Laboratory Exercise #	3 (Sessions and MySQLi)	Date	
Name		Section	

Project Name / File Name: LE3_SurnameFirstNameInitial.zip (e.g. LE3_ResuelloG.zip)
In your MyMapua drive "IT135-8L" repository, create a folder named "LE3" where you will upload your submission for this laboratory exercise. In Blackboard, only submit the link to the LE3 folder. Make sure you enabled access settings.

1. Create a home page (home.php), registration page (register.php) and a login page (login.php) for your store. You only need to capture username and password from your users.
2. Creating the database and tables:
   a. Type localhost/phpmyadmin in the URL bar

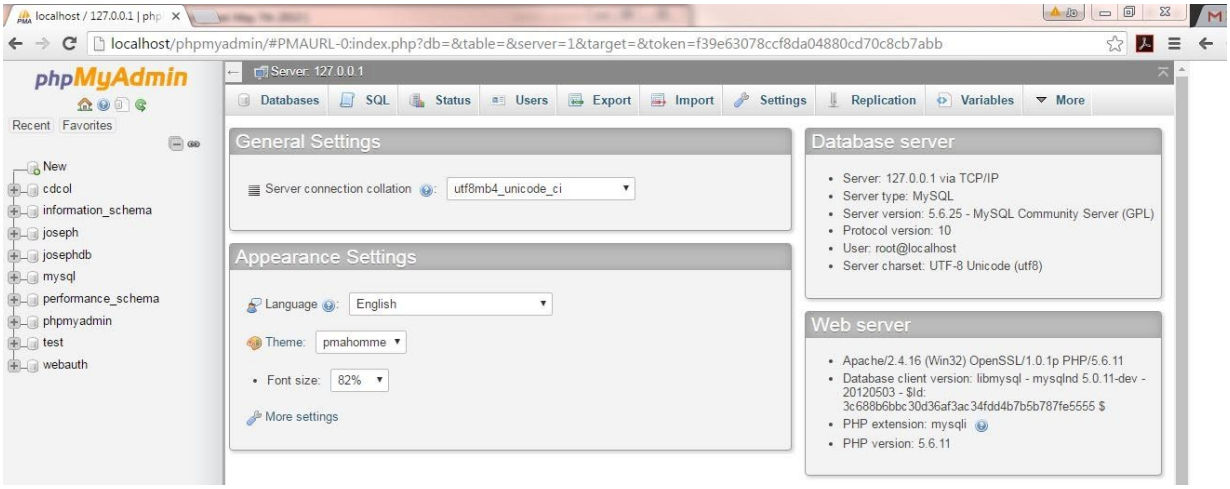

Figure 1. localhost/phpmyadmin page

   b. Click the Database tab
   c. In the Create the database field box - input the database name, (**deliverydb**) the click the Create button.

