

PHP: Arrays, Functions and Form processing

Overview

This lab walks you through using PHP to create simple applications. PHP is popular for many Web applications, so becoming comfortable with the syntax of PHP will help you diagnose and identify potential security issues. It is not envisioned you will become an expert in PHP from this course, but you will be able to create simple Web applications; and, in the near future of this course, analyze that code for security issues.

Learning Outcomes:

At the completion of the lab you should be able to:

1. Create and test PHP scripts that include Arrays
2. Create and test PHP scripts that include Functions
3. Create, and test PHP scripts to process HTML Forms
4. Compare and contrast session creating mechanisms in PHP

Lab Submission Requirements:

After completing this lab, you will submit a word (or PDF) document that meets all of the requirements in the description at the end of this document. In addition, your Web Application files should be submitted. You can submit multiple files in a zip file.

Virtual Machine Account Information

Your Virtual Machine has been preconfigured with all of the software you will need for this class. The default username and password are:

Username : umucsdev

Password: umuc\$d8v

Part 1 – Create and test PHP scripts that include Arrays

This exercise will walk through creating a PHP script that creates, uses and manipulates arrays. We will use the gedit text editor to create the PHP file in the Virtual Machine.

1. After launching the gedit text editor, create a new document and type or copy and paste the PHP code shown below:

```
<!-- PHP and Arrays
Date: Jan 01, XXXX
Author: Dr. Robertson
Title: ArraysDemo.php
description: Demo how to use Arrays in PHP
-->
<!DOCTYPE html>
<html>
<head>
  <title>Arrays Demo </title>
```

```

</head>
<body>
<h1>PHP Arrays Demo </h1>
<?php
// Create a simple array of numbers
$numbers = array( 11,43,4,5,7,10);

echo "<h3> Current Numbers </h3>";
// Create a table and display the numbers

echo "<table border='1'>";
    foreach ( $numbers as $val ) {
        echo "<tr>";
        echo "<td>$val</td>";
        echo "</tr>";
    }
echo "</table>";

// Sort the array
sort($numbers);

echo "<h3> Sorted Numbers </h3>";
// Create a table and display the numbers

echo "<table border='1'>";
    foreach ( $numbers as $val ) {
        echo "<tr>";
        echo "<td>$val</td>";
        echo "</tr>";
    }
echo "</table>";

// Create an Associate array
// Multi-dimensional array
$gpa=array(
    array(
        "student"=>"Joe Smith",
        "grade" =>"A"
    ),
    array(
        "student"=>"Mary Jones",
        "grade" =>"A"
    ),
    array(
        "student"=>"John Perry",
        "grade" =>"C"
    ),
);

// Display the Student Data
echo "<h3> Student data </h3>";
echo "<table border='1'>";
echo "<tr>
    <td>Student Name </td>
    <td> Current Grade </td>
</tr>";

```

```

// Loop through each dimension of the array
foreach ( $gpa as $g ) {

    echo "<tr>";
    foreach ( $g as $value ) {
        echo "<td>$value</td>";
    }

    echo "</tr>";
}
echo "</tr>";
echo "</table>";

// Sort the Associative Array
sort($gpa);

// Display the Sorted Student Data
echo "<h3> Sorted Student data </h3>";
echo "<table border='1'>";
echo "<tr>
    <td>Student Name </td>
    <td> Current Grade </td>
</tr>";
// Loop through each dimension of the array
foreach ( $gpa as $g ) {

    echo "<tr>";
    foreach ( $g as $value ) {
        echo "<td>$value</td>";
    }

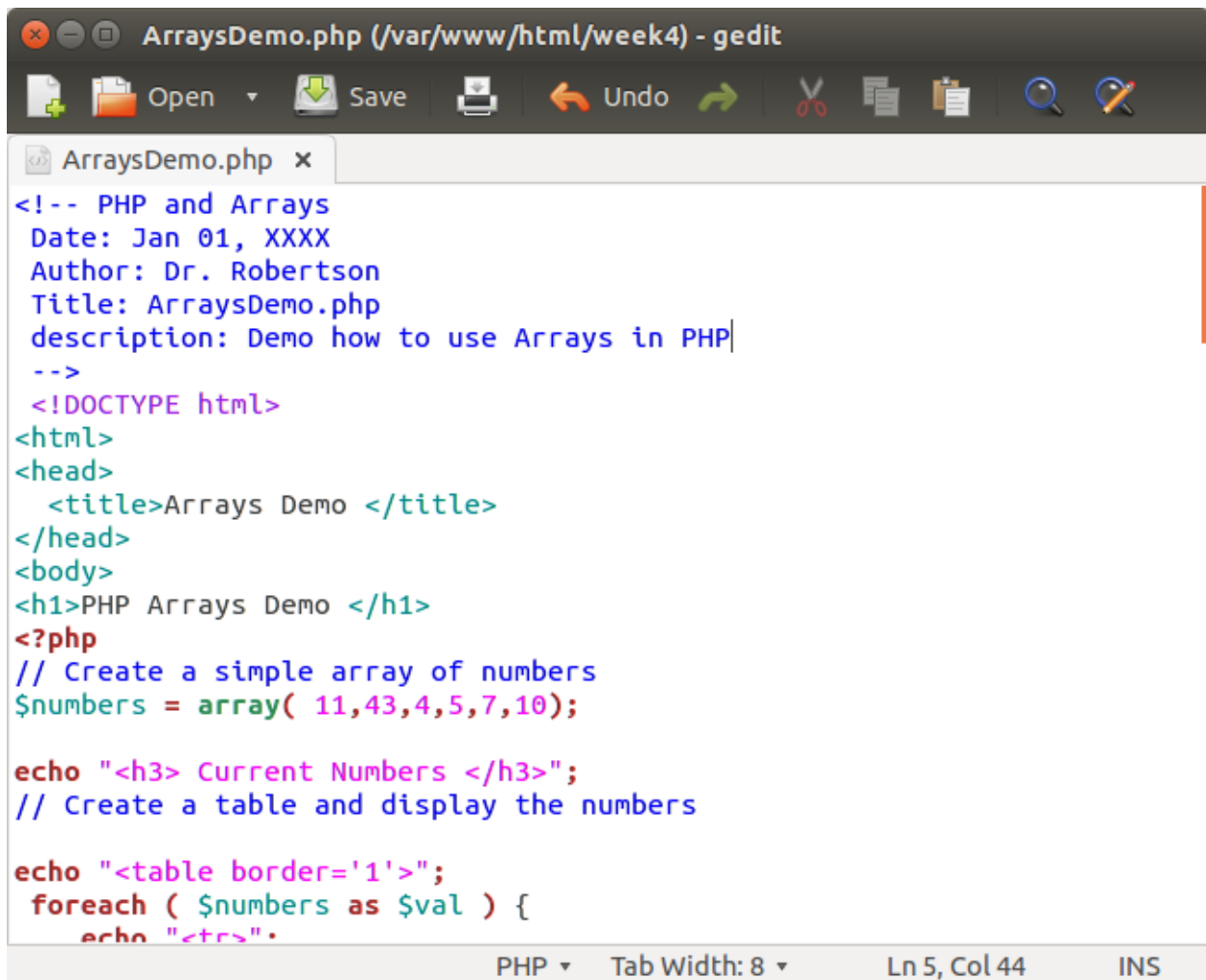
    echo "</tr>";
}
echo "</tr>";
echo "</table>";

?>
</body>

</html>

