

LAMP Apps

Overview

This lab walks you through using Linux, Apache, MySQL and PHP (LAMP) to create simple, yet very powerful PHP applications connected to a MySQL database. For developers using Windows, the acronym becomes WAMP (Linux is replaced by Windows). The basics of inserting, updating, deleting and selecting from MySQL using PHP forms will be provided. Some “bad” security practices that lead to SQL injection vulnerabilities will be exposed as well as some techniques to mitigate these issues.

Learning Outcomes:

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

1. Insert data into a MySQL database using PHP forms
2. Query existing data in a MySQL database using PHP forms
3. Delete data from a MySQL database using PHP forms
4. Update data in a MySQL database using PHP forms

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 LAMP application and all associated files should be submitted.

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

MySQL Username: sdev_owner

MySQL password: sdev300

MySQL database: sdev

Part 1 – Insert data into a MySQL database using PHP forms

In this exercise we will create a small table in MySQL and then use a PHP form to insert collected from the user into the form. We will first use a technique very susceptible to SQL injection and then a better approach using prepared statements.

1. Assuming you have already launched and logged into your SDEV32Bit Virtual Machine (VM) from the Oracle VirtualBox, pen up the terminal by clicking on the terminal icon.



2. To start the MySQL database type the following the terminal prompt:

`mysql -u sdev_owner -p`

When prompted for the password enter sdev300

```
umucsdev@umucsdev-VirtualBox: ~  
umucsdev@umucsdev-VirtualBox:~$ mysql -u sdev_owner -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 36  
Server version: 5.5.38-0ubuntu0.14.04.1 (Ubuntu)  
  
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

3. To display the available databases type the following at the mysql prompt:
show databases;

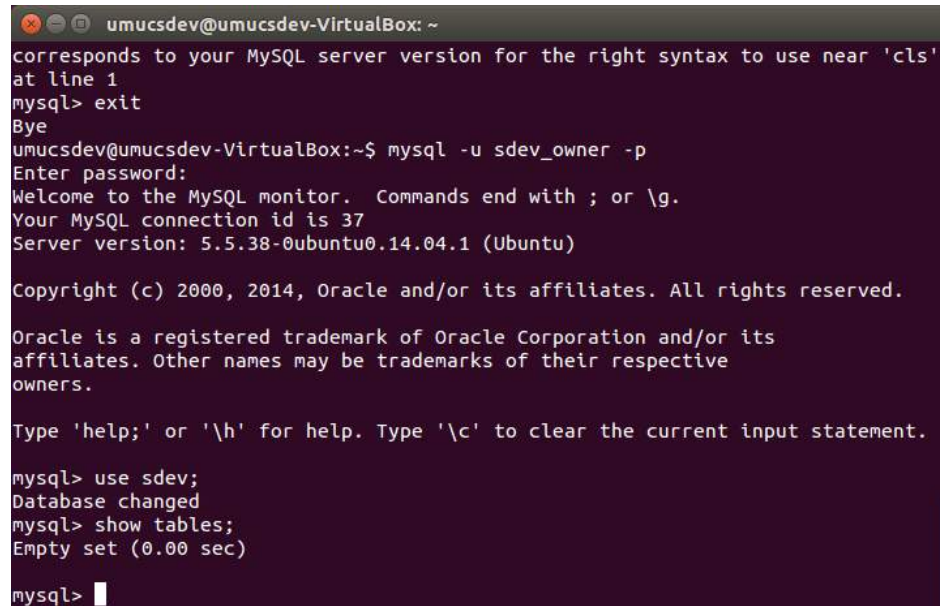
```
umucsdev@umucsdev-VirtualBox: ~  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 36  
Server version: 5.5.38-0ubuntu0.14.04.1 (Ubuntu)  
  
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sdev |  
+-----+  
4 rows in set (0.03 sec)  
  
mysql> 
```

4. The database we will be using for this course is sdev. To use this database, type the following at the mysql prompt:
use sdev;

```
umucsdev@umucsdev-VirtualBox: ~  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sdev |  
+-----+  
4 rows in set (0.03 sec)  
  
mysql> use sdev;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> 
```

5. To display the current tables in the sdev database, type the following command at the mysql prompt:

show tables;



```
umucsdev@umucsdev-VirtualBox: ~  
corresponds to your MySQL server version for the right syntax to use near 'cls'  
at line 1  
mysql> exit  
Bye  
umucsdev@umucsdev-VirtualBox:~$ mysql -u sdev_owner -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 37  
Server version: 5.5.38-0ubuntu0.14.04.1 (Ubuntu)  
  
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> use sdev;  
Database changed  
mysql> show tables;  
Empty set (0.00 sec)  
  
mysql>
```

You may already have some tables in your database. If so, the names of those tables would be displayed. If not, you would see Empty set as illustrated above.

6. Create a Students table in the SDEV database, if one does not already exist:

```
use sdev;  
  
// Create a student table  
CREATE TABLE Students (  
    tychoName varchar(30) primary key,  
    firstName varchar(30),  
    lastName varchar(30),  
    eMail varchar(60)  
);
```

7. Next, we will create the PHP code that will provide an HTML form and response for entering data into the database table from the form. Type, or copy and paste from the code examples, the following code into your text editor and save as InsertApp.php. This code has many components including the use of PHP classes, reading parameters from files and other functionality. The code is relative long and may take some experimentation and analysis for full understanding. You should review and tinker with all aspects of the code to become comfortable with the functionality.

```
<html>  
<head>
```

```

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Create Student </title>
</head>
<body OnLoad="document.createstudent.firstname.focus();">

<?php
    if(isset($_POST["CreateSubmit"]))
    {
        validate_form();
    }
    else
    {
        $messages = array();
        show_form($messages);
    }
?>

<?php
function show_form($messages) {

    // Assign post values if exist
    $firstname="";
    $lastname="";
    $wsname="";
    $email="";
    if (isset($_POST["firstname"]))
        $firstname=$_POST["firstname"];
    if (isset($_POST["lastname"]))
        $lastname=$_POST["lastname"];
    if (isset($_POST["wsname"]))
        $wsname=$_POST["wsname"];
    if (isset($_POST["email"]))
        $email=$_POST["email"];

    echo "<p></p>";
    echo "<h2> Enter New Student</h2>";
    echo "<p></p>";
    ?>
    <h5>Complete the information in the form below and click Submit to
create your account. All fields are required.</h5>
    <form name="createstudent" method="POST" action="InsertApp.php">
    <table border="1" width="100%" cellpadding="0">
        <tr>
            <td width="157">Firstname:</td>
            <td><input type="text" name="firstname" value='<?php
echo $firstname ?>' size="30"></td>
        </tr>
        <tr>
            <td width="157">Lastname:</td>
            <td><input type="text" name="lastname" value='<?php
echo $lastname ?>' size="30"></td>
        </tr>
        <tr>
            <td width="157">WebTycho username:</td>

```

```

        <td><input type="text" name="wsname" value='<?php
echo $wsname ?>' size="30"></td>
    </tr>
    <tr>
        <td width="157">Email:</td>
        <td><input type="text" name="email" value='<?php echo
$email ?>' size="30"></td>
    </tr>
    <tr>
        <td width="157"><input type="submit" value="Submit"
name="CreateSubmit"></td>
        <td>&nbsp;</td>
    </tr>
</table>
</form>

