In Class Lab #2 – PHP and MySQL
Part 1

Purpose:	This lab gives you hands-on experience with writing PHP web apps that connect to the MySQL DBMS.
Turn-ins:	A single php file containing all of the required elements listed in the lab directions.
Submission:	Upload your php file 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 (company.sql).
   a. Download the SQL Source file from Sakai->Resources->Lab2-Part1 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 to 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`

3. Download the PHP template for the lab (lab2_part1.php) from Sakai and save it to: `G:\424_SITE\`

4. Open the lab2_part1.php file in Notepad++.

5. Examine the PHP file and identify the Sections of this part of the lab.
   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.
   b. You will be completing the missing code from the `<?php … ?>` sections. (Notice that comments have been inserted to help guide you through your code development.)

6. For the remainder of the lab, please refer to the "database.php" example distributed in the previous class. This example should provide you with all of the necessary commands for this lab. They just need to be modified slightly to fit the lab's requirements.

Section I - Creating a PHP Web App that connects to MySQL:

7. In the first section of the lab, you will need to establish a connection to the MySQL server (localhost, root) and the "COMPANY" database.
8. Please be sure to include `IF` statements with `die()` function calls to catch any connection errors/problems.
9. Run the "COPY WEBFILES TO SERVER" script to copy your php file to the Apache webserver on your machine.
10. Try opening your PHP file in the browser. Open a browser of your choice and in the address box, enter the following URL:

```
http://localhost/lab2_part1.php
```

11. If you see any errors in your PHP syntax or any errors connecting to the database, you will need to debug to determine the cause of the problem. (Please let me know and I will try to help.)

Section II – Running an SQL Query:

12. In the second section of the lab, you will create and submit an SQL query to the database.
13. The query should select all projects in the PROJECT table whose associated department number is "6".
    a. I would recommend displaying your query on the webpage, so that you can be sure that it is entered correctly. Use a command similar to this: `echo "<p> $query</p>";`
14. You will then need to submit the query to the database using the `query()` function.
15. Remember to check to make sure that the query executed properly and results were obtained.
16. You will now need to output the required data from your results in an HTML table. A sample table is provided in the HTML code before the php code segment.
    a. You should have a header row that identifies each column.
    b. Each row of the table should display a Project Name, a Project Number, a Project Location.
17. Save your file and re-run the server script. Refresh your PHP in the browser window to determine if your code executes properly. (If you have errors or are stuck, please let me know.)

Section III – Running an advanced SQL Query with a sub-query:

18. In the third section, you will duplicate your code from the second section and modify it slightly. This modification includes running an additional SQL query for each row in the result of the main query.
19. The sub-query will look up employees in the WORKS_ON table that are assigned to each project selected by the main query.
20. Copy and paste the entire code that you created for Section II into the Section III php code segment.
21. You will now need to modify the portion of your code that creates the HTML table output.
    a. You will be adding an additional column, as shown in the example.
22. In order to obtain the employee SSNs, you will need to create a sub-query that runs in your while loop.
    a. Utilize the same procedure as you did for the main query. You MUST have different variable names for the sub-query. (Otherwise you will be overwriting the list of projects that you already retrieved.) For simplicity, just preface each variable with "sub". For example: `$subquery, $subrow, $subresult`
23. Remember to properly indent your code, so that it is clear to see your sub-query code. (Indentation is your friend!)
24. Refresh your PHP in the browser window to determine if your code executes properly. (If you have errors or are stuck, please let me know.)