Lab #2 – PHP and MySQL
Part 2

Due Date:	Friday, October 25, 2019 [11:59 PM]
Purpose:	This lab gives you hands-on experience with writing PHP web apps that connect to the MySQL DBMS.
Turn-ins:	Two php files ("form.php" and "process.php").
Submission:	Upload your php files to Sakai.
Resources:	Lab computer (with XAMPP, Apache and MySQL installed), Lecture notes/handouts, Text Book

---

Initial Setup:

1. Starting the Apache and MySQL services in Windows using XAMPP.
    a. In the search box of the Start Menu, type "XAMPP" and select "XAMPP Control Panel (Beta)" in the search list.
    b. In the XAMPP control panel, click "START" next to MySQL and APACHE.
    c. Ensure the services start properly and that they are given PIDs and PORTs.

2. Importing SQL source file to create and populate the "COMPANY" database.
    a. Download the SQL Source file from Sakai to `c:\xampp\mysql\bin`
    b. In the search box of the Start Menu, type "cmd" and select the cmd application that appears in the search list.
    c. Navigate the directory that contains the MySQL application with: `cd c:\xampp\mysql\bin`
    d. Start the MySQL command-line application with the following command: `mysql –h localhost –u root`
    e. Import the SQL Source file with the following command `source company.sql;`
    f. You should see the database being created and populated.
        i. For your reference, a PDF of the table descriptions and table data is also available on Sakai.
    g. Close MySQL command line interface. `\q`

    h. Rename your PHP file from Part #1 of the lab to **"process.php".**

3. Copy the PHP template (form.php) from Sakai for Part #2 of the lab to the same directory.

4. Open both "form.php" and "process.php" files in Notepad++.

5. Examine the "form.php" file and identify the two sections of the file.
    a. You will also notice that most of the HTML for the output web page is created for you. There are also some CSS style rules created to modify the presentation.

form.php - Section I - Creating a simple Web Form

6. Under Section #1 on "form.php", you will see that an HTML form has already been started for you. The action of the form is your "process.php" file.

7. You need to create a single text box with the name "deptnum" and the default value of "6". You will also need to include a Submit and Reset button. You do not need to use PHP for this section (refer to the FORM examples from class).

process.php – Modify your existing database query:

8. In the first part of the lab, you created several queries in your PHP file. We will only need to use the last portion of your code where you queried the database and displayed the results in a table (including employee SSNs). You may remove any code that you no longer need.

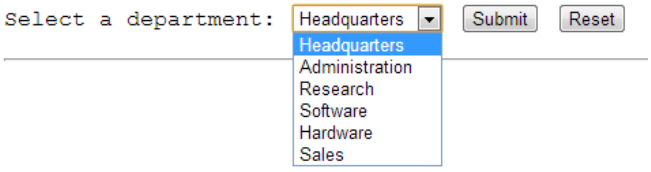
9. You will modify your process.php file, by adding code that allows you to access the data supplied by the user in the form. You will use the number entered by the user as the department number in your SQL query. (Previously, we hard-coded the department number to be 6.)

    a. You will need to use `$_POST['deptnum']` to access the form data.
    b. Remember that you should to check to see if FORM data is valid before you start to use it in your queries. (Utilize the PHP examples from previous lectures for reference.)

10. Next, test your form.php and process.php files in the browser. Try entering "6" in the text box and your resulting data table should be identical to the output that you generated in the previous part of the lab.

form.php - Section II – Creating a Web Form with a dynamic Select Menu:

11. In the second section on the form.php, you will create a Web Form that contains a dynamic select menu. The menu will display all of the departments listed in the database. You will need to query the database in order to create the select menu.



12. The shell of the form is already written. You will need to complete the code inside of the <?php … ?> section.

13. Since you are querying the database and displaying results, you will need to follow all of the steps that you used previously for your queries.

14. When you are displaying the results, you will be using them to create a select menu. For your reference, this is how a select menu looks like normal HTML:

```
<select name="selectmenu">
     <option value="selectchoice1"> Choice 1 </option>
     <option value="selectchoice2"> Choice 2 </option>
     <option value="selectchoice3"> Choice 3 </option>
 </select>
```

15. The while loop of your PHP code that displays query results will create the <option> tag for each department. The "value" attribute should be set to the department number and the department name should be displayed between the starting and ending tags.
    a. Example: `<option value="6"> Research </option>`
    b. HINT: You will need to output quotation marks ", which means that you need to use single quotes ' in your echo statements. Like this: `echo ' <option value=" ';` `echo ' "> ';`

16. Try viewing your form.php in the browser. You should see a select menu that lists all of the departments in the database. If that code is working correctly, you should then be able to select a department and click SUBMIT. The results of your existing query in the process.php should automatically display.

If you have extra time and/or want to make a more robust Web Application:

- Allow the user to also search for employees in the company. The user can search for an employee by their First Name. (For simplicity: We will assume that the first name entered by the user has to match the first name in the database exactly.) The output of this new search feature should be the first and last name of all employees that match the user input.

- In order to support both the department project search feature and this new employee name search feature, we will have to add a radio button that allows the user to select one of the two searches. (We will assume that only one of the two searches can be completed at one time.)

- You will have to modify both the "form.php" and "process.php".