    <?php
} // End Show form
?>

<?php
function validate_form()
{
    $messages = array();
    $redisplay = false;
    // Assign values
    $firstname = $_POST["firstname"];
    $lastname = $_POST["lastname"];
    $wsname = $_POST["wsname"];
    $email = $_POST["email"];

    $student = new StudentClass($firstname,$lastname,$email,$wsname);
    $count = countStudent($student);

    // Check for accounts that already exist and Do insert
    if ($count==0)
    {
        $res = insertStudent($student);
        echo "<h3>Welcome to UMUC!</h3> ";
    }
    else
    {
        echo "<h3>A student account with that WenTycho username already
exists.</h3> ";
    }
}

function countStudent ($student)
{
    // Connect to the database
    $mysqli = connectdb();
    $firstname = $student->getFirstname();
    $lastname = $student->getLastName();
    $wsname = $student->getTychoName();
    $email = $student->getEmail();

```

```

        // Connect to the database
        $mysqli = connectdb();

        // Define the Query
        // For Windows MYSQL String is case insensitive
        $Myquery = "SELECT count(*) as count from Students
                    where tychoName='$wsname'";

        if ($result = $mysqli->query($Myquery))
        {
            /* Fetch the results of the query */
            while( $row = $result->fetch_assoc() )
            {
                $count=$row["count"];
            }

            /* Destroy the result set and free the memory used for it */
            $result->close();
        }

        $mysqli->close();

        return $count;
    }

function insertStudent ($student)
{
    // Connect to the database
    $mysqli = connectdb();

    $firstname = $student->getFirstname();
    $lastname = $student->getLastname();
    $wsname = $student->getTychoName();
    $email = $student->getEmail();

    // Now we can insert
    $Query = "INSERT INTO Students
              (firstName,lastName,eMail,tychoName)
              VALUES ('$firstname', '$lastname', '$email', '$wsname')";

    $Success=false;
    if ($result = $mysqli->query($Query)) {
        $Success=true;
    }
    $mysqli->close();

    return $Success;
}

function getDbparms()
{
    $trimmed = file('parms/dbparms.txt', FILE_IGNORE_NEW_LINES |
FILE_SKIP_EMPTY_LINES);
    $key = array();

```

```

$vals = array();
foreach($trimmed as $line)
{
    $pairs = explode("=", $line);
    $key[] = $pairs[0];
    $vals[] = $pairs[1];
}
// Combine Key and values into an array
$mypairs = array_combine($key, $vals);

// Assign values to ParametersClass
$myDbparms = new
DbparmsClass($mypairs['username'], $mypairs['password'],
             $mypairs['host'], $mypairs['db']);

// Display the Paramters values
return $myDbparms;
}

function connectdb() {
    // Get the DBParameters
    $mydbparms = getDbparms();

    // Try to connect
    $mysqli = new mysqli($mydbparms->getHost(), $mydbparms->
    getUsername(),
                        $mydbparms->getPassword(), $mydbparms->getDb());

    if ($mysqli->connect_error) {
        die('Connect Error (' . $mysqli->connect_errno . ') '
            . $mysqli->connect_error);
    }
    return $mysqli;
}

class DBparmsClass
{
    // property declaration
    private $username="";
    private $password="";
    private $host="";
    private $db="";

    // Constructor
    public function __construct($myusername, $mypassword, $myhost, $mydb)
    {
        $this->username = $myusername;
        $this->password = $mypassword;
        $this->host = $myhost;
        $this->db = $mydb;
    }

    // Get methods
    public function getUsername ()
    {
        return $this->username;
    }
}

```



```

        public function getPassword ()
        {
            return $this->password;
        }

        public function getHost ()
        {
            return $this->host;
        }

        public function getDb ()
        {
            return $this->db;
        }

        // Set methods
        public function setUsername ($myusername)
        {
            $this->username = $myusername;
        }
        public function setPassword ($mypassword)
        {
            $this->password = $mypassword;
        }
        public function setHost ($myhost)
        {
            $this->host = $myhost;
        }
        public function setDb ($mydb)
        {
            $this->db = $mydb;
        }

    } // End DBparms class

    // Class to construct Students with getters/setter
    class StudentClass
    {
        // property declaration
        private $firstname="";
        private $lastname="";
        private $email="";
        private $tychoname="";

        // Constructor
        public function __construct($firstname,$lastname,$email,$tychoname)
        {
            $this->firstname = $firstname;
            $this->lastname = $lastname;
            $this->email = $email;
            $this->tychoname = $tychoname;
        }

        // Get methods
        public function getFirstname ()
        {
            return $this->firstname;
        }

        public function getLastName ()

```

```

    {
        return $this->lastname;
    }
    public function getEmail ()
    {
        return $this->email;
    }
    public function getTychoname ()
    {
        return $this->tychoname;
    }

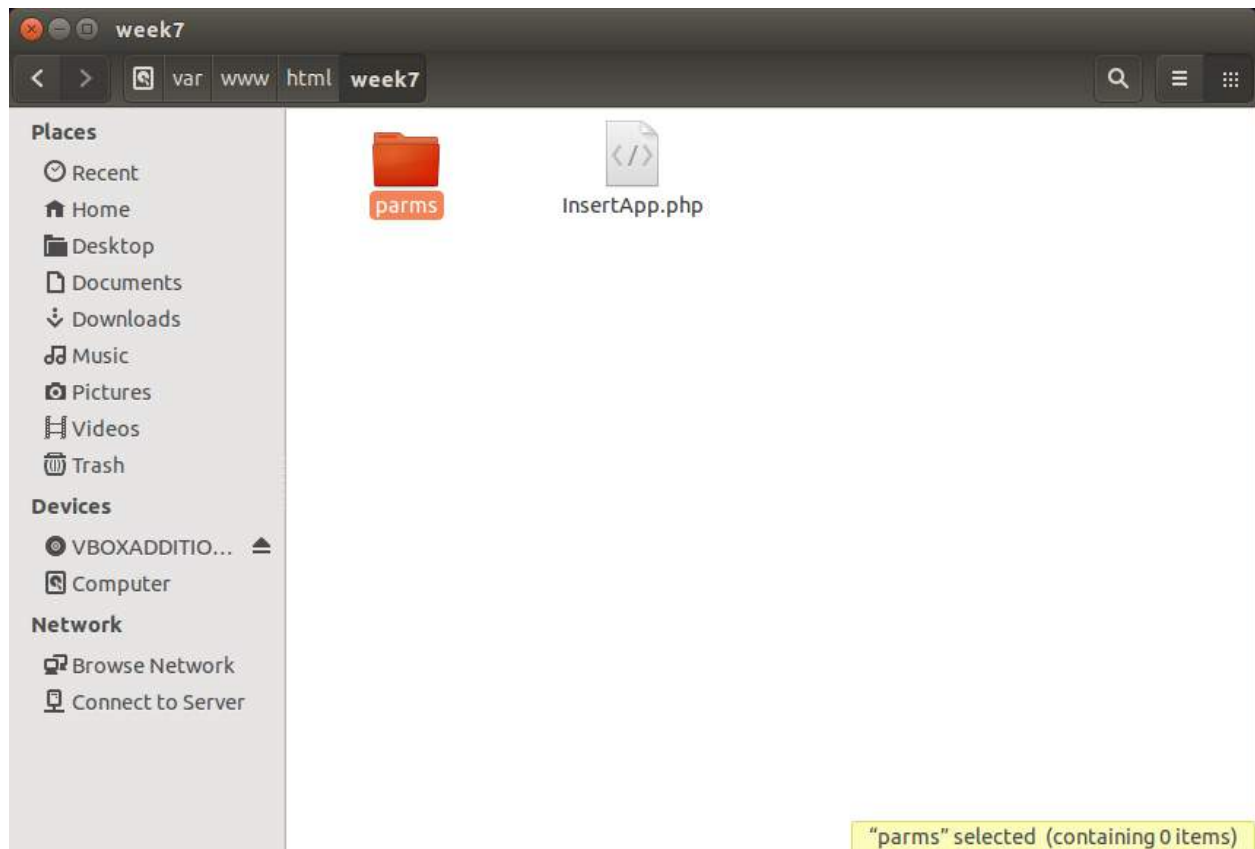
    // Set methods
    public function setFirstname ($value)
    {
        $this->firstname = $value;
    }
    public function setLastname ($value)
    {
        $this->lastname = $value;
    }
    public function setEmail ($value)
    {
        $this->email = $value;
    }
    public function setTychoname ($value)
    {
        $this->tychoname = $value;
    }