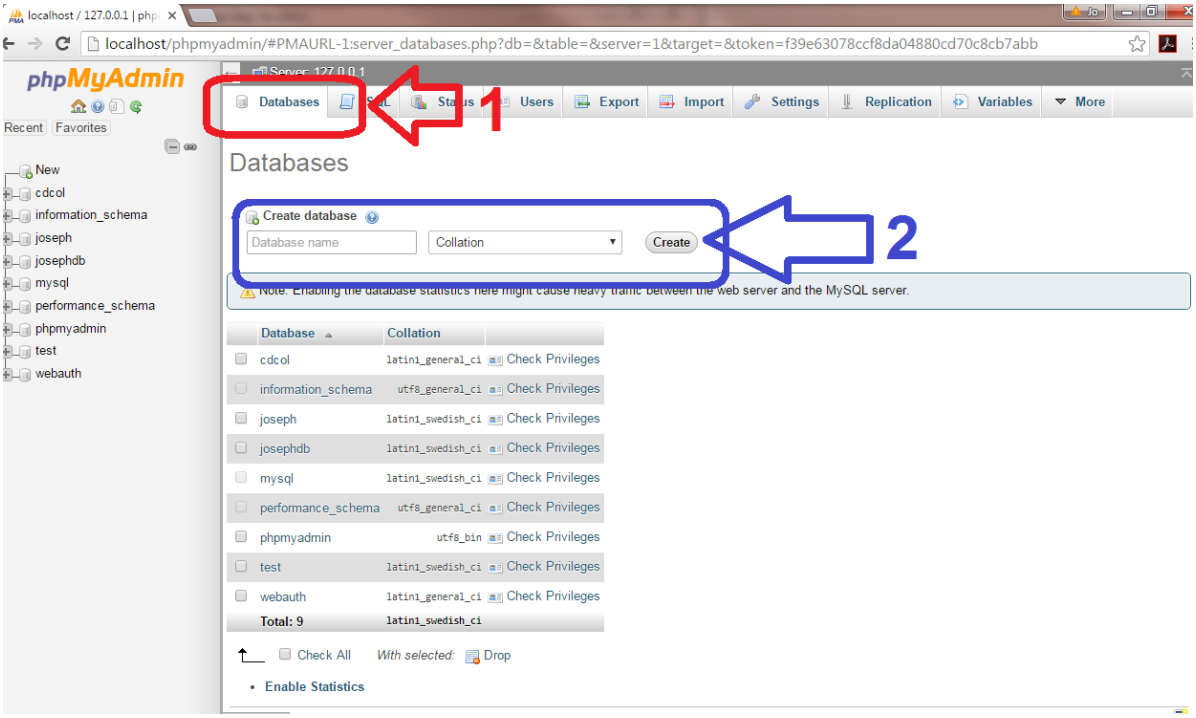

Figure 2. localhost/phpmyadmin database page

d. The database is now ready to be used. Create the table for the users and display information. Select and click the **deliverydb (1)** in the database list (on the left hand side) and then create a table named **users (2) with 3 columns (3),** then click **Go (4)**.

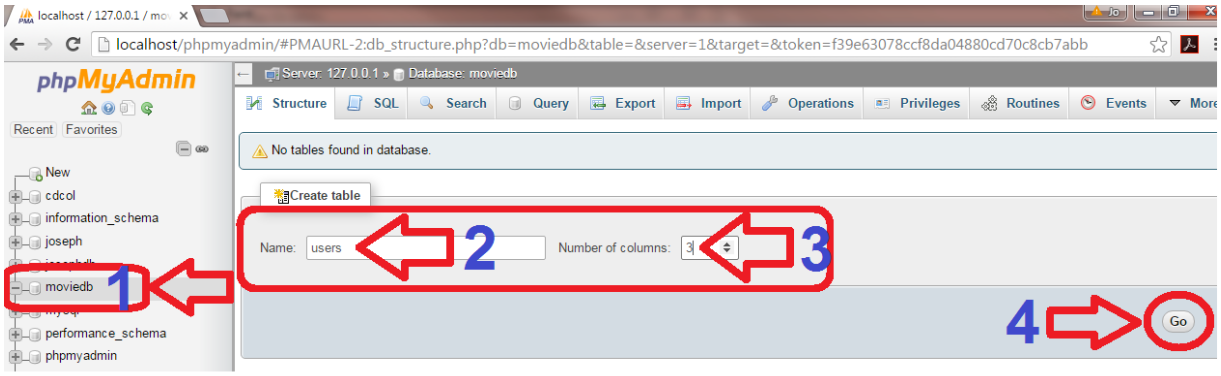

Figure 3. localhost/phpmyadmin database CREATE TABLE page

For the table's structure, make sure to have the following fields then click on save:

- Format: Column Name - Type - Length - Null Property - Other Properties
- id - INT - N/A - Not Null - Auto Increment
- username - varchar - 50 - Not null
- password - varchar - 50 - Not null

e. In the table structure, fill out the following fields as shown below:

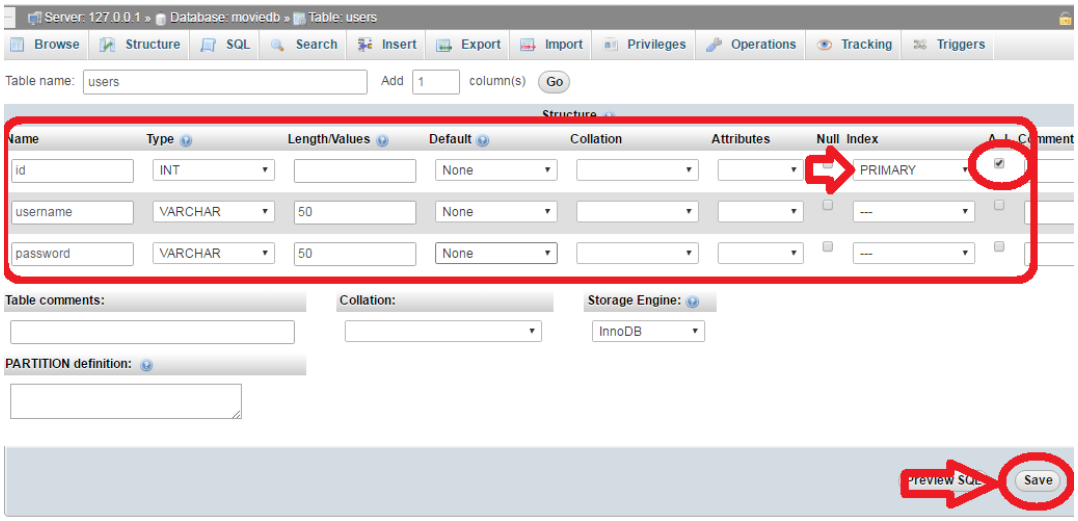

Figure 4. localhost/phpmyadmin database table STRUCTURE page for USERS

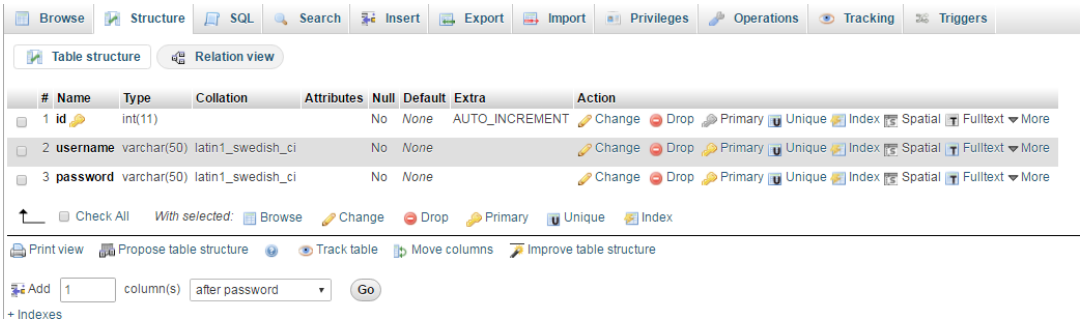

Figure 5. localhost/phpmyadmin database table structure page for USERS

f. Create another table named **list with 7 columns** as shown below
Next, create another table named *list* with 7 columns and for the table's structure:

- id - INT - N/A - Not Null - Auto Increment
- details - text - Not null
- date_posted - varchar - 30 - Not null
- time_posted - Time - Not null
- date_edited - varchar - 30 - Not null
- time_edited - Time - Not null
- public -varchar - 5 - Not null

