Csci 4131 Internet Programing
Spring 2018
Programming Assignment 8

Issued: Sunday 4/22/18

Submission Date/Time with Instant Bonus: Wed May 2nd at 11:55pm

Final Submission Date/Time *(No submissions accepted after this date, a grade of zero assigned)*: Friday May 4th at 11:55pm

(Note, no penalty and no bonus for submitting on Final Submission Date).

In this assignment, you are introduced to server-side programming with PHP. You will learn how to query the database, read and parse XML files with PHP, and use PhP Sessions. Your task is to dynamically create your favorites page using PHP. You will need following pages:

- *login.php*: This page is the PHP version of your "login.html" page from previous assignments. Your PhP script file should handle basic input validation and provide login functionality in this page. Once the user successfully logs in, the user should be redirected to the PhP script favplaces.php.
- *favplaces.php*: This page is the PHP version of your "favourites.html" page from previous assignments. You will basically display your favourite places you have created.
- *style.css*: Your CSS file (you can use the same file you used in your previous, non-PhP assignments).

1. Preparation:

The PhP script files for this assignment will reside on the server. Therefore, you need to setup your personal web folder .www folder on the CSE Lab 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:

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

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

illustrate how to create and set permissions on your .php files, and if you set it up, 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 above and the additional files provided to your home directory (or. www file, whose permissions should be set to at least 701 via the command: `chmod 701 .www`). Also, don't forget to change the permission of your `login.php` and `favplaces.php` files to 644 via the following commands:

```
chmod 644 login.php
chmod 644 favplaces.php
```

As specified in the CSELabs instructions, you can test your programs via the web browser by accessing the pages via the following urls;

http://www-users.cselabs.umn.edu/~YourX.500.id/login.php
http://www-users.cselabs.umn.edu/~YourX.500id/favplaces.php

- In the files for this assignment, you will find the skeleton file:
  - `dbconfig.xml` (`dbconfig.xml` contains the information regarding your database configuration)
- Download the dbconfig.xml and copy it to your .www directory (or the directory with your `login.php` and `favplaces.php` files).

- You need to edit the file `dbconfig.xml` and insert the database id and password we gave you for previous assignments in the appropriate places in the file `dbconfig.xml`.
- You should create a file ***database.php*** which reads and parses the file `dbconfig.xml` that you edited, and then assigns the values obtained from the xml file to the following variables
    - `$db_host,`
    - `$db_username,`
    - `$db_password,`
    - `$db_name, and`
    - `$db_port`

  Please note that the name of the variables should have the same name as the variables used in the PhP `mysqli` call provided below.

Database:

- The Database in this assignment contains two tables*: tbl_accounts and tbl_places* which are from previous assignments – you can use the information you have stored in them from previous assignments to login and query & construct your favourite places table.
- PhP files to re-create your tables and insert values have been provided, should you need them. Remember, you can use the information stored in the tables from your previous assignments to complete this assignment! Before you attempt to re-create and repopulate table, make sure any existing tables with the same name have been deleted from the database (DROP `tbl_accounts` and `tbl_places`).
- You can use following PHP code to initiate a database connection:

```
<?php
//…

include once 'database.php';

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

	// report error

} else {

	// setup your query
}
//…
?>
```

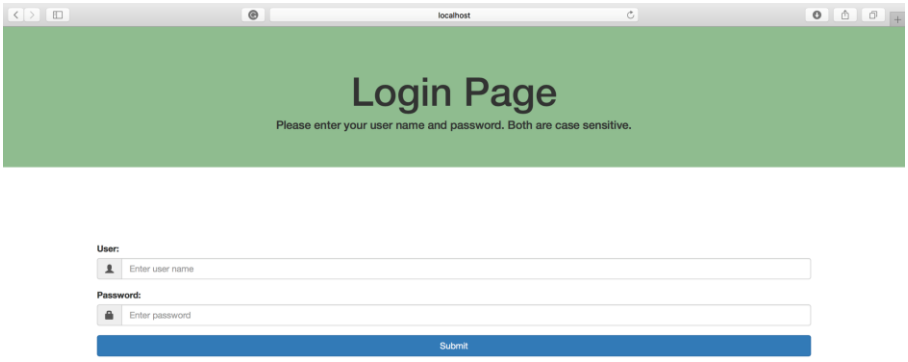
2. Functionality

Your PhP supported website will have at least 2 pages (you can add pages should your solution require it):