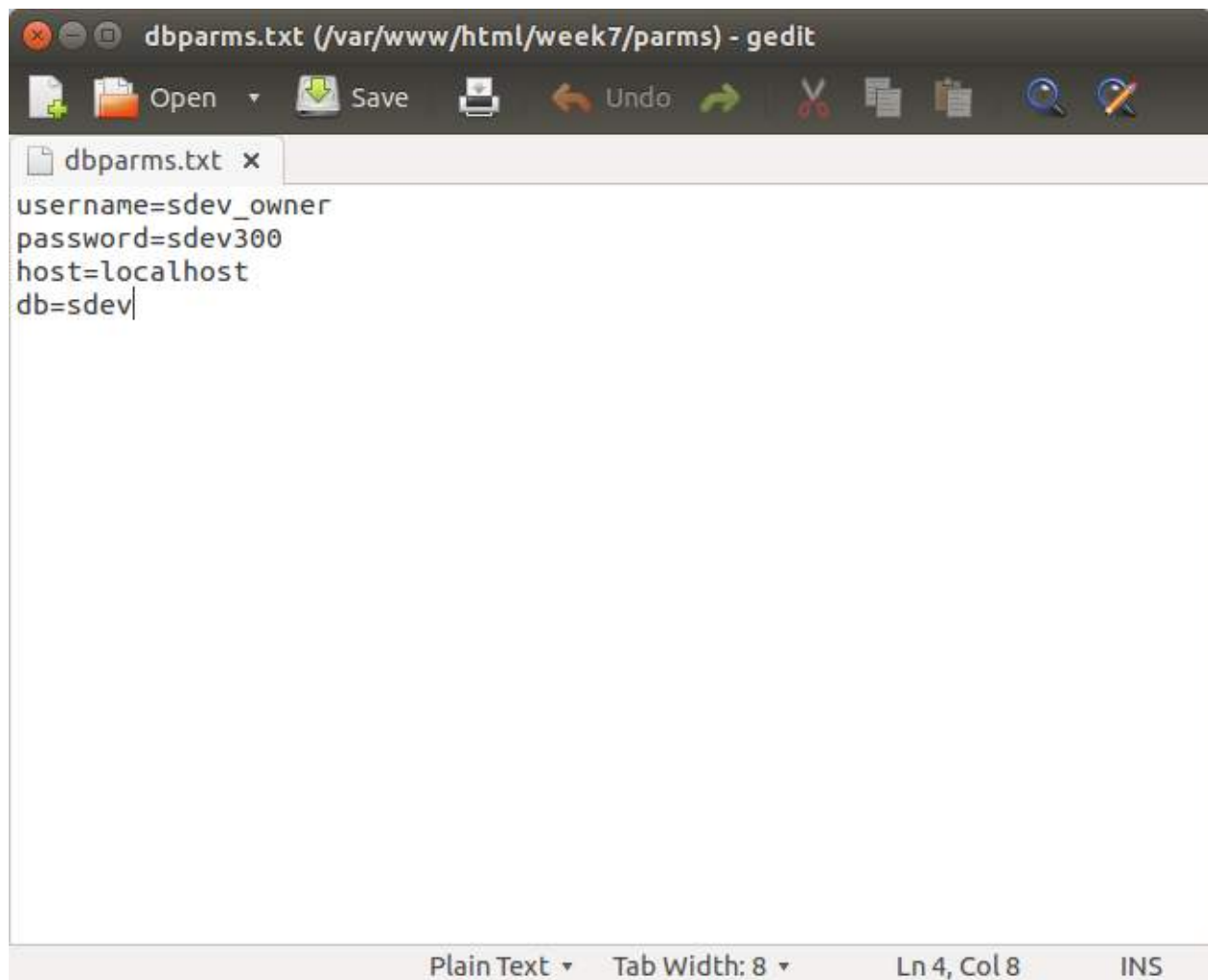
} // End Studentclass

?>
</body>
</html>

```

8. To run the code place the file in a week7 folder in the appropriate location on your VM and launch it. Note: **Be sure to create a parms folder and place the dbparms.txt file in the folder or your application will not connect to the database.**





The image shows a screenshot of a gedit text editor window. The title bar at the top reads "dbparms.txt (/var/www/html/week7/parms) - gedit". Below the title bar is a toolbar with icons for Open, Save, Print, Undo, Redo, Cut, Copy, Paste, Find, and Replace. The main text area contains the following text:

```
dbparms.txt x
username=sdev_owner
password=sdev300
host=localhost
db=sdev|
```

At the bottom of the window, there is a status bar with the following information: "Plain Text", "Tab Width: 8", "Ln 4, Col 8", and "INS".

9. Add an entry to verify a student was successfully entered.

Create Student x +

localhost/week7/insertApp.php Google

Enter New Student

Complete the information in the form below and click Submit to create your account. All fields are required.

Firstname:	Jimmy
Lastname:	johnson
WebTycho username:	jjohnson1
Email:	jjohnson@umuc.edu
Submit	

Create Student x +

localhost/week7/insertApp.php Google

Welcome to UMUC!

10. Note the following code is assuming you have honest users.

```
$Query = "INSERT INTO Students
          (firstName,lastName,eMail,tychoName)
          VALUES ('$firstname', '$lastname', '$email', '$wsname')";
```

11. Replace this with a prepared statements to help mitigate the SQL injection in the insertStudent function:

```
function insertStudent ($student)
{
    // Connect to the database
    $mysqli = connectdb();

    $firstname = $student->getFirstname();
    $lastname = $student->getLastName();
    $wsname = $student->getTychoName();
    $email = $student->getEmail();

    // Add Prepared Statement
    $Query = "INSERT INTO Students
              (firstName,lastName,eMail,tychoName)
              VALUES (?, ?, ?, ?)";

    $stmt = $mysqli->prepare($Query);

    $stmt->bind_param("ssss", $firstname, $lastname, $wsname, $email);
    $stmt->execute();