```

Save the file in the `/var/www/html/week4` folder in a file named `ArraysDemo.php`. Note, you may need to create a folder named `week4`. Recall the `/var/www/html` is the location of the Apache2 web server html files. Creating separate folders for each week or application will help organize the server.



```
ArraysDemo.php (/var/www/html/week4) - gedit
Open Save Undo
ArraysDemo.php x
<!-- PHP and Arrays
Date: Jan 01, XXXX
Author: Dr. Robertson
Title: ArraysDemo.php
description: Demo how to use Arrays in PHP
-->
<!DOCTYPE html>
<html>
<head>
  <title>Arrays Demo </title>
</head>
<body>
<h1>PHP Arrays Demo </h1>
<?php
// Create a simple array of numbers
$numbers = array( 11,43,4,5,7,10);

echo "<h3> Current Numbers </h3>";
// Create a table and display the numbers

echo "<table border='1'>";
foreach ( $numbers as $val ) {
  echo "<tr>";
```

PHP Tab Width: 8 Ln 5, Col 44 INS

Launch the Firefox browser and run your home page by entering the following URL:
localhost/week4/ArraysDemo.php

4
5
7
10

Sorted Numbers

4
5
7
10
11
43

Student data

Student Name	Current Grade
Joe Smith	A
Mary Jones	A
John Perry	C

Sorted Student data

Student Name	Current Grade
Joe Smith	A
John Perry	C
Mary Jones	A

As you analyze and experiment with the code, you should note the integration of the Arrays into the html displays. Also, note how the arrays are declared and initialized with data:

For a single dimensional array the declaration and initialization is fairly straight forward:

```
$numbers = array( 11,43,4,5,7,10);
```

For a multi-dimensional associative array the syntax is trickier:

```
$gpa=array(
    array(
        "student"=>"Joe Smith",
        "grade" =>"A"
    ),
    array(
        "student"=>"Mary Jones",
        "grade" =>"A"
    ),
    array(
        "student"=>"John Perry",
        "grade" =>"C"
    ),
);
```

Notice the use of a nested array statements and use of => to associated a value for array element.

2. As before, you can also run the PHP code directly from the shell prompt. To run from the shell prompt, open a shell prompt, change to the location of the ArraysDemo.php file and type:

```
php ArraysDemo.php
```

Running from the shell may provide some insight when you php Errors that prevent the cause the script to stop running prior to producing the HTML output.

Part 2 Create and test PHP scripts that include Functions

In this exercise we will create a PHP web page that uses both existing and user-defined PHP functions. Functions are used to help organize code into sub-units to allow for code reuse and reproducible results.

1. Copy and paste the following code into a file named FunctionsDemo.php in the /var/www/html/week4 folder on your Virtual Machine.

```
<!-- PHP and Functions
Date: Jan 01, XXXX
Author: Dr. Robertson
Title: FunctionsDemo.php
description: Demo how to use Functions in PHP
-->
<!DOCTYPE html>
<html>
<head>
  <title>Functions Demo </title>
</head>
<body>
<h1>PHP Functions Demo </h1>
<?php
// Create a simple array of Degrees
$numbers = array( 15,30,45,75,90);

echo "<h3> Example PHP Functions </h3>";
// Create a table and display the numbers

echo "<table border='1'>";
echo "<tr>
  <th>Degree </th>
  <th> Sqrt(Degree) </th>
  <th> sin(Degree) </th>
  <th> cos(Degree) </th>
  <th> tan(Degree) </th>
  <th> cubeIt(Degree) </th>
</tr>";
foreach ( $numbers as $val ) {
  echo "<tr>";
  echo "<td>" . $val . "</td>";
  echo "<td>" . sqrt($val). "</td>";
  echo "<td>" . sin(deg2rad($val)). "</td>";
  echo "<td>" . cos(deg2rad($val)). "</td>";
```

```

        echo "<td>" . tan(deg2rad($val)). "</td>";
        echo "<td>" . cubeIt($val). "</td>";
        echo "</tr>";
    }
    echo "</table>";

// Simple Cube function
// Return the cube of the input value
function cubeIt($val) {
    return $val*$val*$val;
}

?>
</body>
</html>

```

2. Launch your Firefox browser and run the Web application. Assuming you placed the file in the /var/www/html/week4 folder you can run this by typing the following URL on your Virtual machine: localhost/week4/FunctionsDemo.php.

If successful, the resulting output will look similar to this:

The screenshot shows a web browser window with the title "Functions Demo". The address bar displays "localhost/week4/FunctionsDemo.php". The page content features a main heading "PHP Functions Demo" and a sub-heading "Example PHP Functions". Below this is a table with the following data:

Degree	Sqrt(Degree)	sin(Degree)	cos(Degree)	tan(Degree)	cubeIt(Degree)
15	3.8729833462074	0.25881904510252	0.96592582628907	0.26794919243112	3375
30	5.4772255750517	0.5	0.86602540378444	0.57735026918963	27000
45	6.7082039324994	0.70710678118655	0.70710678118655	1	91125
75	8.6602540378444	0.96592582628907	0.25881904510252	3.7320508075689	421875
90	9.4868329805051	1	6.1232339957368E-17	1.6331239353195E+16	729000

3. Reviewing the code you should note the following:

- a. Existing PHP functions can be used easily by calling the function name and any required parameters. For this example, `sqrt()`, `deg2rad()`, `sin()`, `cos()` and `tan()` existing functions were called.

- b. PHP functions you create should be of the format:

```
function functionName($parameter1, $parameter2 ...) {  
    // Code here  
    return $returnvalue;  
}
```

- c. You can create functions with any level of rigor and complexity as needed to solve the computing problem at hand. The simple PHP function provided for this example calculates the cube of the input parameter:

```
function cubeIt($val) {  
    return $val*$val*$val;  
}
```

Part 3 Create and test PHP scripts to process HTML Forms

In this exercise we will create a PHP web pages that include simple forms that use get and post methods for submission of data. The first set of code below is the HTML file providing 3 textfields and a PHP get method.