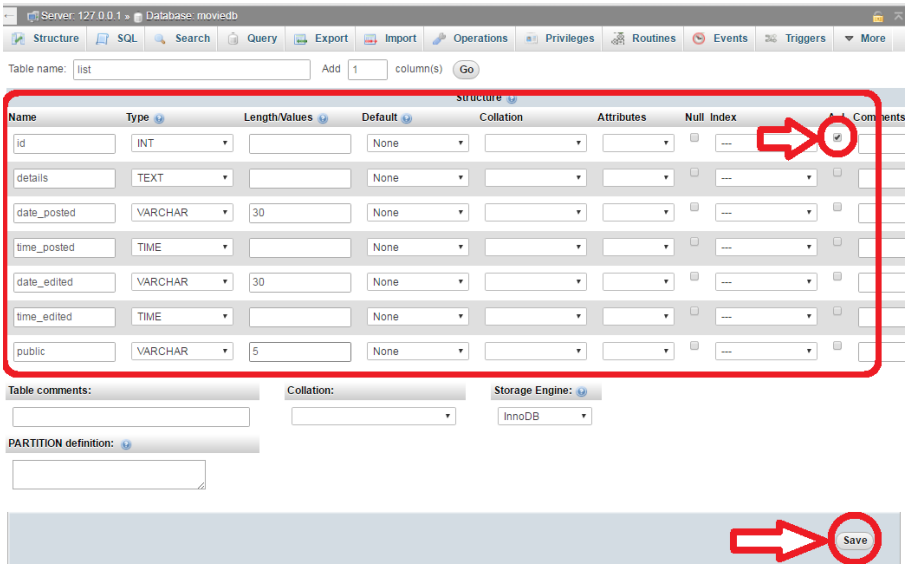

Figure 6. localhost/phpmyadmin database table structure page for LIST

3. Adding users to the database:

register.php

```
<html>
   <head>
      <title>My Online Store</title>
   </head>
   <body>
         <center>
      <h2>Registration Page</h2>
      <form action="register.php" method="POST">
      <table>
                  <tr>
                        <td>Enter Username:
                        <td><input type="text" name="username" required="required" />
                  </tr>
                  <tr>
                        <td>Enter Password:
                        <td><input type="password" name="password" required="required" />
                  </tr>
      </table><br>
      <input type="submit" value="Register"/><br/><br/>
                  <a href="login.php">Have an Account? Login Here!</a>
         </center>
   </body>
</html>

<?php
if($_SERVER["REQUEST_METHOD"] == "POST"){
 $username = ($_POST['username']);
 $password = ($_POST['password']);

 echo "Username entered is: " . $username. "<br/>";
 echo "Password entered is: " . $password;
 }
?>
```

Some Notes:

$_SERVER["REQUEST_METHOD"] == "POST" - checks if the form has received a POST method when the submit button has been clicked. The POST method is created in the html from the method="POST". Click here for the form method reference.

$_POST["] - gets the name coming from a POST method. This just simply gets the input based on the name from the form. In our case it's username and password.

mysql_real_escape_string() - encapsulates the input into a string to prevent inputs from SQL Injections. This ensures that your strings don't escape from unnecessary characters. Click here to learn more about SQL Injections.

Figure 7. Sample Registration page

Add the following php code shown below in the **register.php** page

```
<?php
 if($_SERVER["REQUEST_METHOD"] == "POST")
 {
        $username = ($_POST['username']);
        $password = ($_POST['password']);
        $bool = true;
        $db_name = "deliverydb";
        $db_username = "root";
        $db_pass = "";
        $db_host = "localhost";
        $con = mysqli_connect("$db_host","$db_username","$db_pass", "$db_name") or
die(mysqli_error()); //Connect to server
        $query = "SELECT * from users";
        $results = mysqli_query($con, $query); //Query the users table
        while($row = mysqli_fetch_array($results)) //display all rows from query
        {
                $table_users = $row['username']; // the first username row is passed on to
$table_users, and so on until the query is finished
                if($username == $table_users) // checks if there are any matching fields
                {
                        $bool = false; // sets bool to false
                        Print '<script>alert("Username has been taken!");</script>'; //Prompts the user
                        Print '<script>window.location.assign("register.php");</script>'; // redirects to
register.php
                }
        }
        if($bool) // checks if bool is true
        {
                mysqli_query($con, "INSERT INTO users (username, password) VALUES
('$username','$password')"); //Inserts the value to table users
                Print '<script>alert("Successfully Registered!");</script>'; // Prompts the user
                Print '<script>window.location.assign("register.php");</script>'; // redirects to
register.php
        }
 }
?>
```

Some notes:

mysqli_connect("Server name","Server Username","Server Password") - The syntax used to connect to our XAMPP server. localhost or 127.0.0.1 is the name of the server. The default username is root and no password for default.

mysqli_select_db("database name") - Selects the database to be used.

or die('Message') - Displays the error message if the condition wasn't met.

mysqli_query('sql query') - does the SQL queries. Click here for some SQL query samples. The again, i'm not here to discuss about mySQL.

mysqli_fetch_array('query') - fetches all queries in the table to display or manipulate data. It is placed in a while loop so that it would query all rows. Take note that only 1 row is queried per loop that's why a while loop is necessary.

$row['row name'] - the value of the column in the current query. It is represented as an array. In our case $row is the name of the variable for our row in the loop.

