Lab 6 Assignment Instructions with a very detailed outline if you need help

In order to get credit for Lab 6, you must submit not only a screen shot of your directory structure, but also zip the folder with your new “contribution” application.

**Contribution Application:**

Goal: Using the Company database, calculate Employee contributions for a company party, and display these contribution on a web page. Your output should look like this:



Create two folders: Includes and Contribution.

* The Includes folder was already created in class. If you were following along with the lab, you placed it in the Document Root folder, which is htdocs.
	+ 1. The Includes folder will contain two files: **db.inc.php** and **error.inc.php**, both of which we wrote and saved in class.

**Db.inc.php** connects to the database. If there is a problem connecting, **error.inc.php** will display an error message and kill the application. If you didn’t do it during the lab, then do it now. These are also included in the files that I posted in the skeleton. If you are using my copies, make sure that you update the user name and password that are currently in db.inc.php. I just used my initials and last name as a generic login and pwd.

* + 1. Either move the Includes folder to your ROOT, or change the directory specified in your contribution application so that what the application is “including” matches where the includes folder is located.
* The Contribution folder should be stored wherever you store your Lab applications, and this is the folder that you should zip and submit as your assignment. The Contribution folder should contain two files:
	1. **Index.php**: This document “includes” db.inc.php (in order to connect to the database), and runs the SQL query to extract the data from the database, assign the database results to PHP variables, calculate the contribution and display the results in a table in html. Use the following steps:
		+ 1. “INCLUDE” db.inc.php. Be sure to include where it is stored. For instance, in class we placed it in the document root directory, so we used:

include\_ once $\_SERVER[‘DOCUMENT-ROOT’].’includes/db.inc.php’;

* + - 1. “INCLUDE” contribution\_function.inc.php. (see step B below to see what that will contain). This is in the same folder as your application, so you don’t have to specify a path. If you choose to place all of your functions in a different folder, then you would have to specify the path.
			2. Use a Try…Catch block (enclosed in PHP tags), as demonstrated in previous labs, to run the query, which is to select everything from the employee table.
				1. The “Try” should assign to a PHP variable $employees” (or any name you choose) the pdo->query with your Select \* FROM Employees query.
				2. The “Catch” should assign an error message to $error and Include error.html.php, just as we have done in previous labs. It should be followed by an “exit()”, (still part of “catch”), just as in previous labs.
			3. In regular html not enclosed in PHP tags, Create a table, entitled “Employee Contributions” or using some kind of header such as “Here are all the employee contributions for the party:” You can be as plain or fancy with the table formatting as you want. Your table title doesn’t have to be part of the table if you are not familiar with how to do that.
			4. Using PHP, use a foreach loop to extract each row of your query results (which you stored in $employees variable when you ran the query in the “try” block, above). Inside your foreach loop, you will access each record (I called each record ‘$employee’, so my foreach looks like: foreach ($employees as $employee).
			5. assign the ‘salary’ attribute from the db record to the $salary php variable.
			6. Assign the ‘bdate’ attribute from the db record to the $bdate php variable.
			7. If you want, you can do the same for lname and fname. Or, you can simply reference these later in the table cells. You have to extract the salary and bdate, because they will require further processing. Lname and fname do not require any further processing—you will just display them in the table.
			8. At this point, you have to send $salary and $bdate to your contribution function, which is the function that you will define in your contribution\_function.inc.php file (see below). Assign the result that is returned from this function to a PHP variable with a name like $empCont or $contribution or something that makes sense.
			9. Back in html, (but still inside of your foreach loop) create one table row (that’s <tr>) for each employee. Create three table domains (which are columns, which are constructed using <td> ….. </td>). The first td should have the employees lname, which is $employee[‘lname’]. The second td should have employees first name. The third td should have the employees contribution. You may want to use number\_format or printf or any type of formatting that you find online to format the contribution as currency. Source of confusion: <td> and </td> are html. But between those html tags, you need to “echo” using php. Or, you can echo the whole thing, but I think that is more confusing.
			10. Close your row with a </tr> tag. Close your foreach loop. Close your </table>
	1. **Contribution\_function.inc.php.** This file will have a function that calculates the employee contribution as:

contribution = 2 \* $salary/1000 + $month +3

 The variable $bdate was sent to the function as a parameter. To get the month, you have to extract it (try using the substr function) from $bdate.

Recall that $salary is a PHP variable that is assigned the value of the MySQL attribute ‘salary’ that you have extracted from the Employee table of the Company database. Recall that you ran that query in the “try” block.

This function should return the contribution to the calling program.

You should then be able to display the information by accessing your contribution/index.php file through localhost

To help you out, I have posted some slides that go through some number formatting (and also date and time formats) in PHP.