    $stmt->close();
    $mysqli->close();

    return true;
}
```

12. Note the bind statement is using “ssss” representing 4 strings. Other options include i for integer and d for double. We will use the prepared statement in the remaining examples.

Part 2 Query existing data in a MySQL database using PHP forms

Now that we have a form to Insert data into a table, we can expand and leverage the previous code to select from the database and display the results in an HTML table. We will also add a link to the Insert Table so we can demonstrate adding additional students.

1. Create the PHP code that will display the data in the Students table. Type, or copy and paste from the source code examples, the following code into your text editor and save as SelectApp.php. You should review and tinker with all aspects of the code to become comfortable with the functionality. Notice the show_form function queries the table and returns the student data for display.

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Select Student </title>
    </head>
    <body OnLoad="document.createstudent.firstname.focus();">

<?php
    show_form();

    // Provide option for inserting another student
    echo "<p></p>";
    echo "<a href=InsertApp.php> Insert Another Students </a>";

    ?>

<?php
function show_form() {

    echo "<p></p>";
    echo "<h2> Select the Student to Update</h2>";
    echo "<p></p>";
    // Retrieve the students
    $students = selectStudents();

    echo "<h3> " . "Number of Students in Database is: " .
sizeof($students) . "</h3>";
    // Loop through table and display
    echo "<table border='1'>";
    foreach ($students as $data) {
    echo "<tr>";
        echo "<td>" . $data->getFirstname() . "</td>";
        echo "<td>" . $data->getLastname() . "</td>";
        echo "<td>" . $data->getEmail() . "</td>";
        echo "<td>" . $data->getTychoname() . "</td>";
    echo "</tr>";
    }
    echo "</table>";

} // End Show form
?>

<?php

function selectStudents ()
{

    // Connect to the database
```

```

$mysqli = connectdb();

    // Add Prepared Statement
    $Query = "Select firstName,lastName,eMail,tychoName from
Students";

    $result = $mysqli->query($Query);
    $myStudents = array();
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        // Assign values
        $firstname = $row["firstName"];
        $lastname = $row["lastName"];
        $email = $row["eMail"];
        $tychoname= $row["tychoName"];

        // Create a Student instance
        $studentData = new
Studentclass($firstname,$lastname,$email,$tychoname);
        $myStudents[] = $studentData;
    }
}

    $mysqli->close();

    return $myStudents;

}

function getDbparms()
{
    $trimmed = file('parms/dbparms.txt', FILE_IGNORE_NEW_LINES |
FILE_SKIP_EMPTY_LINES);
    $key = array();
    $vals = array();
    foreach($trimmed as $line)
    {
        $pairs = explode("=", $line);
        $key[] = $pairs[0];
        $vals[] = $pairs[1];
    }
    // Combine Key and values into an array
    $mypairs = array_combine($key,$vals);

    // Assign values to ParametersClass
    $myDbparms = new
DbparmsClass($mypairs['username'],$mypairs['password'],
                $mypairs['host'],$mypairs['db']);

    // Display the Paramters values
    return $myDbparms;
}

function connectdb() {
    // Get the DBParameters
    $mydbparms = getDbparms();

```



```

        // Try to connect
        $mysqli = new mysqli($mydbparms->getHost(), $mydbparms->
>getUsername(),
                                $mydbparms->getPassword(), $mydbparms->getDb());

        if ($mysqli->connect_error) {
            die('Connect Error (' . $mysqli->connect_errno . ') '
                . $mysqli->connect_error);
        }
        return $mysqli;
    }

class DBparmsClass
{
    // property declaration
    private $username="";
    private $password="";
    private $host="";
    private $db="";

    // Constructor
    public function __construct($myusername, $mypassword, $myhost, $mydb)
    {
        $this->username = $myusername;
        $this->password = $mypassword;
        $this->host = $myhost;
        $this->db = $mydb;
    }

    // Get methods
    public function getUsername ()
    {
        return $this->username;
    }

    public function getPassword ()
    {
        return $this->password;
    }

    public function getHost ()
    {
        return $this->host;
    }

    public function getDb ()
    {
        return $this->db;
    }

    // Set methods
    public function setUsername ($myusername)
    {
        $this->username = $myusername;
    }

    public function setPassword ($mypassword)
    {
        $this->password = $mypassword;
    }
}

```

```

        public function setHost ($myhost)
        {
            $this->host = $myhost;
        }
        public function setDb ($mydb)
        {
            $this->db = $mydb;
        }
    } // End DBparms class

// Class to construct Students with getters/setter
class StudentClass
{
    // property declaration
    private $firstname="";
    private $lastname="";
    private $email="";
    private $tychoname="";

    // Constructor
    public function __construct($firstname,$lastname,$email,$tychoname)
    {
        $this->firstname = $firstname;
        $this->lastname = $lastname;
        $this->email = $email;
        $this->tychoname = $tychoname;
    }

    // Get methods
    public function getFirstname ()
    {
        return $this->firstname;
    }
    public function getLastName ()
    {
        return $this->lastname;
    }
    public function getEmail ()
    {
        return $this->email;
    }
    public function getTychoname ()
    {
        return $this->tychoname;
    }

    // Set methods
    public function setFirstname ($value)
    {
        $this->firstname = $value;
    }
    public function setLastName ($value)
    {
        $this->lastname = $value;
    }
}

```

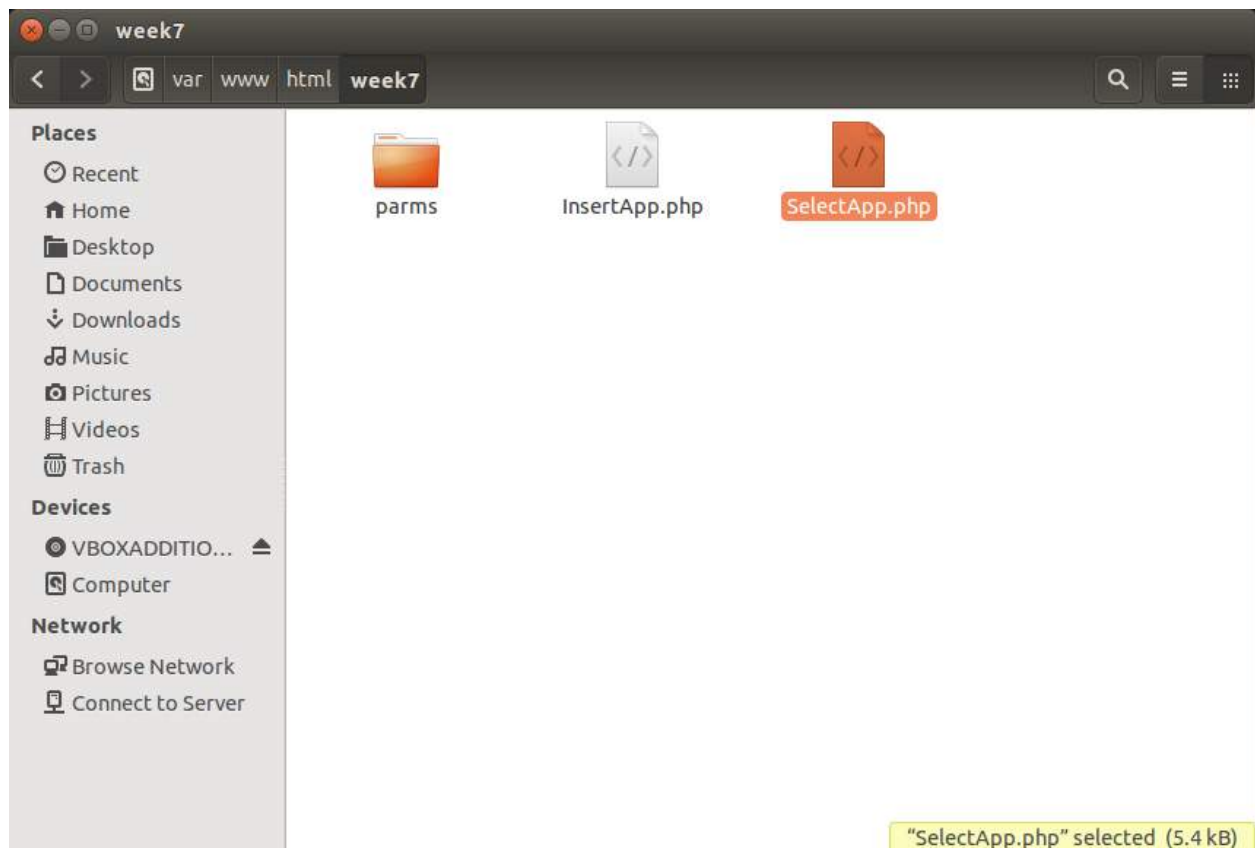
```

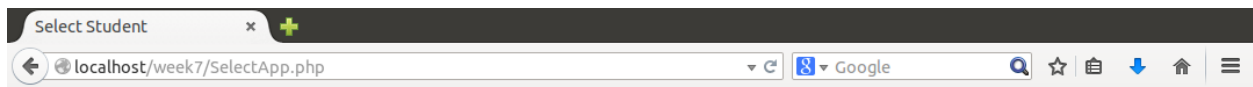
    public function setEmail ($value)
    {
        $this->email = $value;
    }
    public function setTychoname ($value)
    {
        $this->tychoname = $value;
    }
} // End Studentclass

?>
</body>
</html>