4. For user log-in, create a page named checklogin.php. Take note that the form action has this <form action = "checklogin.php" method= "POST">

checklogin.php

```
<?php
 session_start();
 $username = ($_POST['username']);
 $password = ($_POST['password']);
 $db_name = "deliverydb";
 $db_username = "root";
 $db_pass = "";
 $db_host = "localhost";
 $con = mysqli_connect("$db_host","$db_username","$db_pass", "$db_name") or
die(mysqli_error()); //Connect to server
 $query = "SELECT * from users WHERE username='$username'";
 $results = mysqli_query($con, $query); //Query the users table if there are matching rows equal to
$username
 $exists = mysqli_num_rows($con, $query); //Checks if username exists
 $table_users = "";
 $table_password = "";
 if($results != "") //IF there are no returning rows or no existing username
 {
        while($row = mysqli_fetch_assoc($results)) //display all rows from query
        {
                $table_users = $row['username']; // the first username row is passed on to
$table_users, and so on until the query is finished
                $table_password = $row['password']; // the first password row is passed on to
$table_users, and so on until the query is finished
        }
        if(($username == $table_users) && ($password == $table_password)) // checks if there are
any matching fields
        {
                if($password == $table_password)
                {
                        $_SESSION['user'] = $username; //set the username in a session. This serves
as a global variable
```

```
                        header("location: home.php"); // redirects the user to the authenticated home
page
                }
        }
        else
        {
                Print '<script>alert("Incorrect Password!");</script>'; //Prompts the user
                Print '<script>window.location.assign("login.php");</script>'; // redirects to login.php
        }
 }
 else
 {
         Print '<script>alert("Incorrect Username!");</script>'; //Prompts the user
         Print '<script>window.location.assign("login.php");</script>'; // redirects to login.php
 }
?>
```

Some notes:
session_start() - Starts the session. This is usually done on authenticated pages. The reason why we used this is because this is required for the $_SESSION[''].

mysql_num_rows() - This returns an integer. This counts all the rows depending on the query.

$_SESSION['name'] - Serves as the session name for the entire session. This is relatively similar to public variables in object-oriented programming. We will be using this for validating whether the user is authenticated or not.

5. Create the home page home.php as given below:

home.php

```
<html>
 <head>
  <title>My Food Delivery Store</title>
 </head>
 <?php
 session_start(); //starts the session
 if($_SESSION['user']){ //checks if user is logged in
 }
 else{
  header("location:index.php"); // redirects if user is not logged in
 }
 $user = $_SESSION['user']; //assigns user value
 ?>
 <body>
  <h2>Home Page</h2>
  <p>Hello <?php Print "$user"?>!</p> <!--Displays user's name-->
  <a href="logout.php">Click here to logout</a><br/><br/>
  <form action="add.php" method="POST">
   Add more to list: <br/>
   Details: <input type="text" name="details"/><br/>
   Public Post? <input type="checkbox" name="public[]" value="yes"/><br/>
   <input type="submit" value="Add to list"/>
  </form>
  <h2 align="center">My list</h2>
  <table border="1px" width="100%">
```

```
   <tr>
    <th>Id</th>
    <th>Details</th>
    <th>Post Time</th>
    <th>Edit Time</th>
    <th>Edit</th>
    <th>Delete</th>
    <th>Public Post</th>
   </tr>
   <?php
    $con = mysqli_connect("localhost", "root", "", "deliverydb") or die(mysqli_error()); //Connect to
server
    $query = mysqli_query($con, "Select * from list"); // SQL Query
    while($row = mysqli_fetch_array($query))
    {
     Print "<tr>";
      Print '<td align="center">'. $row['id'] . "</td>";
      Print '<td align="center">'. $row['details'] . "</td>";
      Print '<td align="center">'. $row['date_posted']. " - ". $row['time_posted']."</td>";
      Print '<td align="center">'. $row['date_edited']. " - ". $row['time_edited']. "</td>";
      Print '<td align="center"><a href="edit.php?id='. $row['id'] .'">edit</a> </td>';
      Print '<td align="center"><a href="#" onclick="myFunction('.$row['id'].')">delete</a> </td>';
      Print '<td align="center">'. $row['public']. "</td>";
     Print "</tr>";
    }
   ?>
  </table>
  <script>
   function myFunction(id)
   {
   var r=confirm("Are you sure you want to delete this record?");
   if (r==true)
     {
      window.location.assign("delete.php?id=" + id);
     }
   }
  </script>
 </body>
</html>
```

Some notes:
session_start() - Basically starts the session. Required for $_SESSION[''].

header() - redirects the user.