1. Copy and paste the following code into a file named `DemoGetForm.html` in the `/var/www/html/week4` folder on your Virtual Machine.

```
<html>  
<head><title>Simple Form with Get Method </title>  
</head>  
<body>  
<h1> Please complete the Form </h1>  
<form action="get_Submit.php" method="get">  
    First name: <input type="text" name="fname"><br>  
    Last name: <input type="text" name="lname"><br>  
    Password: <input type="password" name="mypass"><br>  
    <input type="submit" value="Submit">  
</form>  
</body>  
</html>
```

2. Create an additional file that will be used to process the HTML form that is submitted. The file should be named `get_Submit.php` and be placed in same location as the HTML file. The file should contain these contents:

```
<!-- HTML Forms with Get Submit  
Date: Jan 01, XXXX  
Author: Dr. Robertson
```

```

Title: get_Submit.php
description: Demo how to retrieve Form data
-->
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Get Form Echo</title>
</head>
<body>

<?php
  // Retrieve Data using GET method

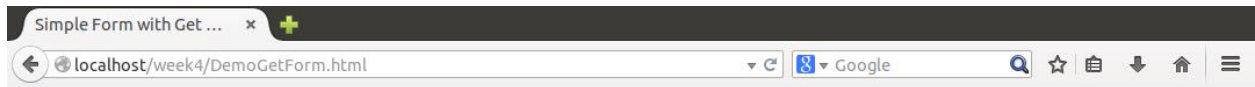
  $fname = $_GET["fname"];
  $lname = $_GET["lname"];
  $mypassword = $_GET["mypass"];

  // Display in a table
  echo "<h3> Form Data </h3>";
  echo "<table border='1'>";
  echo "<tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Password</th>
  </tr>";
  echo "<tr>
    <td>$fname</td>
    <td>$lname</td>
    <td>$mypassword</td>
  </tr>";
  echo "</table>";

?>
</body>
</html>

```

3. Launch the HTML file from your localhost/week4/DemoGetForm.html folder.



Please complete the Form

First name:

Last name:

Password:

4. Fill out the form and press submit



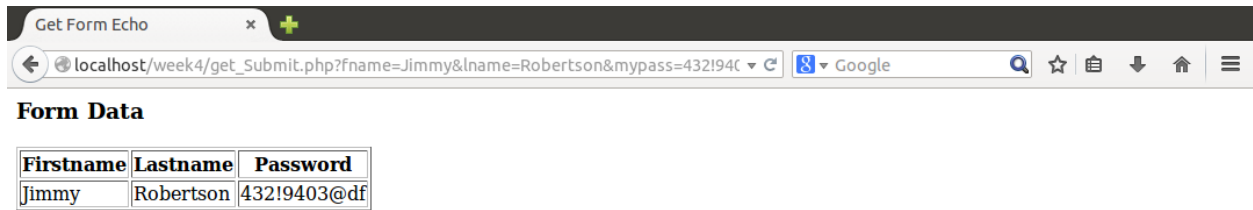
Please complete the Form

First name:

Last name:

Password:

5. The form data will be submitted the data will be echoed to display.



As you review the code and results, notice the query string sent at the URL provides all of the field parameters and their values. Also, notice this is in clear text and very insecure.

http://localhost/week4/get_Submit.php?fname=Jimmy&lname=Robertson&mypass=432!9403%40df

6. To compare with the Post method create two additional files using the following code:

HTML file: DemoPostForm.html

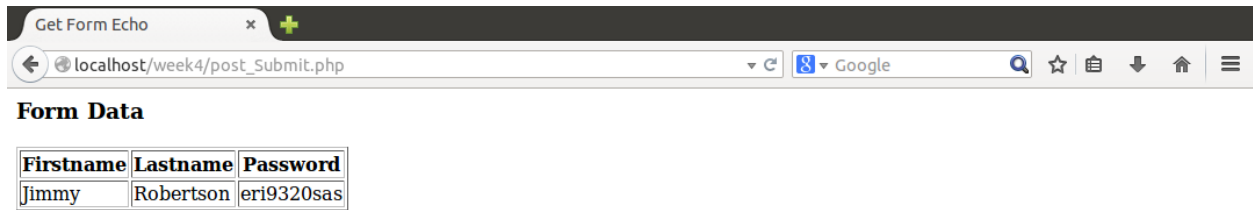
```
<html>
<head><title>Simple Form with Post Method </title>
</head>
<body>
<h1> Please complete the Form </h1>
<form action="post_Submit.php" method="post">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  Password: <input type="password" name="mypass"><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

PHP file: post_Submit.php

```
<!-- HTML Forms with Post Submit
Date: Jan 01, XXXX
Author: Dr. Robertson
Title: post_Submit.php
description: Demo how to retrieve Form data
-->
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Get Form Echo</title>
</head>
<body>
<?php
  // Retrieve Data using Post method
  $fname = $_POST["fname"];
  $lname = $_POST["lname"];
  $mypassword = $_POST["mypass"];
  // Display in a table
  echo "<h3> Form Data </h3>";
  echo "<table border='1'>";
  echo "<tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Password</th>
  </tr>";
  echo "<tr>
    <td>$fname</td>
    <td>$lname</td>
    <td>$mypassword</td>
  </tr>";
  echo "</table>";

?>
</body>
</html>
```

7. Launch and run the application and note the output display.



When comparing the get and post methods, note that the post doesn't display the fields in clear text for the URL:

http://localhost/week4/post_Submit.php

The post method for password transfer isn't necessarily more secure as there are issues with post submissions also, but at least the URL doesn't display the sensitive data in a query string.

Part 4 Compare and contrast session creating mechanisms in PHP

In this exercise we will create a simple PHP page that creates and reads available cookies as well as PHP session variables. For the first session example, we will create a cookie in PHP and use a form based submission to expire the cookie.

1. Copy and paste the following code into a file named DemoCookies.php in the /var/www/html/week4 folder on your Virtual Machine.

```
<!-- PHP and Cookies
Date: Jan 01, XXXX
Author: Dr. Robertson
```

```

Title: Demo_Cookies.php
description: Demo how to use Cookies with PHP
-->
<!DOCTYPE html>
<html>
<head>
  <title>Cookies Demo </title>
</head>
<body>
<h1>PHP Cookies Demo </h1>
<?php
$cookie_name = "UMUCGamer";
$cookie_value = "CMSC325";
// Set cookie for 7 days
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
// Check for cookie
if(!isset($_COOKIE[$cookie_name])) {
  echo $cookie_name . ", " . $cookie_value . "' is not set!";
} else {
  echo "Welcome back" . $cookie_name . "-" . $_COOKIE[$cookie_name];
}
?>
</br>
<!-- Form to expire cookie -->
<form action="expireCookie.php" method="post">
  <input type="submit" value="Expire Cookie">
</form>