```

2. Place the SelectApp.php in the week7 folder on your VM and run launch from your local host browser. As you insert data from the previous InsertApp.php you will be able to watch the table grow in the number of records.





Select the Student to Update

Number of Students in Database is: 3

Jimmy	Johnson	jjohnson@umuc.edu	jjohnson1
Jimmy	Johnson	jjohnson2@umuc.edu	jjohnson2
Jimmy	Robertson	james.robertson@umuc.edu	jrobertson1

[Insert Another Students](#)

Part 3 Delete data from a MySQL database using PHP forms

Now that we have a form to Insert and Select data, we can continue to expand and add the delete functionality. This code shows you an approach to deleting data from a data table. Deleting data from a table can be a dangerous and often an unrecoverable event so make sure your application really requires this type of functionality.

1. Type, or copy and paste from the source code examples, the following code into your text editor and save as DeleteApp.php. You should review and tinker with all aspects of the code to become comfortable with the functionality. Notice the Delete functionality and associated queries.

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

  <?php

  // Check to see if Delete name is provided
```

```

if (isset($_GET["tychoname"])) {
    $toDelete = $_GET["tychoname"];
    // A bit dangerous without checks and use of getMethod
    deleteIt($toDelete);
    echo "Thanks for the deletion of $toDelete";
    echo "<p></p>";
    echo "<a href=InsertApp.php> Insert Another Students </a>";
    echo "<p></p>";
    echo "<a href=SelectApp.php> Select Students </a>";
    echo "<p></p>";
    echo "<a href=DeleteApp.php> Delete Students </a>";

}
else {
    show_form();

    // Provide option for inserting another student
    echo "<p></p>";
    echo "<a href=InsertApp.php> Insert Another Students </a>";
    echo "<p></p>";
    echo "<a href=SelectApp.php> Select Students </a>";
}

?>

<?php
function show_form() {

    echo "<p></p>";
    echo "<h2> Select the Student to Delete</h2>";
    echo "<p></p>";
    // Retrieve the students
    $students = selectStudents();

    echo "<h3> " . "Number of Students in Database is: " .
sizeof($students) . "</h3>";
    // Loop through table and display
    echo "<table border='1'>";
    foreach ($students as $data) {
        echo "<tr>";
        // Provide Hyperlink for Selection
        // Could also use Form with Post method
        echo "<td> <a href=DeleteApp.php?tychoname=" . $data->getTychoname() .
">" . "Delete" . "</a></td>";
        echo "<td>" . $data->getFirstname() . "</td>";
        echo "<td>" . $data->getLastname() . "</td>";
        echo "<td>" . $data->getEmail() . "</td>";
        echo "<td>" . $data->getTychoname() . "</td>";
        echo "</tr>";
    }
    echo "</table>";

} // End Show form
?>

<?php

```

```

function deleteIt($studentD) {
    echo "About to Delete " . $studentD ;
    // Connect to the database
    $mysqli = connectdb();

    // Add Prepared Statement
    $Query = "Delete from Students
              where tychoName = ?";

    $stmt = $mysqli->prepare($Query);

    // Bind and Execute
    $stmt->bind_param("s", $studentD);
    $stmt->execute();

    // Clean-up

    $stmt->close();

    $mysqli->close();
}
function selectStudents ()
{
    // Connect to the database
    $mysqli = connectdb();

    // Add Prepared Statement
    $Query = "Select firstName,lastName,eMail,tychoName from
Students";

    $result = $mysqli->query($Query);
    $myStudents = array();
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            // Assign values
            $firstname = $row["firstName"];
            $lastname = $row["lastName"];
            $email = $row["eMail"];
            $tychoname= $row["tychoName"];

            // Create a Student instance
            $studentData = new
Studentclass($firstname,$lastname,$email,$tychoname);
            $myStudents[] = $studentData;
        }
    }

    $mysqli->close();

    return $myStudents;

}

function getDbparms()

```

```

        {
            $trimmed = file('parms/dbparms.txt', FILE_IGNORE_NEW_LINES |
FILE_SKIP_EMPTY_LINES);
            $key = array();
            $vals = array();
            foreach($trimmed as $line)
            {
                $pairs = explode("=", $line);
                $key[] = $pairs[0];
                $vals[] = $pairs[1];
            }
            // Combine Key and values into an array
            $mypairs = array_combine($key, $vals);

            // Assign values to ParametersClass
            $myDbparms = new
DbparmsClass($mypairs['username'], $mypairs['password'],
                $mypairs['host'], $mypairs['db']);

            // Display the Paramters values
            return $myDbparms;
        }

function connectdb() {
    // Get the DBParameters
    $mydbparms = getDbparms();

    // Try to connect
    $mysqli = new mysqli($mydbparms->getHost(), $mydbparms->
getUsername(),
                                $mydbparms->getPassword(), $mydbparms->getDb());

    if ($mysqli->connect_error) {
        die('Connect Error (' . $mysqli->connect_errno . ') '
            . $mysqli->connect_error);
    }
    return $mysqli;
}

class DBparmsClass
{
    // property declaration
    private $username="";
    private $password="";
    private $host="";
    private $db="";

    // Constructor
    public function __construct($myusername, $mypassword, $myhost, $mydb)
    {
        $this->username = $myusername;
        $this->password = $mypassword;
        $this->host = $myhost;
        $this->db = $mydb;
    }

    // Get methods

```

```

        public function getUsername ()
        {
            return $this->username;
        }
        public function getPassword ()
        {
            return $this->password;
        }
        public function getHost ()
        {
            return $this->host;
        }
        public function getDb ()
        {
            return $this->db;
        }

        // Set methods
        public function setUsername ($myusername)
        {
            $this->username = $myusername;
        }
        public function setPassword ($mypassword)
        {
            $this->password = $mypassword;
        }
        public function setHost ($myhost)
        {
            $this->host = $myhost;
        }
        public function setDb ($mydb)
        {
            $this->db = $mydb;
        }

