Assignment IV – Web Application Programming with PHP/MySQL

CPS 499-02/592-02

Software/Language Based Security

Fall 2020

Dr. Phu Phung

Evan Krimpenfort

Task I: Mock-up login check with session in the index.php file

a. Set a "logged" value

```
17      if (isset($username) and isset($password))
18      {
19          //the case username and password is provided
20          if (/*TODO for TASK 3.b*/mockchecklogin($username,$password))
21          {
22              $_SESSION["logged"] = TRUE;
23              $_SESSION["username"] = $username;
24              $welcome = "Welcome "; //not previously logged-in
25          }
26          else
27          {
28              //failed
29              redirect_login('Invalid username/password');
30          }
31      }
```

Figure 1: Setting the Logged Value

When the user logs in, we want to be able to set a flag that states that a person with that session has some variable by the name of "logged" with the value of true. We can use this kind of variable to access certain things that a person with a different session of "logged" cannot access.

b. Check if the "logged" value isn't true

```
32      else
33      {
34          //no username/password is provided
35          //check if the session has NOT been logged in, redirect to the login page
36          if ($_SESSION["logged"]!=TRUE)
37          {
38              redirect_login('You have not logged in. Please login first!');
39          }
40      }
```

Figure 2: Checking if the Logged Value isn't true

Because we set that "logged" variable, we can now use it for page refreshes because when there is a refresh, the data in the text boxes goes away, but that variable doesn't. It doesn't go away because we tied that data to the session. If we already logged in and we decided to refresh the page, we check to see if that data is true so we can redirect that page now to the logged in page.