</body>
</html>

```

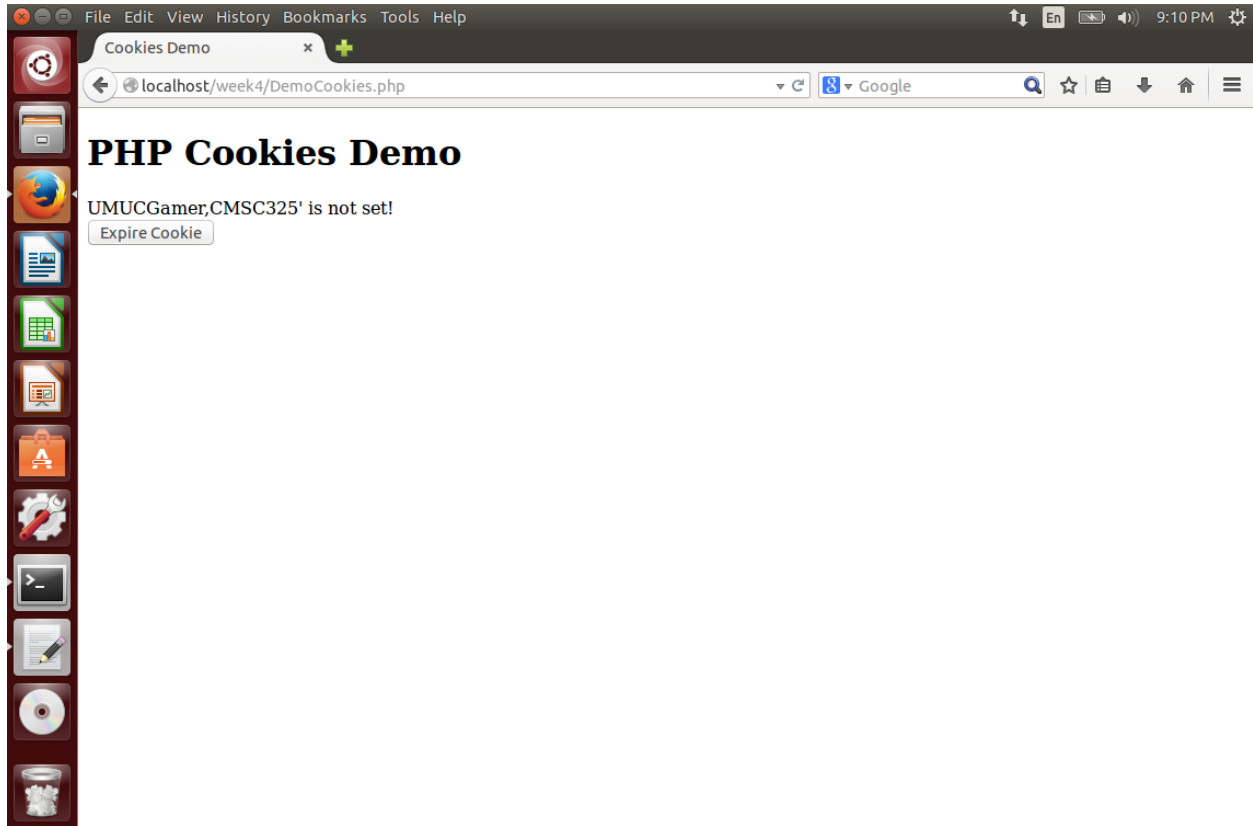
2. Create an additional file that will be respond when the Expire Cookie button is selected. The file should be named expireCookie.php and be placed in same location as the HTML file. The file should contain these contents:

```

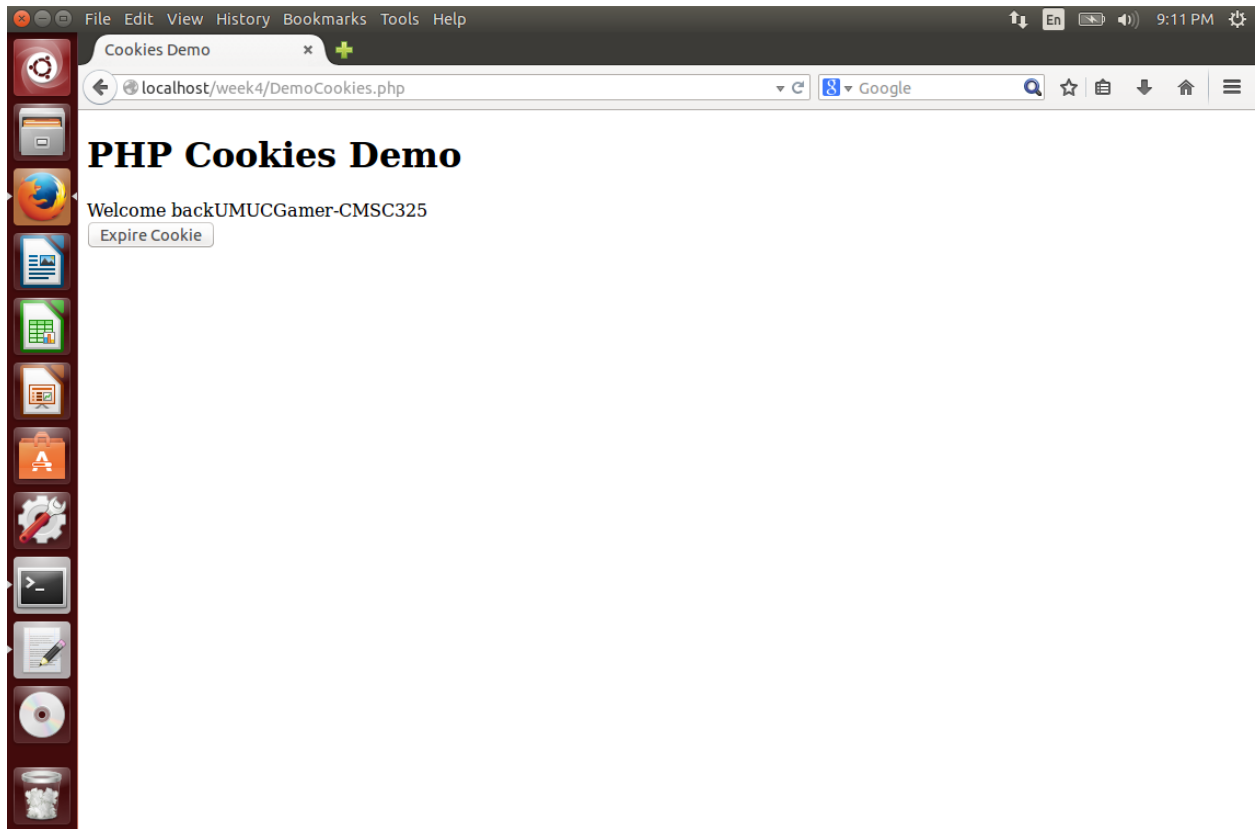
<html>
<head><title>Expire the cookies </title>
</head>
<body>
<?php
// Expire the cookie
$cookie_name = "UMUCGamer";
if(isset($_COOKIE[$cookie_name])) {
  setcookie( $cookie_name, "", time() - 3600, "/" );
  echo "Expiring the cookie: " . $cookie_name;
}
else {
  echo "Cookie not found to expire ";
}
?>
<h2> Thanks for playing with PHP cookies </h2>
</body>
</html>

```

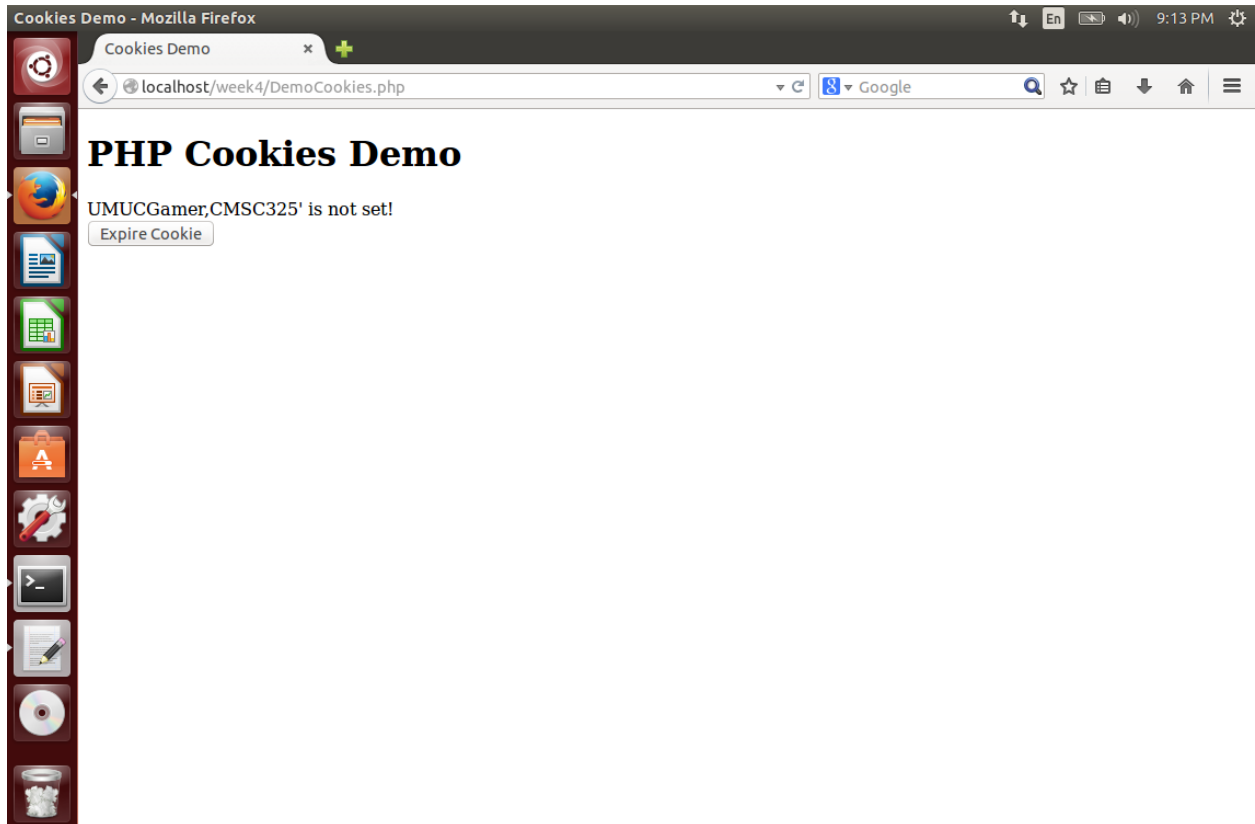
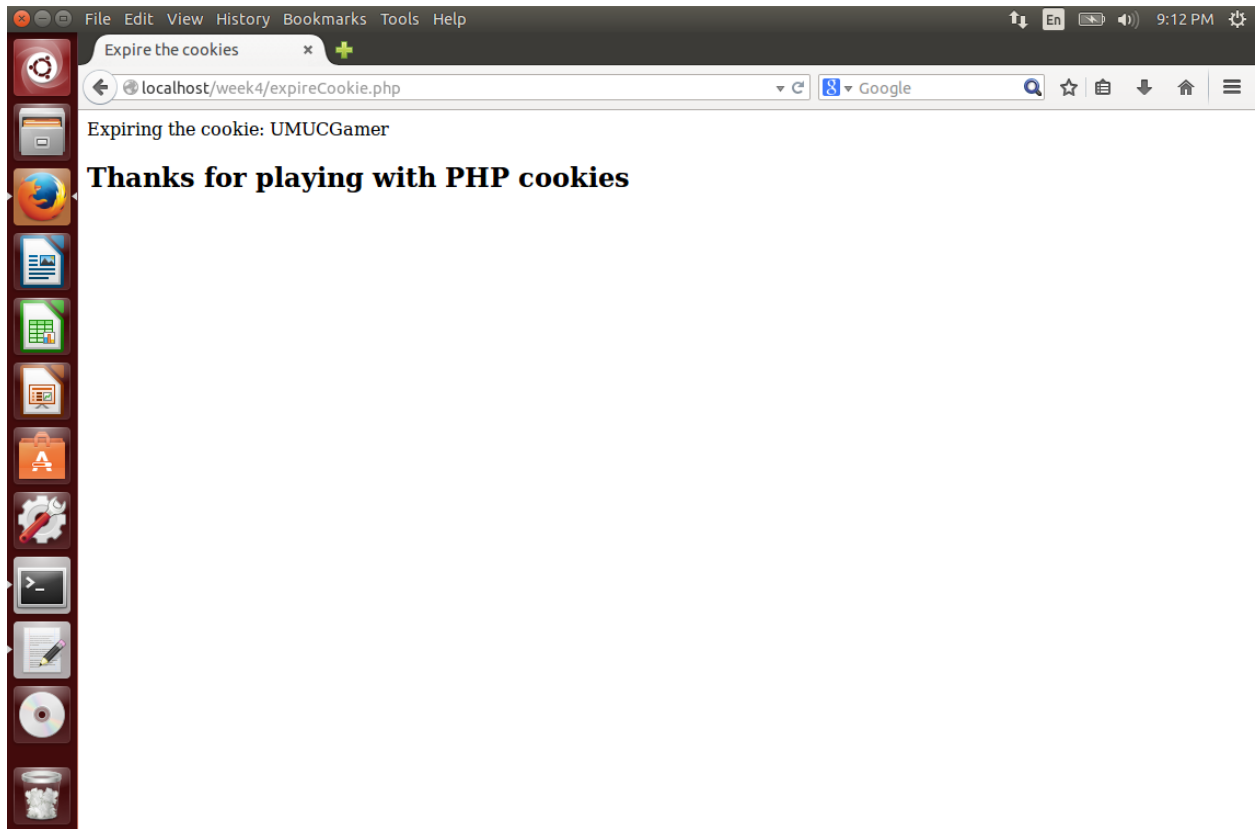
3. Launch the HTML file from your localhost/week4/DemoCookies.html folder. Cookies can be challenging to work with, because they often linger beyond their actual expiration date. This does have security implications which we will touch on through this program. When launching the application for the first time, the cookie has not been set.



When you refresh the browser and open it again, the cookie will be present as indicated by the welcome back message.



If you select the Expire Cookie button and refresh the browser, the cookie will no longer be present.



4. Next, we will look at the Sessions options in PHP. In this example, we will create 3 files. An html file will be used to input a username and email address using an html form. Upon submitting the form, a php file will start a session and store the username and email address in session variables. Finally, an option to logout and unset the stored session variables will be provided in a logout.php file.
5. Using gedit, create 3 files in the /var/www/html/week4 directory. The files should be named loginAuth.html, authcheck.php and logout.php; respectively.

File: loginAuth.html

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Form Login</title>
</head>

<body OnLoad="document.main.username.focus ();">

<table >
    <tr>
        <td colspan="2">
<h4>Enter your Username and Email Address to continue</h4>
</td>
</tr>
<!-- create the main form with an input text box named uid and a password
text box named mypassword -->
<form name="main" method="post" action="authcheck.php">
<tr>
<td>username:</td>
<td><input name="username" type="text" size="50"></td>
</tr>
<tr>
<td>Email Address:</td>
<td><input name="emailadd" type="text" size="50"></td>
</tr>
<tr>
<td colspan="2" align="center"><input name="btnsubmit" type="submit"
value="Submit"></td>
</tr>
</table>
</form>

</body>
</html>
```