    } // End DBparms class

    // Class to construct Students with getters/setter
    class StudentClass
    {
        // property declaration
        private $firstname="";
        private $lastname="";
        private $email="";
        private $tychoname="";

        // Constructor
        public function __construct($firstname,$lastname,$email,$tychoname)
        {
            $this->firstname = $firstname;
            $this->lastname = $lastname;
            $this->email = $email;
            $this->tychoname = $tychoname;
        }

        // Get methods
        public function getFirstname ()

```



```

    {
        return $this->firstname;
    }
    public function getLastName ()
    {
        return $this->lastname;
    }
    public function getEmail ()
    {
        return $this->email;
    }
    public function getTychoname ()
    {
        return $this->tychoname;
    }

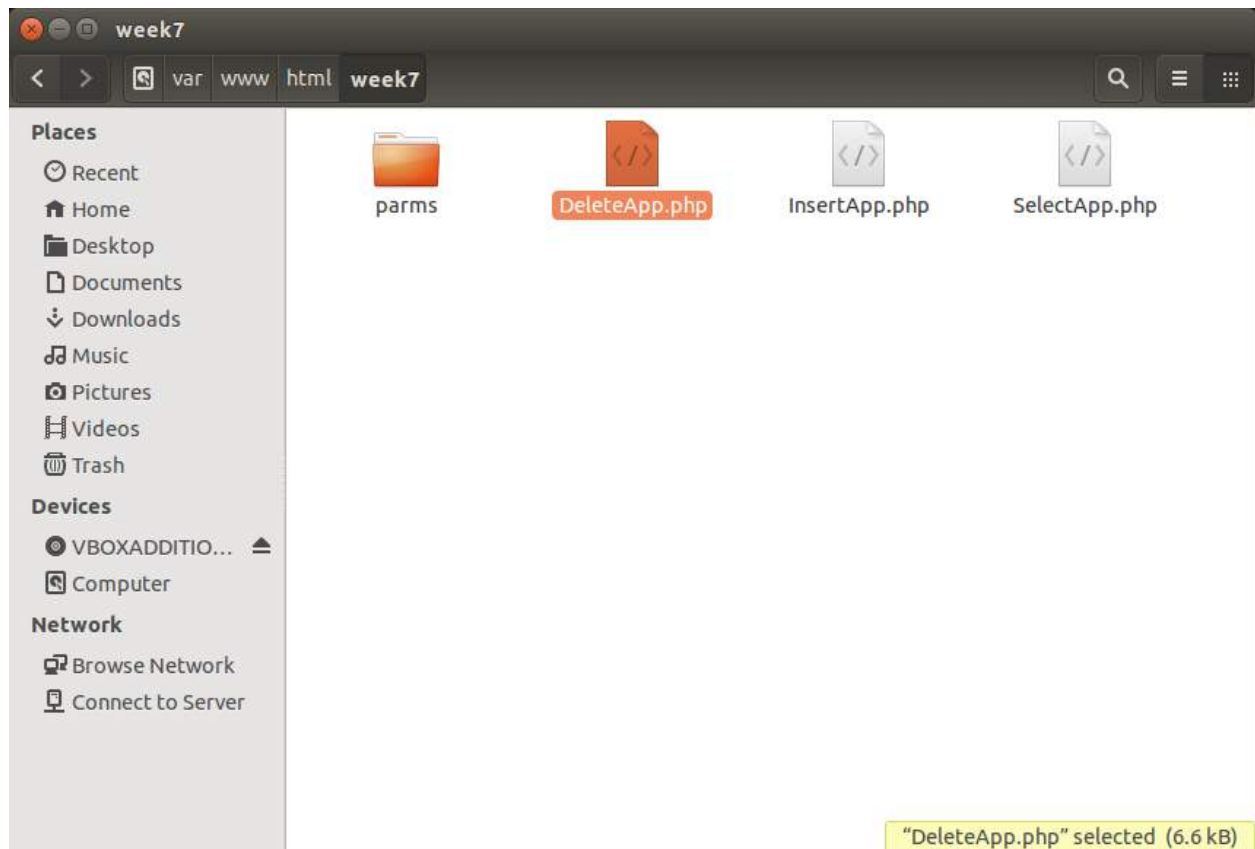
    // Set methods
    public function setFirstname ($value)
    {
        $this->firstname = $value;
    }
    public function setLastName ($value)
    {
        $this->lastname = $value;
    }
    public function setEmail ($value)
    {
        $this->email = $value;
    }
    public function setTychoname ($value)
    {
        $this->tychoname = $value;
    }

