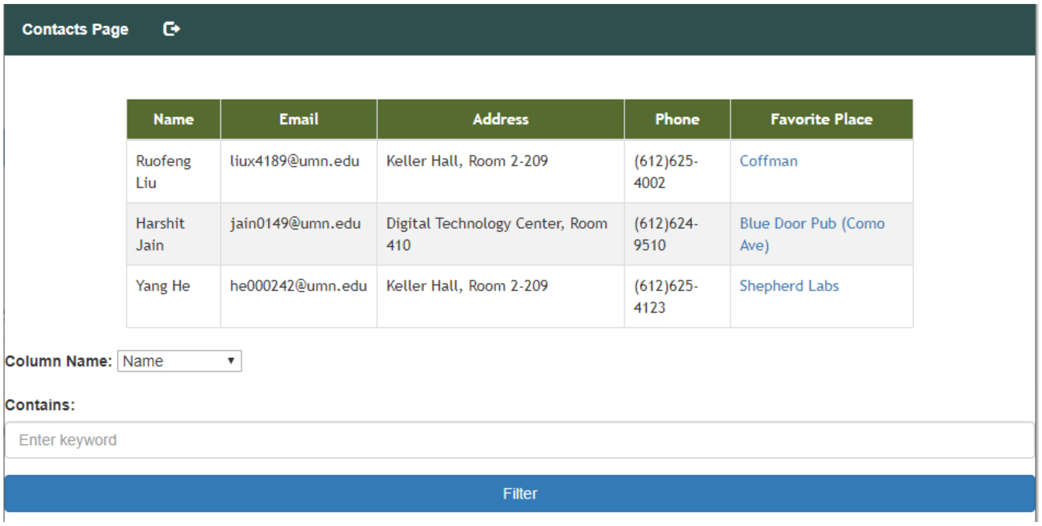
Please enter your user name and password. Both are case sensitive.

User: Enter user name

Password: Enter password

- This page should display an HTML 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 (columns) of the *tbl_accounts* table. As with Homework Assignments 6 and 7, passwords are stored in the database in a SHA256 hashed format in *tbl_accounts.* The PHP page server should convert the received password string received via the global POST array into a SHA256 hash format and compare it to the SHA256 hashed password stored in the database.
- Upon successful validation/authentication, your PHP script should
  - Create a user session (You can use *session_start()*)
  - Store user information in the PHP session variable named *$_SESSION.*
- If the login is successful, user should be routed (redirected) to *contacts.php* page.
- If the validation fails, the PHP script should redirect the user back to the PHP page: *login.php*.

2. Contacts Page (name: contacts.php)



Contacts Page

Name	Email	Address	Phone	Favorite Place
Ruofeng Liu	liux4189@umn.edu	Keller Hall, Room 2-209	(612)625-4002	Coffman
Harshit Jain	jain0149@umn.edu	Digital Technology Center, Room 410	(612)624-9510	Blue Door Pub (Como Ave)
Yang He	he000242@umn.edu	Keller Hall, Room 2-209	(612)625-4123	Shepherd Labs

Column Name: Name ▾

Contains:

Enter keyword

- If a user tries to access this page without a valid login, the user should be routed to "**login.php**" page.
- The page should have a navigation bar with logout button.
- The table in this page should be dynamically populated.
- To achieve this, your PHP page should query your MySQL database from the following table: `tbl_accounts`.
- Your PHP script provides a form for filtering contacts according to various columns, i.e., name, email, address, phone and favorite place.
- Filter functionality: **SEE THE APPENDIX (SECTION 5) on PAGE 8 for a illustration of the functionality described below:**

  ○ The form your page uses to filter data has two fields: Column Name and Keyword.

    ➢ Column Name is a dropdown list consisting of names all the columns in the table, that is: name, email, address, phone and favorite place.

    ➢ Keyword is a textbox for users to enter a keyword to search for in the column name selected.

  ○ Based on the column name a user selects from the dropdown list, and the keywork value a user enters in the textbox, your PHP page should obtain and create the HTML to display the filtered content (after clicking the filter/submit button) as follows:

    ■ After selecting a column name and entering a keyword in the textbox, when the filter/submit button is clicked, your PHP script should create a query to obtain all the contacts which contain the keyword text in corresponding column, and then create the HTML necessary to display the results.

    ■ If the user clicks the filter button without entering values in the textbox, the PHP page should query t the database for all the information in the table and create the HTML necessary to display all the content from the table.

3. Logout

Upon clicking the logout button, the user session should be destroyed, and user should be redirected to **login** page.

3. 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 PHP, HTML, CSS, JavaScript (if you used) and other files if any.

Towards this end, make a copy of your working directory: **yourx500id_PHP.** Rename the copied folder as **yourx500id_PHP**.

Create a README file inside **yourx500id_PHP** directory. This README file you create should include:

- Your x500id,
- login and password combinations for your tbl_accounts table in your MySQL database.
- the project option you have selected (3, 4 or 5)

See the document named: Csci 4131 Final Project Options.pdf in the Resources Module top of the Class Canvas website for a specification of the project options and how they count to your final project grade.

Compress the **yourx500id_PHP** directory and submit it**.**

We will use the login and password values you provide to login to your website. Ensure that these values are correct and can be used to access your website, or we will be unable to grade your assignment, and you will be assigned a zero for the grade.