c. Testing

i. Get the Alert

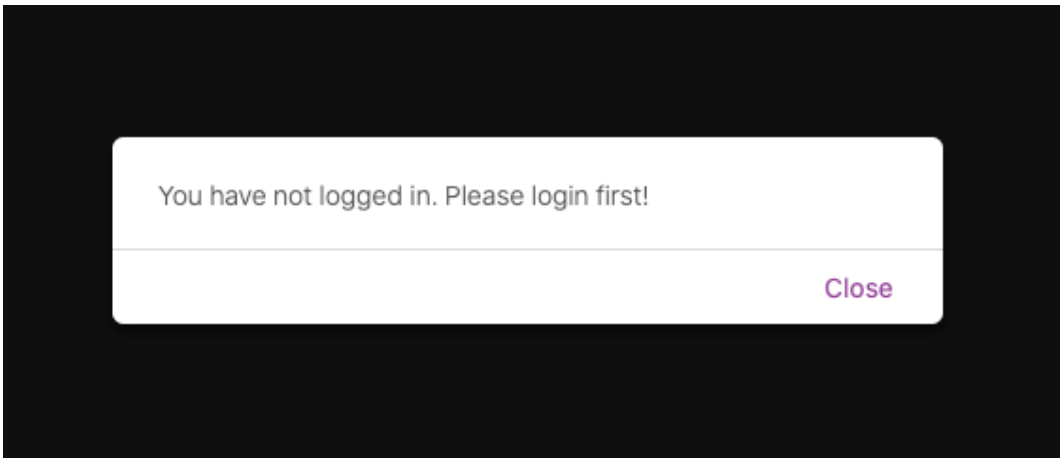

Figure 3: Alert from not logging in first. Logged = False

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Simple Web Application - Lab 4 - SS-LBS</title>
</head>
<body>
    <h1>SS-LBS - Lab 4</h1>
    <h2>Simple Web Application</h2>
    <h2>Simple index page by <font color="blue">Phu Phung</font>, customized by "Evan Krimpenfort"</h2>
DEBUG>Received: username="" and password=""<br>
<script>alert('You have not logged in. Please login first!');</script>

```

Figure 4: View-Source of the alert

Ii. Invalid Password

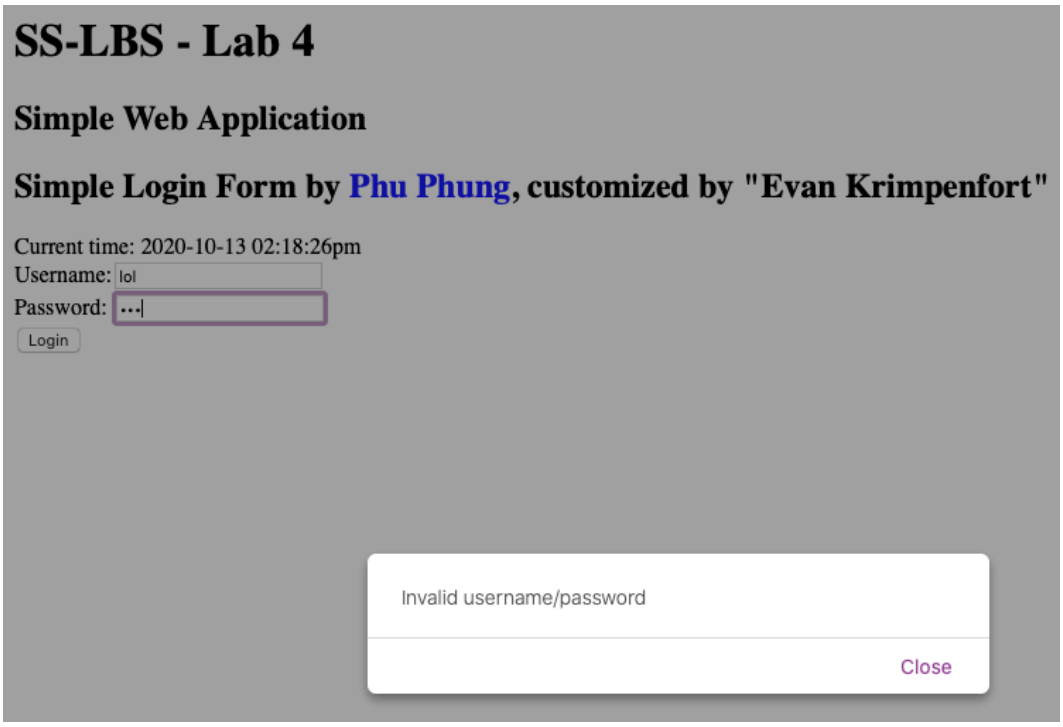

Figure 5: Invalid Password Alert. "Logged" still false

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Simple Web Application - Lab 4 - SS-LBS</title>
</head>
<body>
    <h1>SS-LBS - Lab 4</h1>
    <h2>Simple Web Application</h2>
    <h2>Simple index page by <font color="blue">Phu Phung</font>, customized by "Evan Krimpenfort"</h2>
DEBUG>Received: username="lol" and password="lol"<br>
<script>alert('Invalid username/password');</script>