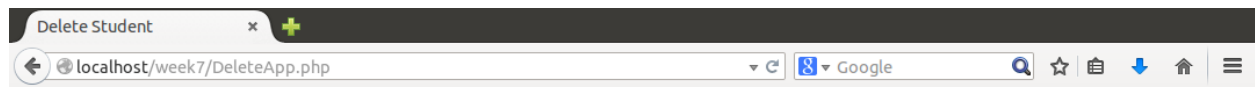
} // End Studentclass

?>
</body>
</html>

```

2. Add the file to your week7 folder on your VM and launch the URL.





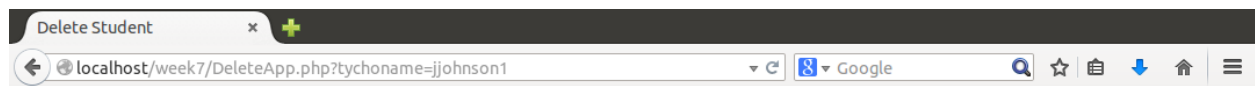
Select the Student to Delete

Number of Students in Database is: 3

Delete	Jimmy	Johnson	jjohnson@umuc.edu	jjohnson1
Delete	Jimmy	Johnson	jjohnson2@umuc.edu	jjohnson2
Delete	Jimmy	Robertson	james.robertson@umuc.edu	jrobertson1

[Insert Another Students](#)

[Select Students](#)

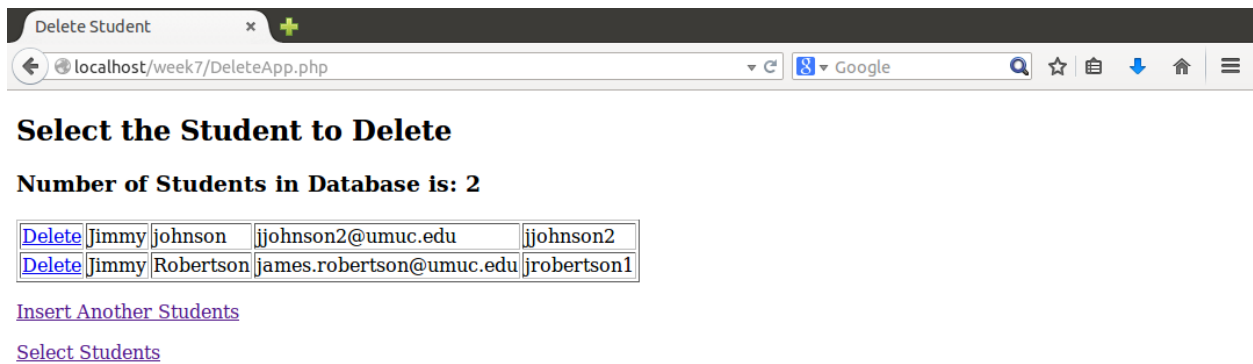


About to Delete jjohnson1 Thanks for the deletion of jjohnson1

[Insert Another Students](#)

[Select Students](#)

[Delete Students](#)



Part 4 - Update data in a MySQL database using PHP forms

Now that we have a form to Insert, delete and Select data, we can continue to expand and add the update functionality. This code shows you an approach to updating data.

1. Type, or copy and paste from the source code examples, the following code into your text editor and save as UpdaeApp.php. You should review and tinker with all aspects of the code to become comfortable with the functionality.

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

  <?php
  // Check to see if Delete name is provided
  if (isset($_GET["tychoname"])) {
    $toUpdate = $_GET["tychoname"];
    // A bit dangerous without checks and use of getMethod
    updateIt($toUpdate);

    echo "<p></p>";
```

```

        echo "<a href=InsertApp.php> Insert Another Students </a>";
        echo "<p></p>";
        echo "<a href=SelectApp.php> Select Students </a>";
        echo "<p></p>";
        echo "<a href=DeleteApp.php> Delete Students </a>";
        echo "<p></p>";
        echo "<a href=UpdateApp.php> UpdateStudents </a>";
    }
    else if (isset($_POST["UpdateMe"])) {
        // Assign values
        $firstname = $_POST["firstname"];
        $lastname = $_POST["lastname"];
        $tychoname = $_POST["tychoname"];
        $email = $_POST["email"];

        $student = new StudentClass($firstname,$lastname,$email,$tychoname);
        // Update the database
        FinalUpdate($student);
        echo "<p></p>";
        echo "<a href=InsertApp.php> Insert Another Students </a>";
        echo "<p></p>";
        echo "<a href=SelectApp.php> Select Students </a>";
        echo "<p></p>";
        echo "<a href=DeleteApp.php> Delete Students </a>";
        echo "<p></p>";
        echo "<a href=UpdateApp.php> UpdateStudents </a>";
    }
    else {
        show_form();

        // Provide option for inserting another student
        echo "<p></p>";
        echo "<a href=InsertApp.php> Insert Another Students </a>";
        echo "<p></p>";
        echo "<a href=SelectApp.php> Select Students </a>";
    }

    ?>

<?php
function show_form() {

    echo "<p></p>";
    echo "<h2> Select the Student to Delete</h2>";
    echo "<p></p>";
    // Retrieve the students
    $students = selectStudents();

    echo "<h3> " . "Number of Students in Database is: " .
sizeof($students) . "</h3>";
    // Loop through table and display
    echo "<table border='1'>";
    foreach ($students as $data) {
        echo "<tr>";
        // Provide Hyperlink for Selection
        // Could also use Form with Post method
        echo "<td> <a href=UpdateApp.php?tychoname=" . $data->getTychoname() .
">" . "Update" . "</a></td>";

```

```

        echo "<td>" . $data->getFirstname() . "</td>";
        echo "<td>" . $data->getLastName() . "</td>";
        echo "<td>" . $data->getEmail() . "</td>";
        echo "<td>" . $data->getTychoname() . "</td>";
        echo "</tr>";
    }

    echo "</table>";

} // End Show form
?>

<?php

function getStudent ($studentD) {
    // Connect to the database
    $mysqli = connectdb();

    // Add Prepared Statement
    $Query = "Select firstName, lastName, eMail, tychoName from
Students
                where tychoName = ?";

    $stmt = $mysqli->prepare($Query);

// Bind and Execute
$stmt->bind_param("s", $studentD);
$result = $stmt->execute();

$stmt->bind_result($firstName,$lastName,$eMail,$tychoName);

/* fetch values */
$stmt->fetch();
$studentData = new Studentclass($firstName,$lastName,$eMail,$tychoName);

// Clean-up
    $stmt->close();
    $mysqli->close();
    return $studentData;
}

function updateIt($studentD) {

    $student = getStudent($studentD);
    // Extract data
    $firstname = $student->getFirstname();
    $lastname = $student->getLastName();
    $email = $student->getEmail();
    $tychoname= $student->getTychoname();

    // Show the data in the Form for update
    ?>
    <p></p>

    <form name="updateStudent" method="POST" action="UpdateApp.php">
    <table border="1" width="75%" cellpadding="0">
        <tr>
            <td width="157">Firstname:</td>

```

```

        <td><input type="text" name="firstname" value='<?php
echo $firstname ?>' size="30"></td>
    </tr>
    <tr>
        <td width="157">Lastname:</td>
        <td><input type="text" name="lastname" value='<?php
echo $lastname ?>' size="30"></td>
    </tr>
    <tr>
        <td width="157">WebTycho username:</td>
        <td><input type="text" name="tychoname" value='<?php
echo $tychoname ?>' size="30"></td>
    </tr>
    <tr>
        <td width="157">Email:</td>
        <td><input type="text" name="email" value='<?php echo
$email ?>' size="30"></td>
    </tr>
    <tr>
        <td width="157"><input type="submit" value="Update"
name="UpdateMe"></td>
        <td>&nbsp;</td>
    </tr>
</table>
</form>

<?php
}
function selectStudents ()
{
    // Connect to the database
    $mysqli = connectdb();

    // Add Prepared Statement
    $Query = "Select firstName,lastName,eMail,tychoName from
Students";

    $result = $mysqli->query($Query);
    $myStudents = array();
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            // Assign values
            $firstname = $row["firstName"];
            $lastname = $row["lastName"];
            $email = $row["eMail"];
            $tychoname= $row["tychoName"];

            // Create a Student instance
            $studentData = new
Studentclass($firstname,$lastname,$email,$tychoname);
            $myStudents[] = $studentData;
        }
    }

    $mysqli->close();
}

```

```

        return $myStudents;
    }

    function getDbparms()
    {
        $trimmed = file('parms/dbparms.txt', FILE_IGNORE_NEW_LINES |
FILE_SKIP_EMPTY_LINES);
        $key = array();
        $vals = array();
        foreach($trimmed as $line)
        {
            $pairs = explode("=", $line);
            $key[] = $pairs[0];
            $vals[] = $pairs[1];
        }
        // Combine Key and values into an array
        $mypairs = array_combine($key, $vals);

        // Assign values to ParametersClass
        $myDbparms = new
DbparmsClass($mypairs['username'], $mypairs['password'],
                $mypairs['host'], $mypairs['db']);

        // Display the Parameters values
        return $myDbparms;
    }

    function connectdb() {
        // Get the DBParameters
        $mydbparms = getDbparms();

        // Try to connect
        $mysqli = new mysqli($mydbparms->getHost(), $mydbparms->
getUsername(),
                                $mydbparms->getPassword(), $mydbparms->getDb());

        if ($mysqli->connect_error) {
            die('Connect Error (' . $mysqli->connect_errno . ') '
                . $mysqli->connect_error);
        }
        return $mysqli;
    }
}

class DBparmsClass
{
    // property declaration
    private $username="";
    private $password="";
    private $host="";
    private $db="";

    // Constructor
    public function __construct($myusername, $mypassword, $myhost, $mydb)
    {
        $this->username = $myusername;
    }
}