File: authcheck.php

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>User Authenticate </title>
</head>
<body>
```

```

<?php
    // Retrieve Post Data
    $username = $_POST["username"];
    $email = $_POST["emailadd"];

    // Set the session information
    session_start();
    $_SESSION['appusername'] = $username;
    $_SESSION['appemail'] = $email;

    // Display the Session information
    echo "<h3> Session Data </h3>";
    echo "<table border='1'>";
    echo "<tr>
        <td>Username </td>
        <td> Email </td>
    </tr>";
    echo "<tr>
        <td>" . $_SESSION['appusername'] . "</td>";
    echo "<td>" . $_SESSION['appemail'] . "</td>";
    echo "</tr>";
    echo "</table>";

    // Provide a button to logout

    echo "<form name='logout' method='post' action='logout.php'>
    <input name='btnsubmit' type='submit' value='Logout'>
    </form>";
    ?>
</body>
</html>

```

File: logout.php

```

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Form Login</title>
</head>

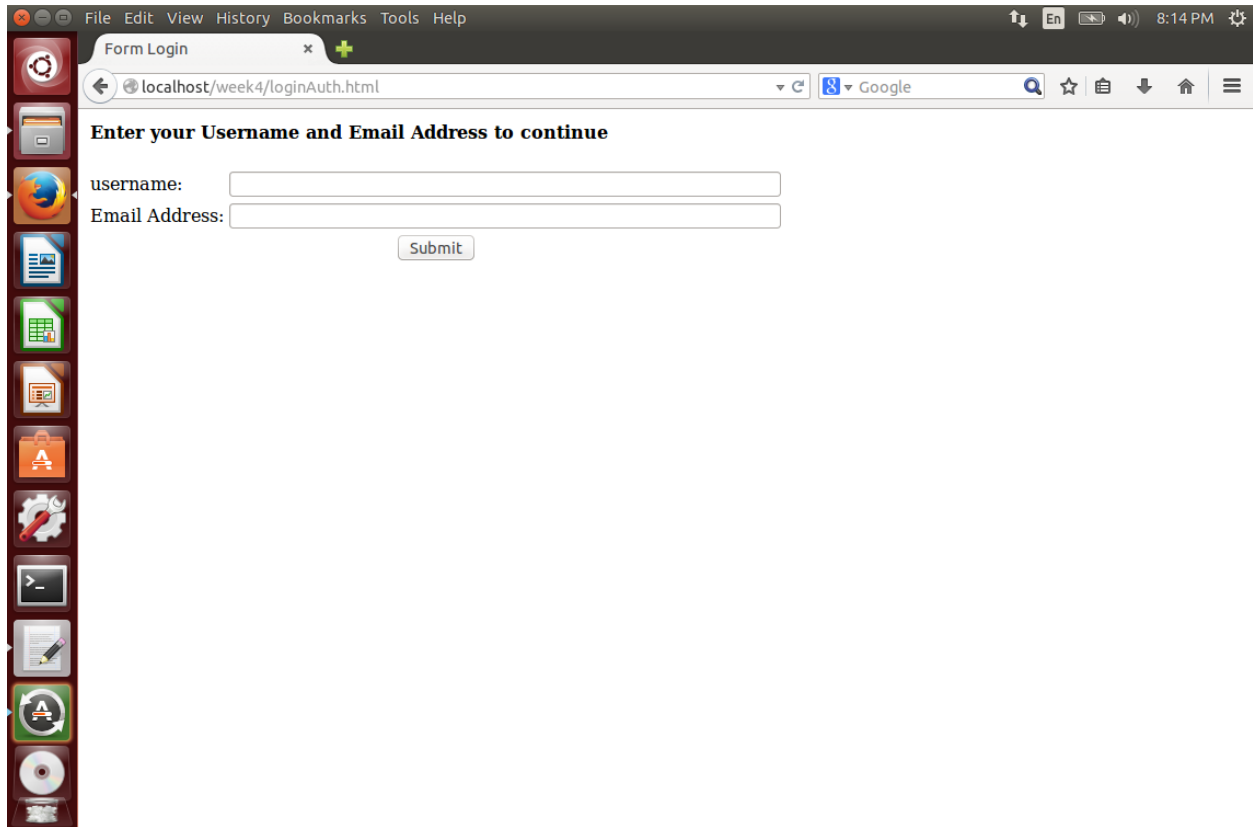
<?php
session_start();
unset($_SESSION['appusername']);
unset($_SESSION['appemail']);

// Display the Session information
echo "<h3> Session Data after Logout </h3>";
<table border='1'>
<tr>
    <td>Username </td>
    <td> Email </td>
</tr>
<tr>
    <td>" . $_SESSION['appusername'] . "</td>" .