```

Figure 6: View-Source of Invalid Password Alert

iii. Correct Password

Figure 7: Correct Password Index page. "Logged" now true

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Simple Web Application - Lab 4 - SS-LBS</title>
</head>
<body>
    <h1>SS-LBS - Lab 4</h1>
    <h2>Simple Web Application</h2>
    <h2>Simple index page by <font color="blue">Phu Phung</font>, customized by "Evan Krimpenfort"</h2>
DEBUG>Received: username="admin" and password="admin"<br>
Current time: 2020-10-13 02:32:57pm
<h2>Welcome <font color='blue'>admin</font>!</h2>
    <a href="logout.php">Logout</a>
</body>
</html>

```

Figure 8: View-Source of Correct Password index page

iv. Go back to the Login page

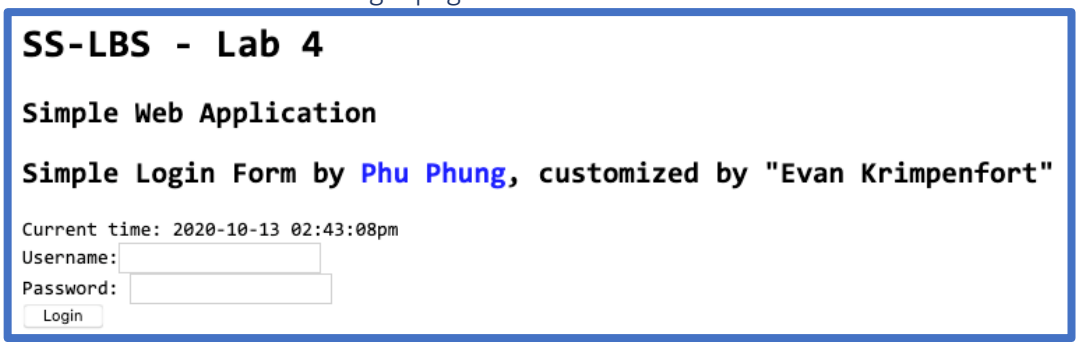

Figure 9: Back to the Login Page

This is because “logged” can’t be viewed from the login.php page pin figure 9. The “logged” value is only seen in the index.php page.

v. Close and try index.php again

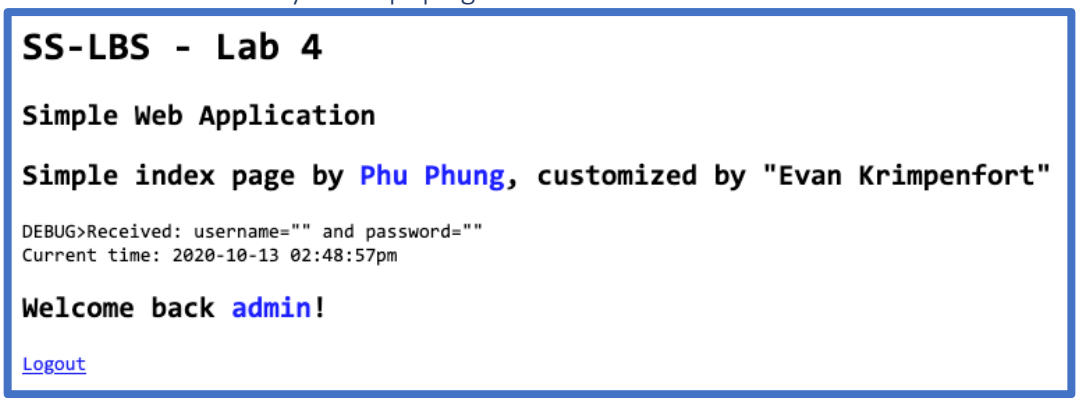

Figure 10: Back to the index page

After closing and opening my browser in figure 10, going to the index.php page allows me to log in right away because the session was never destroyed Thus, allowing that “logged” variable not to be turned false.

vi. Logout and try index.php

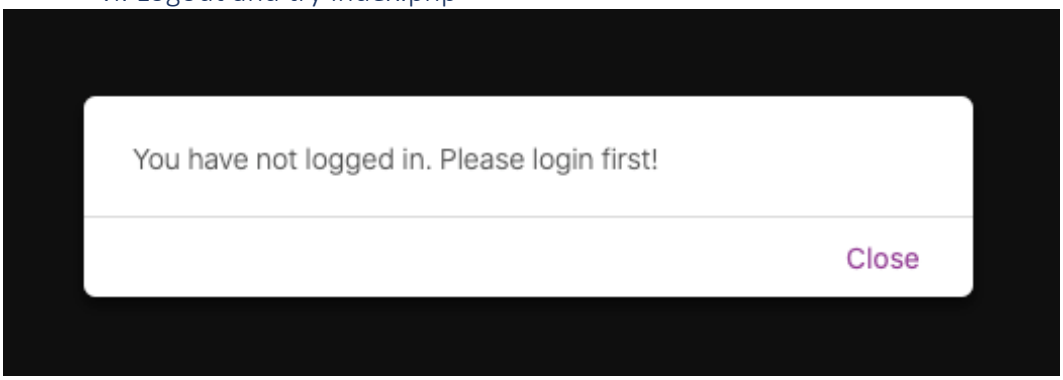

Figure 11: Logout and back to the index page

Because we logged out and the logout page destroys the session, when we go to the index page, it sees "logged" as false and asks the user to login in again.

Task II: Check if the session is logged in login.php page

a. Write code

```php
<?php
    /*TODO for TASK 2.a*/
    session_start();
    if (isset($_SESSION["logged"]) and $_SESSION["logged"] === TRUE)
    {
        echo "<script>alert('You have been logged in. Welcome Back!');</script>";
        header("Refresh:0 url=index.php");
        exit();
    }
    echo "Current time: " . date("Y-m-d h:i:sa") . "<br>\n";
?>
```

Figure 12: Adding "logged" arithmetic to login.php

The implementation I did in figure 12 was getting the "logged" value and checking to see if the value was true (meaning the user as already logged in and has never logged out). After that the code pastes an alert saying you've been logged in and It also refreshes the page so that it's now index.php.

b. Test code

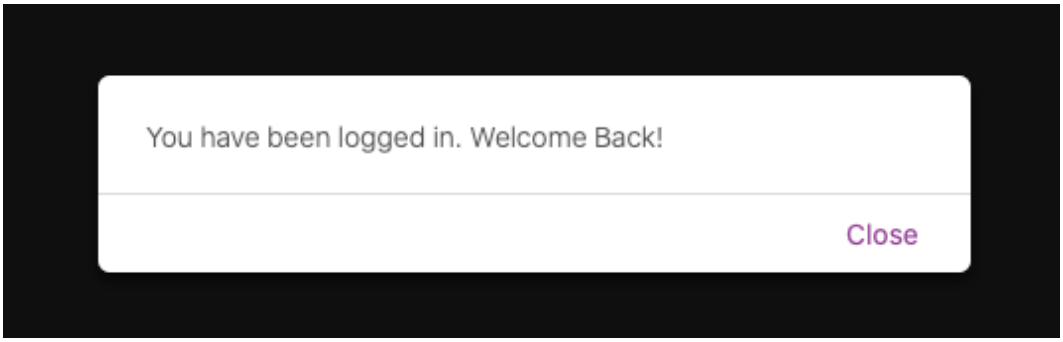

Figure 13: Login page fixed Alert

What happens now after you login in figure 13 and go back to the login page is that now login.php uses the “logged” variable to get the user back into the index page if they decide to close the page and get into it later.

Task III: Database interaction

a. Write code