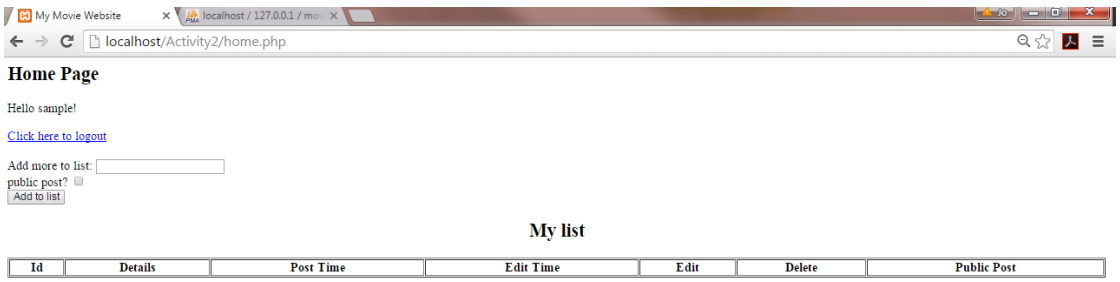

Figure 8. Sample output page of the Home page

6. For user logout, create logout.php as shown below:

logout.php

```
<?php
 session_start();
 session_destroy();
 header("location:index.php");
?>
```

Some notes:

session_destroy() - simply remove's all session's meaning, the value of $_SESSION[''] will be removed and **header()** - will simply redirect it to the home

7. To add data to the list table, create add.php as shown below. This page will add values to the list table by using the <form action="add.php" method="POST">

add.php

```
<?php
 session_start();
 if($_SESSION['user']){
        }
 else{
 header("location:index.php");
        }
 if($_SERVER['REQUEST_METHOD'] = "POST") //Added an if to keep the page secured
        {
 $details = ($_POST['details']);
 $time = strftime("%X");//time
 $date = strftime("%B %d, %Y");//date
 $decision ="no";
 $con = mysqli_connect("localhost", "root", "", "deliverydb") or die(mysqli_error()); //Connect to
server
 foreach($_POST['c'] as $each_check) //gets the data from the checkbox
                {
 if($each_check !=null ){ //checks if the checkbox is checked
        $decision = "yes"; //sets the value
                        }
                }
mysqli_query($con, "INSERT INTO list (details, date_posted, time_posted, public) VALUES
('$details','$date','$time','$decision')"); //SQL query
 header("location: home.php");
        }
 else
        {
 header("location:home.php"); //redirects back to home
        }
?>
```

Some notes:
foreach() - gets the value of the checkbox. As you will notice, the checkbox format in the form **isname="checkbox[]".** - To get data from checkbox, it has to be instantiated as an array. Doing so would make it possible to get data from multiple checkboxes