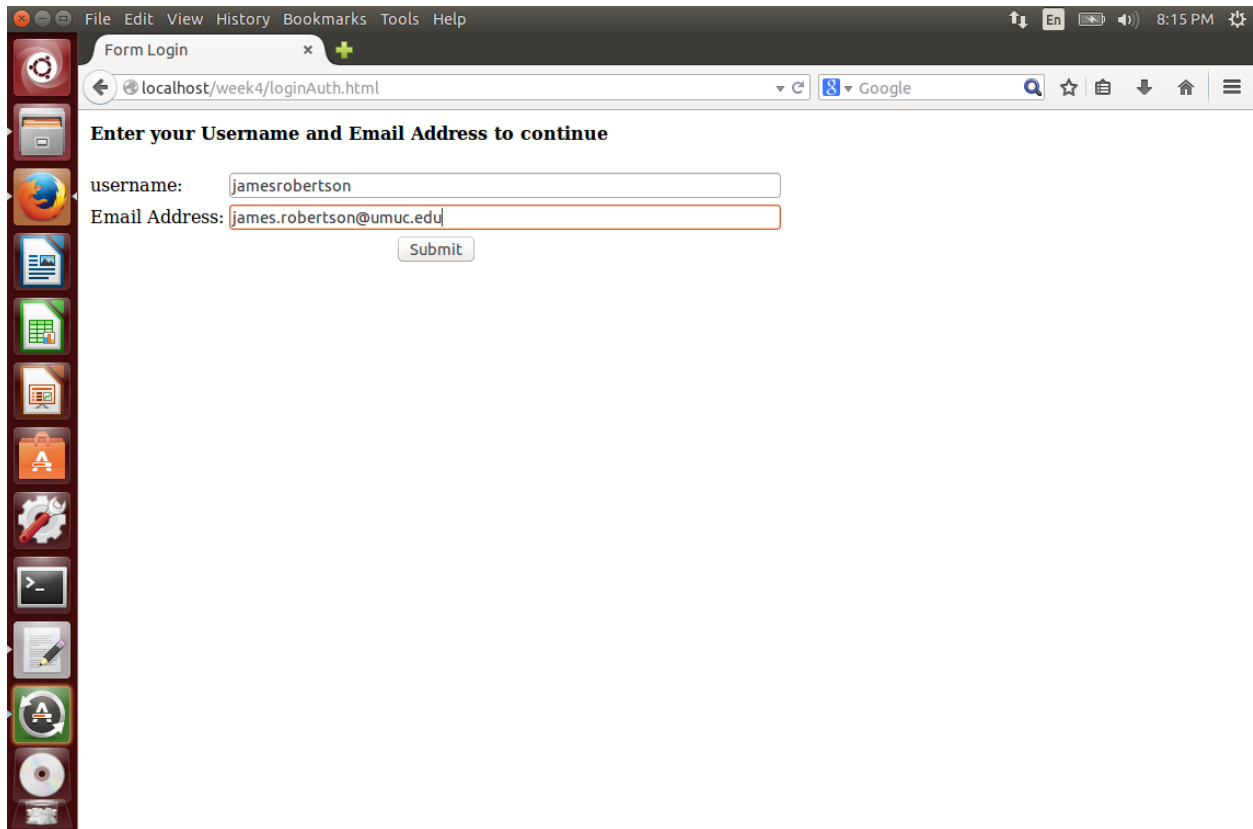
```

```
        "<td>" . $_SESSION['appemail'] . "</td>"  
    </tr>  
</table>";  
?>  
  
