CO1707 Assignment 2 Support:

Log-in Page and Procedure Guidance:

This sheet has been prepared to support students in creating/ refining a log-in page as part of their assignment.

The required 'log-in' functionality can be achieved in three steps:

1. User accesses a web-form and enters their access credentials (username and password)
2. Web form 'posts' data to an external script, to check username exists and passwords match
3. User is then either passed to homepage with welcome message, or passed back to web-form.

We can achieve the above using four unique pages (three are new, one is modified from Assignment 1):

a) login.php: contains a web-form with an associated 'post' action
b) conn.php: external script for creating a database connection (used by any page)
c) myscript.php: external script for securing data and verifying credentials using SQL
d) index.php: modified version of our homepage, including a new 'welcome message'

Step 1: Creating the web form (easy):

We begin by creating a new file with .php extension. This includes simple web-form, and in this example is comprised exclusively of HTML. The page must make use of the following elements:

<form> *remember to include 'action' attribute to load required script, and 'method' attribute' to secure the data*

<input> *remember to include input fields for username and password, plus a 'submit' button to prompt 'action'.*

The minimum script should comprise of the following HTML objects:

```html
<!doctype html>
<html>
<body>
    <!-- Simple web form -->

    <form id="login" action="myscript.php" method="post">
    <p>Enter your username and password to proceed</p>

    <p><label>Username:</label>
    <input type="text" name="username" required></p>

    <p><label>Password:</label>
    <input type="text" name="password" required></p>
    <!-- replace 'text' above with 'password' to conceal password -->

    <p><input type="submit" name="Log Me In"></p>

    </form>
</body>
</html>
```

Guidance: We should refer to Week 12 materials for guidance on capturing and handling web-form data.

Step 2: Creating the connection (medium):

We begin by creating a new file wit a .php extension. This file will consist exclusively of PHP script. We should refer to Week 10 here for guidance on creating a database connection.

The php code is:

```
$connection = mysqli_connect("localhost","username","password","database");
```

Remember: change the 'username', 'password' and 'database' parameters above to match those of your database (refer to Vesta log-in email here for guidance).

The minimum script should comprise of the following php commands:

```
<?php
$connection = mysqli_connect("localhost","jbloggs1","abcdefg","jbloggs1");

// Check connection
if (mysqli_connect_errno())
{
    echo "Could not connect to database: " . mysqli_connect_error();

}//end if

return $connection;

?>
```

Guidance: We should refer to Weeks 11 here for guidance on creating a database connection.

Step 3: Creating the 'verification' php script (hard)

Our verification procedure should operate as follows:

1) Secure the data received in the 'post' (super global) variable using htmlspecialchars()
2) Load our connection variable (see above) and write an SQL statement to extract matching records
3) Combine our SQL with the connection variable (from previous step) and send to server (via mysqli)
4) Record the number of records returned (should be unique) and test this (conditional statement)
5) If unique record returned, process the SQL result as an associative array and query 'password' field.
6) Compare entered password with queried password. Test if they match (second conditional statement)
7) If the passwords match, create two session variables and redirect the user to the homepage (index.php)
8) If passwords do not match, redirect the user to the login page [consider passing an error message here].

Re. Step 6 above, the required session variables are:

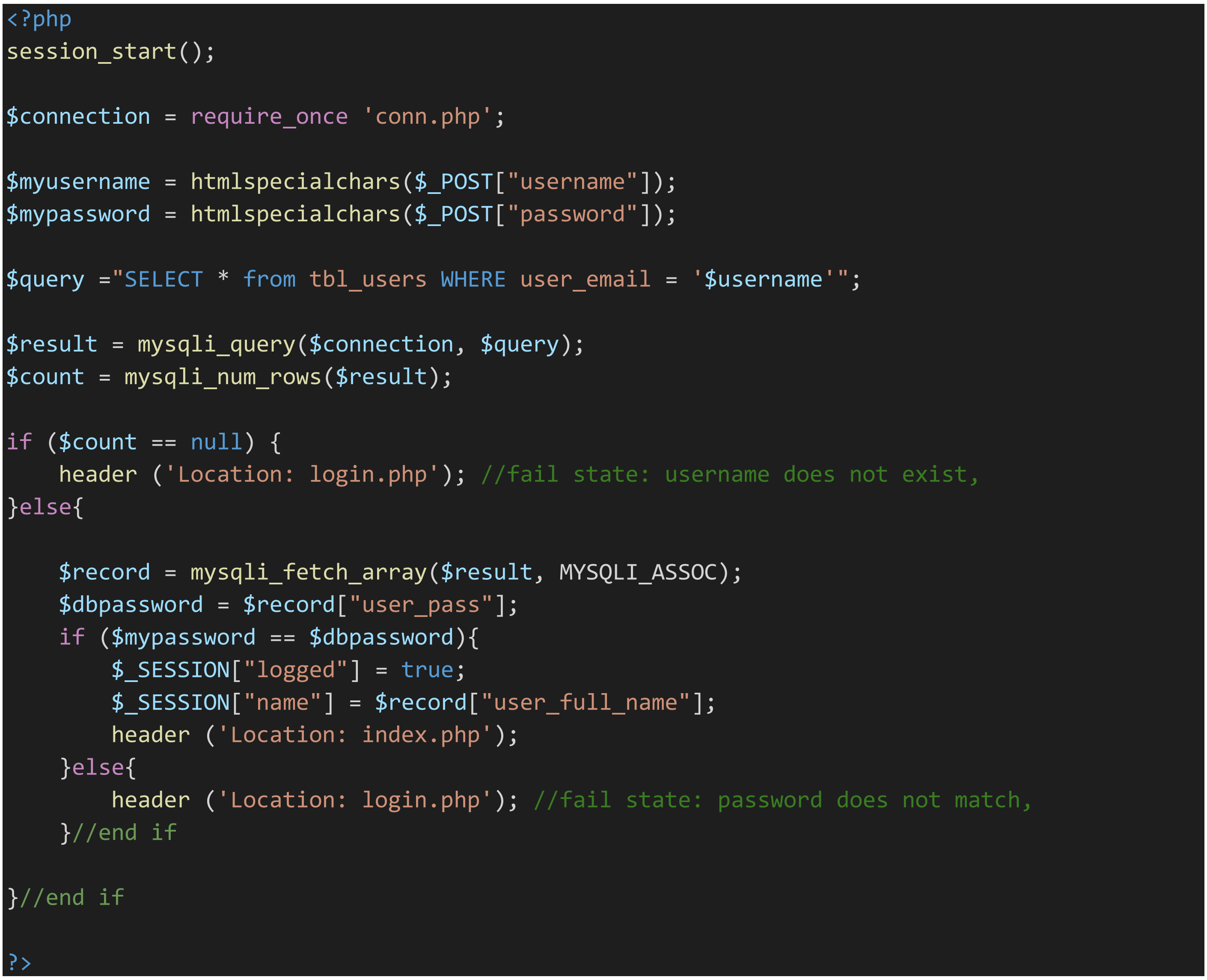
$_SESSION["logged"]

This session variable should be set to 'true' and will be used to verify whether a user is logged-in, across pages

$_SESSION["name"]

This session variable should be set to the value in the associative array matching the 'name' field (in database).

The minimum script should comprise of the following php commands:



```php
<?php
session_start();

$connection = require_once 'conn.php';

$myusername = htmlspecialchars($_POST["username"]);
$mypassword = htmlspecialchars($_POST["password"]);

$query ="SELECT * from tbl_users WHERE user_email = '$username'";

$result = mysqli_query($connection, $query);
$count = mysqli_num_rows($result);

if ($count == null) {
    header ('Location: login.php'); //fail state: username does not exist,
}else{

    $record = mysqli_fetch_array($result, MYSQLI_ASSOC);
    $dbpassword = $record["user_pass"];
    if ($mypassword == $dbpassword){
        $_SESSION["logged"] = true;
        $_SESSION["name"] = $record["user_full_name"];
        header ('Location: index.php');
    }else{
        header ('Location: login.php'); //fail state: password does not match,
    }//end if

}//end if

?>
```

Guidance: We should refer to Weeks 10 and 12 here for guidance on query a database and handling data.

Step 3: Modifying the index.php (easy)

Our index.html from Assignment 1 must be converted to .php to run on the server. We must then add 2 components to our page, using PHP (hence the .php extension):

a) include a ‘session_start()’ php command at the top of the page
b) include a php ‘echo’ immediately below our menu to print the session variable created earlier.

The minimum script should comprise of the following php commands:

```
<?php
<!—a) Open PHP session -->
session_start();

?>

<!doctype html>
<!-- Content continues here... -->

<body>
    <header>
    <!-- This section includes logo, menu, etc. from Assignment 1... -->
    </header>

    <main>
    <!-- b) This section NOW includes PHP, to echo the session variable -->

    <?php
    echo "Welcome back " . $_SESSION["name"];
    ?>

    <!-- Content continues here... -->
    </main>

</body>
</html>
```

We should refer to Week 13 materials here for guidance on PHP Session variables.

Finally: Some limitations of note, which require action on your part:

1. The ‘verification’ script assumes that passwords are stored as plain-text (i.e., ‘password101’). This is permissible to demonstrate ‘log-in’ functionality but does not satisfy security considerations. If using hashed passwords (recommended), the conditional statement (see: myscript.php, Step 6) should then be replaced with:

`if (password_verify($mypassword, $dbpassword)){ ...`

2. The web-form uses the ‘required’ attribute to check whether data has been entered. A more complex web-form (such as that required for ‘user registration’) should make use of Javascript to validate the form data, before passing data to any external script. Refer to the following here: https://www.w3schools.com/js/js_validation.asp

End.