```
/*TODO for TASK 3.a*/
$sql = "SELECT * FROM users where username='" . $username .
    "' AND password = password('" . $password . "')";
echo "DEBUG>sql=" . $sql . "\n<br>";
$result = mysqli_query($dbconnection,$sql);
if ($result)
{
    // check if matching
    $row = mysqli_fetch_assoc($result);
    if ($row['username'] === $username)
        return TRUE;
}
return FALSE;
```

Figure 14: Added code to checklogin(..)

So what’s going on here is that $sql is being written as mysql code to retrieve if the data exists. If it does, the code returns that the username exists and the checklogin function returns true. If the username does not exist, the checklogin function returns false.

b. Change the function call

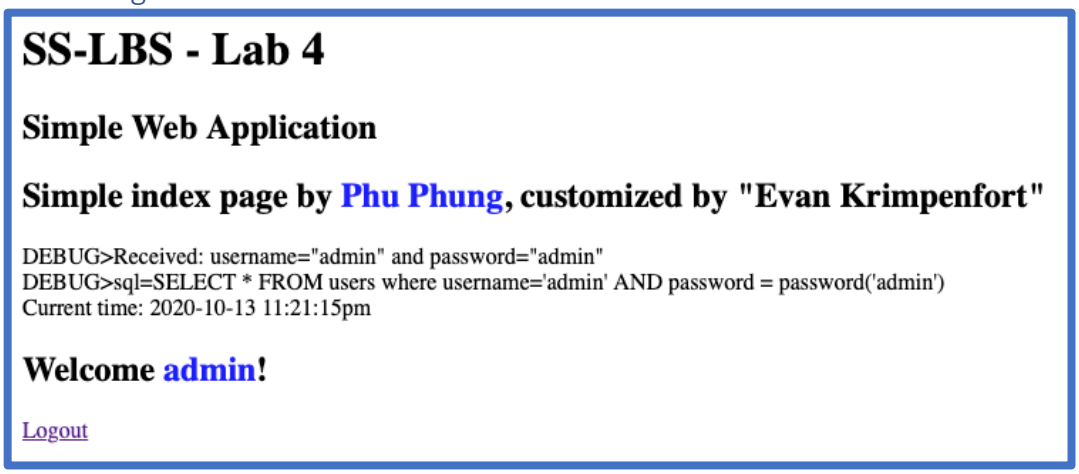

Figure 15: Test success with always true

c. Login to MySQL

```
mysql> insert into users VALUES
    -> ('krimpenforte1@udayton.edu',password('Aaaaaaa1'));
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> select * from users;
+---------------------------+-------------------------------------------+
| username                  | password                                  |
+---------------------------+-------------------------------------------+
| admin                     | *4ACFE3202A5FF5CF467898FC58AAB1D615029441 |
| krimpenforte1@udayton.edu | *E73C0CE2FF6B28F614E09892CE9C29ABC428E28C |
+---------------------------+-------------------------------------------+
2 rows in set (0.00 sec)
```

Figure 16: Email in the Database

The data in the users query has two rows. One row for the admin username and password and one row for my email username and password. This allows the checklogin function to go through the database shown in figure 16 and find the correct username and corresponding password.

d. Show new entry

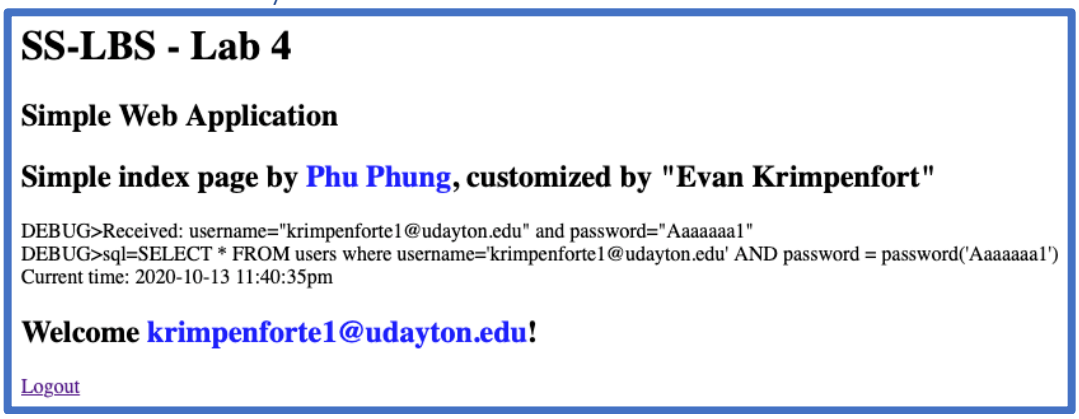

Figure 17: Email success in the index page