</body>  
</html>
```

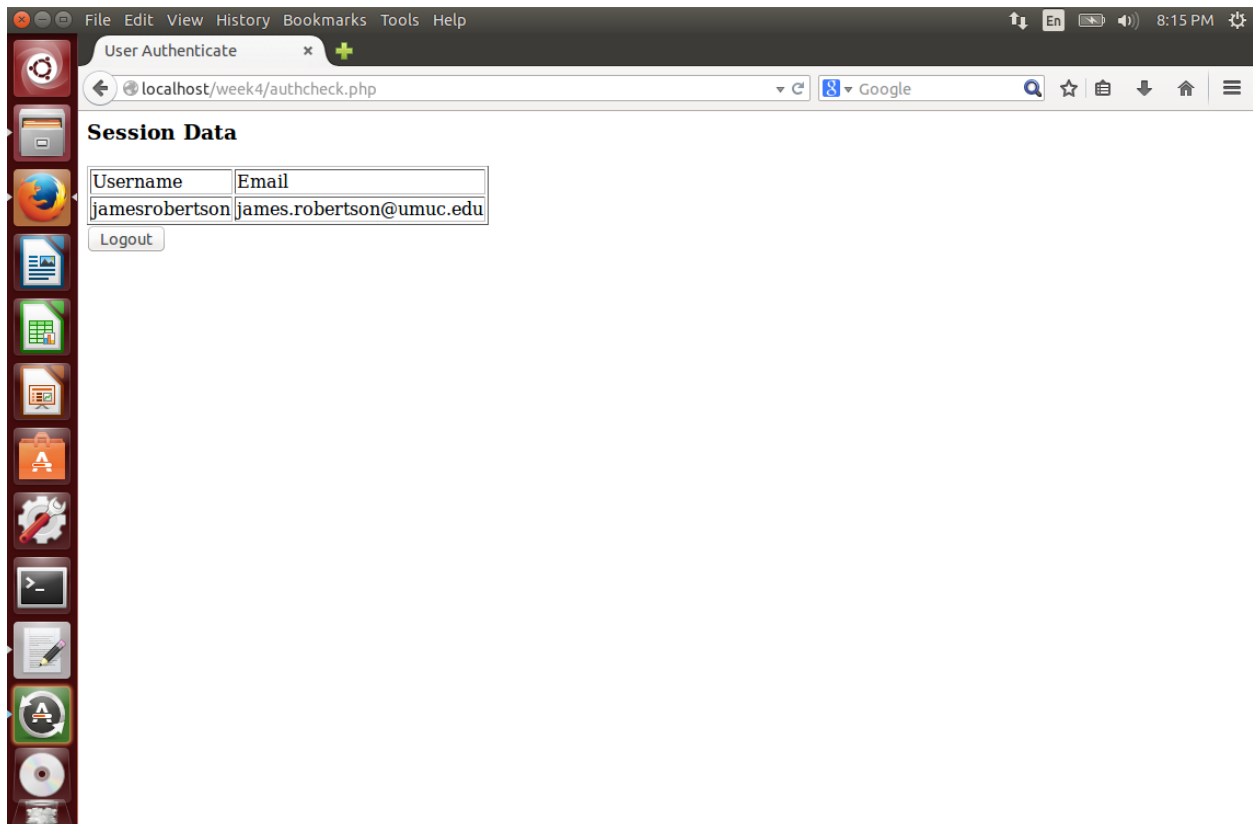
6. To run the application, type the localhost/week4/loginAuth.html into your Browser URL path.



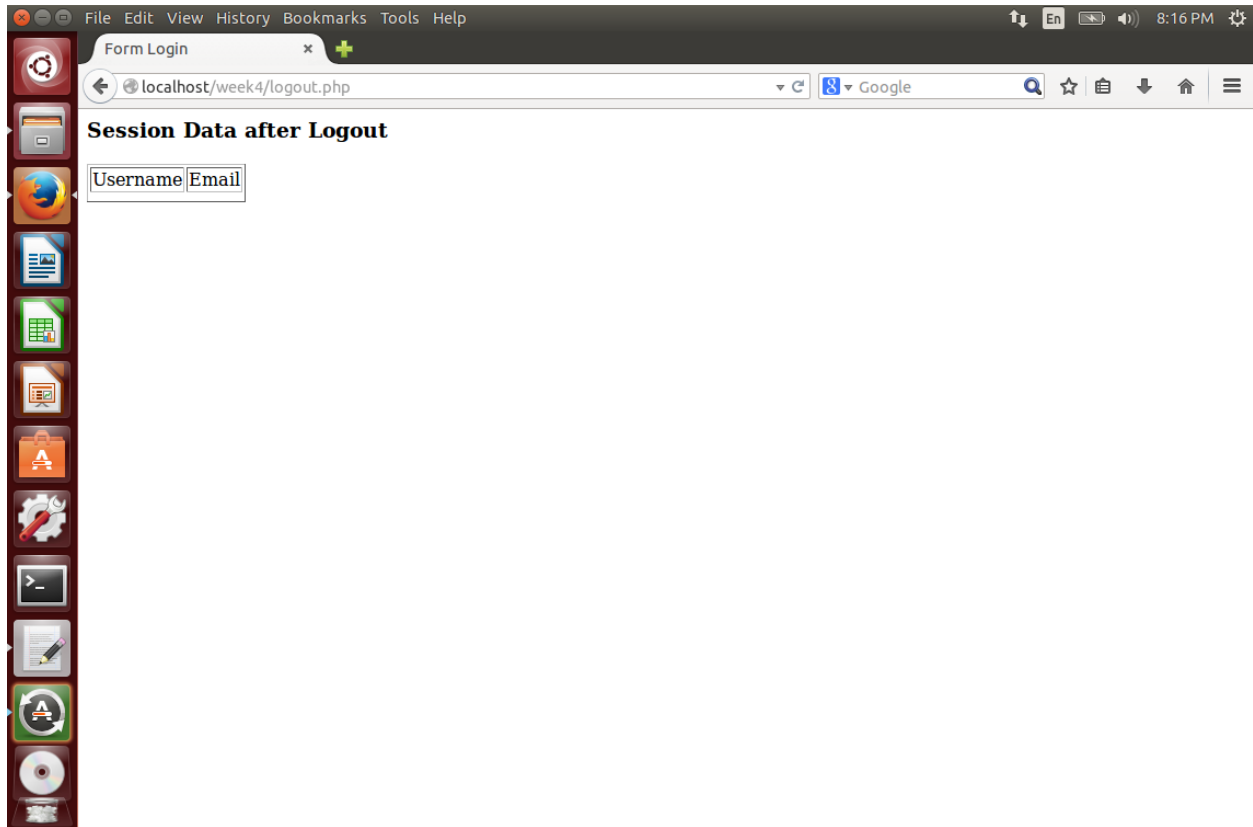
7. Type in a username and email address of your choice.



8. Click on Submit to verify the session variables were saved.



9. Click on logout to unset the session variables.



10. As you analyze the code and experiment with your own variations for this example, be sure to note, the use of the `$_SESSION['appusername']` to store session variables. Also, note you need to use `session_start()`; for session variables to have any meaning and to function properly. Also, note you use the `unset()` to remove or clear session variables.

Lab submission details:

As part of the submission for this Lab, you will create your own Web application to store and use session variables in a simple e-Commerce store.

Specifically, you will create an e-Commerce application using PHP and HTML that allows a user to login to a website, select several products for purchase and then place an order for those items selected. The type of products and price you use for the store are up to you.

The following guidelines should be used in your design and development:

1. The Login form should consist of fields for username, email address and password.
2. After login, a welcome message should appear providing a simple order form with at least 10 products of your choice. Be sure to include an image of the product and price.
3. Users can shop your store for up to 30 minutes before the session will expire.
4. Once all products are selected, the user should be able to checkout and pay for their purchases.

5. The payment page should display the username, email address and the products and total price of the purchases.
6. A "Purchase" button should be available to indicate the product was purchased and should generate a "Thank you" message of your choice.
7. On the Thank you message page, an option for logging out and essentially unsetting the stored sessions variables should be available and implemented.

Feel free to add additional HTML and PHP elements to enhance your web application. Create screen shots showing the successful running of your application.

For your deliverables, you should submit a zip file containing your word document (or PDF file) with screen shots of the application running successfully along with your PHP web application file.

Include your full name, class number and section and date in the document.

Grading Rubric:

Attribute	Meets	Does not meet
PHP App	<p>8 points</p> <p>The Login form includes fields for username, email address and password. (1 points)</p> <p>After login, a welcome message appears providing a simple order form with at least 10 products of your choice. The form Includes an image of the product and price. (2 points)</p> <p>Sessions expire after 30 minutes. (1 point)</p> <p>Once all products are selected, the user is able to checkout and pay for their purchases. (1 point)</p> <p>The payment page displays the username, email address and the products and total price of the purchases. (1 point)</p> <p>A "Purchase" button indicates the product was purchased and generates a "Thank you" message of your choice. (1 point)</p>	<p>0 points</p> <p>The Login form does not include fields for username, email address and password.</p> <p>After login, a welcome message does not appears providing a simple order form with at least 10 products of your choice. The form does not include an image of the product and price.</p> <p>Sessions do not expire after 30 minutes.</p> <p>Once all products are selected, the user is not able to checkout and pay for their purchases.</p> <p>The payment page does not display the username, email address or the products and total price of the purchases.</p> <p>A "Purchase" button does not indicate the product was purchased or generate a "Thank you" message of your choice.</p> <p>On the Thank you message page, an option for logging out and essentially unsetting the stored sessions variables is not implemented.</p>

	<p>On the Thank you message page, an option for logging out and essentially unsetting the stored sessions variables is implemented. (1 point)</p>	
<p>Documentation and submission</p>	<p>2 points Submits a zip file containing your word document and all web files. (1 point)</p> <p>Includes labeled, screen captures of the Virtual Machine running the Web/php page on your host machine for each command used. (0.5 points)</p> <p>Title page includes your full name, class number and section and date. Document is neat, well-organized and free from spelling and grammar errors. (0.5 points)</p>	<p>0 points Does not submit a zip file containing your word document and all web files.</p> <p>Does not include labeled, screen captures of the Virtual Machine running the Web/php page on your host machine for each command used.</p> <p>Title page does not include your full name, class number and section and date. Document is not neat, well-organized and free from spelling and grammar errors.</p>