8. To edit values on the list table, create the edit.php page as shown below:

edit.php

```
<html>
	<head>
		<title>My Online Store</title>
	</head>
<?php
session_start(); //starts the session
if($_SESSION['user']){ //checks if user is logged in
	}
else{
header("location:index.php "); // redirects if user is not logged in
	}
$user = $_SESSION['user']; //assigns user value
$id_exists = false;
?>
	<body>
		<h2>Home Page</h2>
		<p>Hello <?php Print "$user"?>!</p> <!--Displays user's name-->
		<a href="logout.php">Click here to logout</a><br/><br/>
		<a href="home.php">Return to Home page</a>
		<h2 align="center">Currently Selected</h2>
		<table border="1px" width="100%">
			<tr>
				<th>Id</th>
				<th>Details</th>
				<th>Post Time</th>
				<th>Edit Time</th>
				<th>Public Post</th>
			</tr>
<?php
if(!empty($_GET['id']))
				{
$id = $_GET['id'];
$_SESSION['id'] = $id;
$id_exists = true;
$con = mysqli_connect("localhost", "root", "", "deliverydb") or die(mysqli_error()); //Connect to server
$sql = "Select * from list Where id='$id'";
$query = mysqli_query($con, $sql); // SQL Query
$count = mysqli_num_rows($query);
if($count > 0)
					{
while($row = mysqli_fetch_array($query))
						{
Print "<tr>";
Print '<td align="center">'. $row['id'] . "</td>";
Print '<td align="center">'. $row['details'] . "</td>";
Print '<td align="center">'. $row['date_posted']. " - ". $row['time_posted']."</td>";
Print '<td align="center">'. $row['date_edited']. " - ". $row['time_edited']. "</td>";
Print '<td align="center">'. $row['public']. "</td>";
Print "</tr></table><br/>";

$details = $row['details'];
$public = $row['public'];
						}
					}
else
```

```
                                        {
 $id_exists = false;
                                        }
                                    }
 if($id_exists)
                    {
 Print '<form action="edit.php" method="POST">
                        Update List: <br/>
                        Details: <input type="text" name="details" value="'.$details.'"/><br/>
                        if($public == "yes") {
 Print 'Public Post? <input type="checkbox" name="public[]" checked/><br/>';
 }
 else {
 Print 'Public Post? <input type="checkbox" name="public[]"/><br/>';
 }
 Print'<input type="submit" value="Update List"/></form>';
                    }
 else
                    {
 Print '<h2 align="center">There is no data to be edited.</h2>';
                    }
 ?>
        </body>
</html>

<?php
 if($_SERVER['REQUEST_METHOD'] == "POST")
 {
        $con = mysqli_connect("localhost", "root", "", "deliverydb") or die(mysqli_error()); //Connect to
server
        $details = ($_POST['details']);
        $public = "no";
        $id = $_SESSION['id'];
        $time = strftime("%X");//time
        $date = strftime("%B %d, %Y");//date
        foreach($_POST['public'] as $list)
                        {
        if($list != null)
                                    {
        $public = "yes";
                                    }
                        }
        foreach($_POST['sale'] as $list)
                        {
        if($list != null)
                                    {
        $sale = "yes";
                                    }
                        }
        mysqli_query($con, "UPDATE list SET details='$details', public='$public', date_edited='$date',
time_edited='$time' WHERE id='$id'") ;
        header("location: home.php");
 }
?>
```

Some notes:
!empty() - a method that checks if the value is not empty. The syntax can be reversed if you want to check if it's empty by removing the explanation point (!), therefore it's syntax would be empty().
$_GET[''] - Used to get the value from the parameter. In our case, we use id as our URL parameter so the syntax would be $_GET['id'].
$id_exists - the variable that checks whether the given id exists.
$_SESSION['id'] - we place the value of id into session to use it on another file.

9. To delete values in the list table, it's relatively the same as what we have done on edit but what differs is just the SQL statement. Instead of using UPDATE, we will be using the DELETE syntax. In deleting records, we must prompt people making sure that they really want to delete the record, so we have added a little javascript in home.php.

   As you may have noticed, the href is "#" and added an onclick function for Javascript for the method of myFunction and inside its parameter is the id of the row. Below the table written is the Javascript syntax wherein it prompts the user if he/she wants to delete the record. If the user confirms, the page then directs to delete.php together embedded with the value of the id. Now let us create delete.php

delete.php

```
<?php
 session_start(); //starts the session
 if($_SESSION['user']){ //checks if user is logged in
        }
 else{
 header("location:index.php"); // redirects if user is not logged in
        }
 if($_SERVER['REQUEST_METHOD'] == "GET")
        {
 $con = mysqli_connect("localhost", "root", "", "deliverydb") or die(mysqli_error()); //Connect to server
 $id = $_GET['id'];
 mysqli_query($con, "DELETE FROM list WHERE id='$id'");
 header("location: home.php");
        }
?>
```

Laboratory Exercise Score Sheet

1	Intelligent naming of variables and use of comment	10	
2	Layout and Design of Screen	20	
3	Correct use of sessions and database connection	50	
4	Finished within time frame (180 minutes)	10	
5	Correct Filename	10	
	Total	**100**	