```



```

        $this->password = $mypassword;
        $this->host = $myhost;
        $this->db = $mydb;
    }

    // Get methods
    public function getUsername ()
    {
        return $this->username;
    }

    public function getPassword ()
    {
        return $this->password;
    }

    public function getHost ()
    {
        return $this->host;
    }

    public function getDb ()
    {
        return $this->db;
    }

    // Set methods
    public function setUsername ($myusername)
    {
        $this->username = $myusername;
    }

    public function setPassword ($mypassword)
    {
        $this->password = $mypassword;
    }

    public function setHost ($myhost)
    {
        $this->host = $myhost;
    }

    public function setDb ($mydb)
    {
        $this->db = $mydb;
    }

} // End DBparms class

// Class to construct Students with getters/setter
class StudentClass
{
    // property declaration
    private $firstname="";
    private $lastname="";
    private $email="";
    private $tychoname="";

    // Constructor
    public function __construct($firstname,$lastname,$email,$tychoname)
    {
        $this->firstname = $firstname;
        $this->lastname = $lastname;
    }
}

```

```

        $this->email = $email;
        $this->tychoname = $tychoname;
    }

    // Get methods
    public function getFirstname ()
    {
        return $this->firstname;
    }
    public function getLastname ()
    {
        return $this->lastname;
    }
    public function getEmail ()
    {
        return $this->email;
    }
    public function getTychoname ()
    {
        return $this->tychoname;
    }

    // Set methods
    public function setFirstname ($value)
    {
        $this->firstname = $value;
    }
    public function setLastname ($value)
    {
        $this->lastname = $value;
    }
    public function setEmail ($value)
    {
        $this->email = $value;
    }
    public function setTychoname ($value)
    {
        $this->tychoname = $value;
    }

} // End Studentclass

// Final Update
function FinalUpdate($student) {
    // Assign values
    $firstname = $student->getFirstname();
    $lastname = $student->getLastname();
    $tychoname = $student->getTychoname();
    $email = $student->getEmail();

    // update
    // Connect to the database
    $mysqli = connectdb();

    // Add Prepared Statement
    $Query = "Update Students set FirstName = ?,

```

```

        lastName = ?, email = ?, tychoName = ?
        where tychoName = ?";

        $stmt = $mysqli->prepare($Query);

$stmt->bind_param("sssss", $firstname, $lastname,
$email, $tychoName, $tychoName);
$stmt->execute();

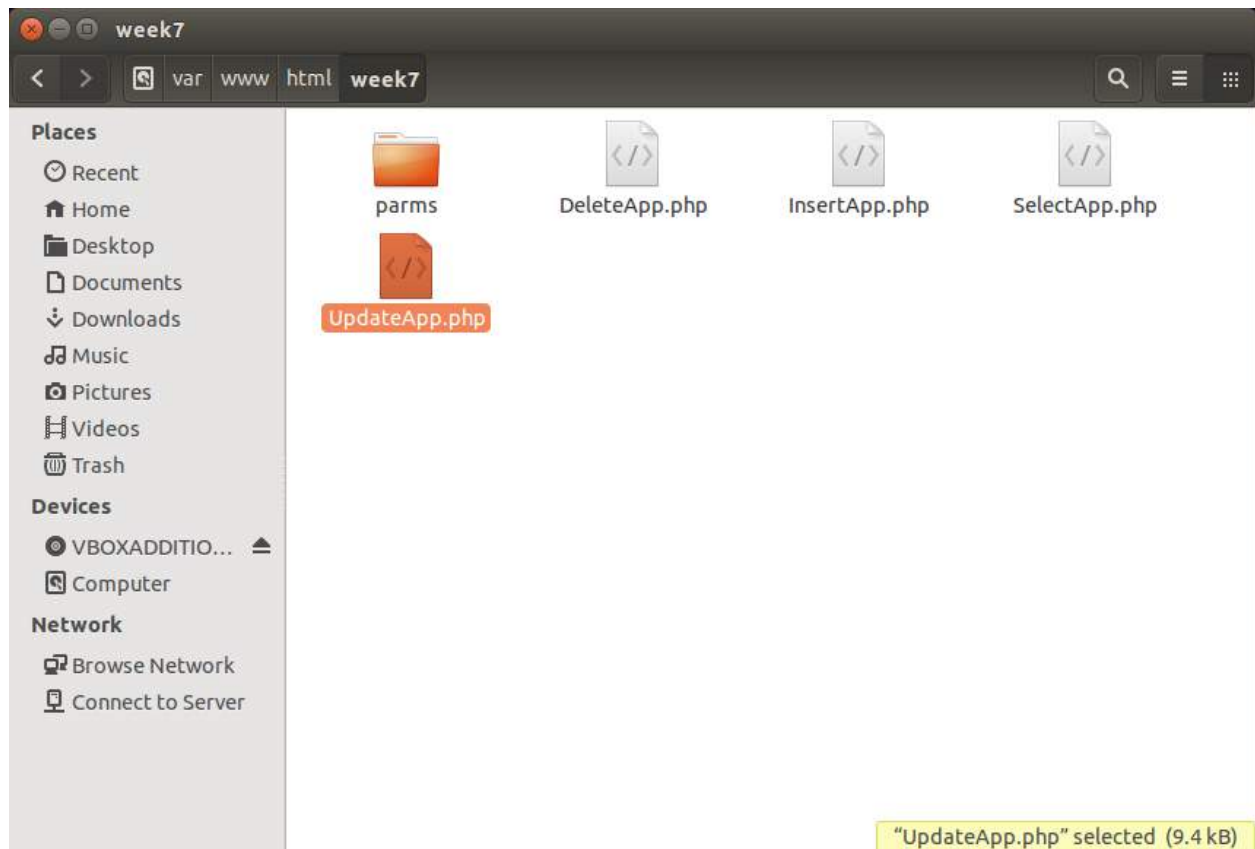
//Clean-up
        $stmt->close();
        $mysqli->close();
}

?>
</body>
</html>

```

2. As shown above, the update usually takes more steps and code. Notice you have to select the record to be updated, then prepare a form to display the current fields and then finally update those fields and save them to the database.

3. After moving the UpdateApps.php file to your VM in the week7 folder and launch with your Web browser.



Select the Student to Delete

Number of Students in Database is: 2

Update	Jimmy	Johnson	jjohnson2@umuc.edu	jjohnson2
Update	Jimmy	Robertson	james.robertson@umuc.edu	jrobertson1

[Insert Another Students](#)

[Select Students](#)

Firstname:	<input type="text" value="James"/>
Lastname:	<input type="text" value="Robertson4"/>
WebTycho username:	<input type="text" value="jrobertson1"/>
Email:	<input type="text" value="james.robertson@umuc.edu"/>
<input type="button" value="Update"/>	

[Insert Another Students](#)

[Select Students](#)

[Delete Students](#)

[UpdateStudents](#)

Update Student x

localhost/week7/UpdateApp.php?tychoname=jrobertson1 Google

Firstname:	James
Lastname:	Robertson4
WebTycho username:	jrobertson1
Email:	james.robertson@umuc.edu
Update	

[Insert Another Students](#)

[Select Students](#)

[Delete Students](#)

[UpdateStudents](#)

Lab submission details:

As part of the submission for this Lab, you will modify your session-based e-Commerce application from week 4 to add database support for your products and sales. Specifically, all product data must be organized in one or more MySQL tables. In addition, you will need to store your customer data as they order from your store. You should store all customer information including, username, password, firstname, lastname, street address, city, state, zip code, phone number, credit card number, credit card type, and expiration date and products purchases. Other fields can be added for your unique design.

Your product data should be dynamic allowing the ability for the store owner to insert new products, update existing products, delete existing products and list all available products. When designing your application, be sure to use prepared statements to minimize SQL injection. Also, make sure your Forms flow logically within your application and are presented in an attractive easy-to-use Web interface.

Create screen shots showing the successful running 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 SQL script file.

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