- **Login** page (`login.php`)
- **Favourite places** page (`favplaces.php`)

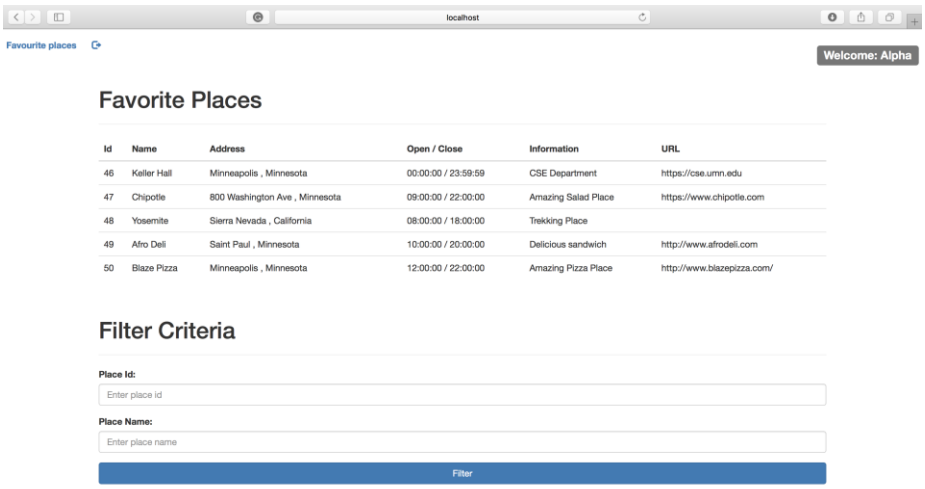
The `favplaces.php` script will create the page that displays the favourite places table and which accepts and issue queries on the information stored in the `tbl_places`. The Favorite places page will have a navigation bar with logout button and the logged in user display name.