4. Evaluation

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

15 points

- Submission instructions are met. **(5 points)**
- `database.php` PHP file should parse dbconfig.xml file to load the necessary variables and the database connection is established **(10 points)**

25 points (Login page)

- Contacts 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 login validation/authentication, the "Login" page redirects the user to "Contacts" page. **(5 points)**
- The logout functionality works correctly and redirects the user to "Login" page. **(5 points)**

60 points (Contacts page)

- "Contacts" page displays and has working navigation. **(10 points)**
- "Contacts" page displays the dropdown list and text box **(5 points)**
- "Contacts" page gets the list of contacts from the MySQL database. The contacts are dynamically added to the table created by the contacts page and displayed. **(20 points)**
- "Contacts" page displays the list of all contacts that matches the filtering criteria upon clicking the filter button. **(20 points)**

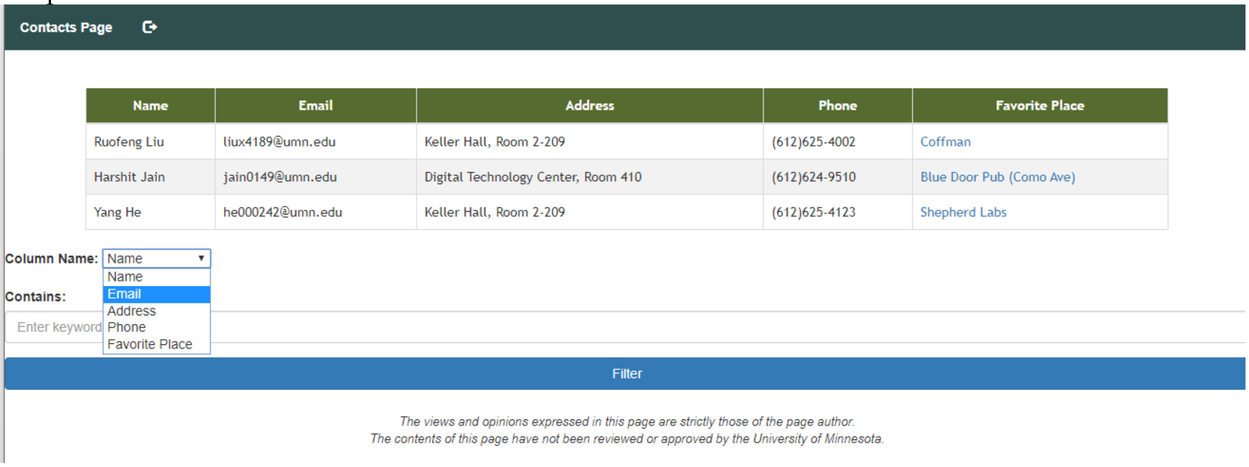
5. Assignment Guidelines and Penalties – we reserve the right to grade subjectively.

1. The illustrations in the lab assignments and demos, though not stated, are implied requirements - though the style can differ, it shouldn't cause your output to look worse than the example given. Therefore, though the CSS and page elements can vary but they should function the same as illustrated in the assignment write-up and demos, and by our judgment, they should not negatively impact the functionality described in the write-up and demos.

You should follow these guidelines while working on your assignments. Failure to do so may result in penalties assessed against your score that are not explicitly stated in the rubric.

6. Appendix

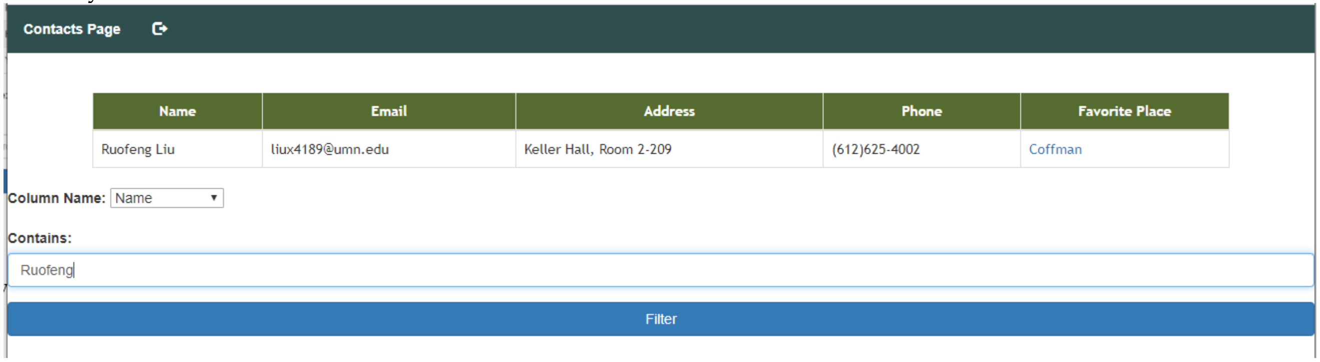
Dropdown list



Name	Email	Address	Phone	Favorite Place
Ruofeng Liu	liux4189@umn.edu	Keller Hall, Room 2-209	(612)625-4002	Coffman
Harshit Jain	jain0149@umn.edu	Digital Technology Center, Room 410	(612)624-9510	Blue Door Pub (Como Ave)
Yang He	he000242@umn.edu	Keller Hall, Room 2-209	(612)625-4123	Shepherd Labs

The views and opinions expressed in this page are strictly those of the page author.
The contents of this page have not been reviewed or approved by the University of Minnesota.

Filter by Contacts Name



Name	Email	Address	Phone	Favorite Place
Ruofeng Liu	liux4189@umn.edu	Keller Hall, Room 2-209	(612)625-4002	Coffman

Column Name: Name

Contains: Ruofeng