1. Login Page (login.php)

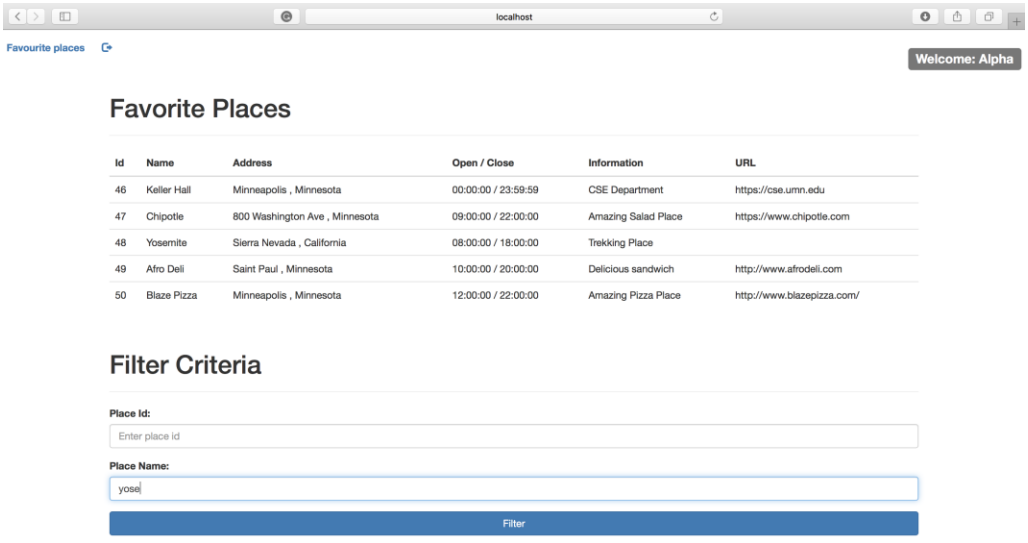


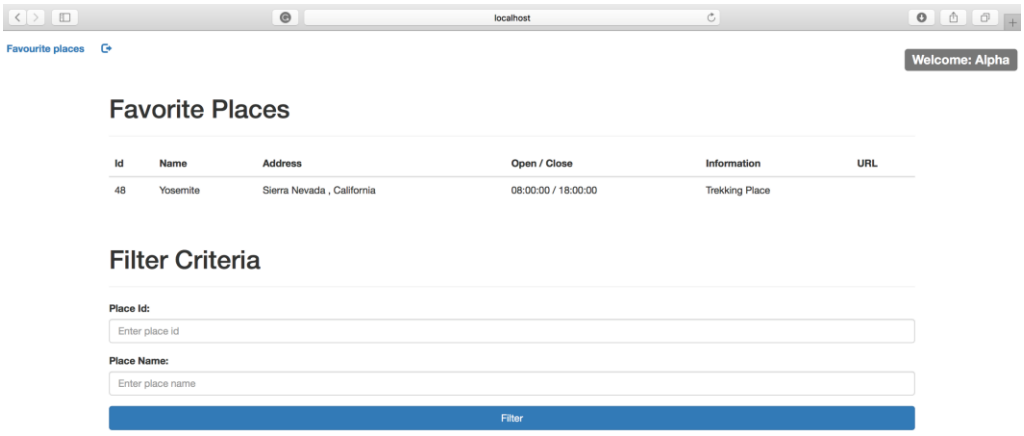
- This page should have a form with two fields: “User”, and “Password”
- Both these fields are mandatory.
- When the submit button is clicked, the values entered for “User” and “Password” should be sent to the server for validation before allowing further access to website.
- The server will validate the received values against the *acc_login*, and *acc_password* fields of *tbl_accounts*. Passwords should be stored in the database in a SHA1 hashed format in the database table *tbl_accounts*. The PhP login.php script on the server should convert the password string it is sent into a SHA1 hash format (using the sha1() function) 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, PhP script on the server should redirect the user back to the login.php:
- If the login is successful, the PhP script on the server should redirect to the *favplaces.php* pag

2. Favorites Page (favplaces.php)



- 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 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_places`.
- Filter (Query) Form:
    - This form has two fields: Place Id and Place Name.
    - Based on the values entered in the textboxes, the server should pull up the filtered content as follows:
        i. Upon entering a place Id, the server should display the place whose place Id matched the entered value.
        ii. Upon entering a text in the place name textbox, the server should display all the places which contains that text in their name.
        iii. If the user clicks the *filter* button without entering values in any of the textboxes, the server should pull up all the content from the table.





3. Logout

Upon clicking the logout button, the user session should be destroyed and user should be routed 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 HTML, CSS, JavaScript files in addition to any PhP files you created and utilize in your solution.

Create a README file inside the directory with your files for HW8. The README file should include: Your x500id, login(s) and password(s) in stored in the `table_accts` table.

Zip or tar the files required to run your assignment, along with the README file and store them in the file: **yourx500id_hw08. Submit the compressed file via the class Moodle site, using the HW 8 Submission Link.**

We will use the login(s) and password(s) to login to your website, so ensure that these values are correct and can be used to access your website. If we can't login to your website, you will be assigned a zero for the assignment.

4. Evaluation

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

15 points
- Submission instructions are met. **(5 points)**
- The `database.php` file should parse `dbconfig.xml` file to load the necessary variables that are used to establish connections to the MySQL database.**(10 points)**

35 points (Login page)
- The "Favourites" 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 incorrect log in credentials are entered on the Login page,, the user should be redirected to the same Login page. **(5 points)**
- After successful validation on server, the "Login" page redirects the user to "Favourite places" page. **(10 points)**
- The logout functionality works correctly and redirects the user to "Login" page. **(10 points)**

50 points (Favourites page)
- The "Favourite places" page diplays and has navigation. **(5 points)**
- The "Favourite places" page displays the name of the user that is logged in **(5 points)**
- The "Favourite places" page displays the list of favorite places from server. The favorite places are dynamically added to the table displayed by this page. **(20 points)**
- The Filter on place id works. **(5 points)**
- The Filter on place name works. **(5 points)**
- The "Favourite places" page gets the list of all favourite places from server when the user clicks the Filter button with no text input. **(5 points)**

BONUS – if your final submission of HW 8 occurs by 11:55pm on Wednesday, 5/2 you will receive a 10 point bonus for your assignment!!!