Lab 9 – Create and Access MySQL database using PDO (PHP Data Object)

# Purpose

* Explore MySQL using PDO (PHP Data Object)

# Due Date

* This lab must be handed in:

**Sunday Mar 30, 2025 – before midnight**

# Assessment

* This Lab is worth 3% of your total course mark.

# Assigned Readings

* **Lecture Slides** posted on Brightspace
* Module 4 -> Part 2
* The following chapters of **Fundamentals of Web Development** will be useful in completing this exercise:
* Chapter 14

# Lab Supplies

To complete this lab you will require the following lab supplies:

* Lecture Slides (**Module 4 -> Part 2**) and Code Examples (**PHP, MySQL CRUD Example-EasyPHP.zip**) posted on Brightspace
* Textbook: **Fundamentals of Web Development** by Randy Connolly and Ricardo Hoar
* EasyPHP, or other WAMP server
* Eclipse, Notepad++ (or other text editor, or IDE)

# Summary of Tasks

1. Gather MySQL Information
2. Understand your database
3. Develop the logic to display your web application
4. View your webpage using a web browser
5. **Submit Source Code of all PHP files on Brightspace**

# Task 1

Before we can get started using MySQL on the web hosting server we need a few pieces of information.

* Host
  + The host variable should contain the value “localhost”.
  + Example: $host = “localhost”;
  + NOTE: Localhost is a networking term meaning ‘this computer’
* Username
  + The Username to access YOUR database (DB) MUST be the following:

$username = “cst8238”;

* Password
  + The password MUST be the following:

$password = "cst@8238";

* Database Name
  + The name of your database (DB) MUST be the following:

$database = "cst8238\_25w";

# Task 2

To create your database, username, password and tables on the EasyPHP Web server, please review the following document on Brightspace:

Course Content -> Module 4 – Dynamic Web Programming -> Part 2- PHP and MySQL -> CRUD Example-EasyPHP.zip -> Instruction\_MySQL\_EasyPHP.docx

# Task 3

Now that you have a database you need to understand the tables inside it. Your database (DB) must contain a table named *Employee*.

The ‘*Employee*’ table should have the following fields/columns:

* EmployeeId, INT, Primary Key, NOT NULL, AUTO INCREMENT
* FirstName, VARCHAR(50), NULL
* LastName, VARCHAR(50), NULL
* EmailAddress, VARCHAR(255), NULL
* TelephoneNumber, VARCHAR(20), NULL
* SocialInsuranceNumber, VARCHAR(11), NULL
* Password, VARCHAR(50), NULL
* Designation, VARCHAR(50), NULL
* AdminCode, INT, NULL
* **NOTE**
* ***EmployeeId*** is listed as ‘AUTO INCREMENT’ meaning that the database will automatically populate this field.
* **You MUST access the DB using PDO (PHP Data Object) as below**:

*$pdo =* ***new*** *PDO("mysql:host=$host;dbname=$database", $username, $password);*

*$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);*

Implement the following Design Pattern to create a ‘Common Look and Feel’ to be used on every page of your website.

|  |
| --- |
|  |

Your web site will include the following PHP scripts:

* Header.php
* Footer.php
* Menu.php
* CreateAccount.php
* Login.php
* ViewAllEmployees.php
* Admin.php
* SelectAccount.php
* UpdateAccount.php
* UpdateComplete.php

**NOTE:** CreateAccount.php, Login.php, ViewAllEmployees.php, Admin.php, SelectAccount.php, UpdateAccount.php, and UpdateComplete.php must include the Common look and feel implemented in Header.php, Footer.php and Menu.php.

**Header.php**

Header.php must contain a script to display a Common Header that will appear on every page. The header must display Program Name and Course Name

**Footer.php**

Footer.php must contain a script to display a Common Footer that will appear on every page. The footer must contain Student Number, First Name, Last Name, and Email Address

**Menu.php**

Menu.php must contain a script to display a Common Menu to be shown on every page. The menu must contain links to *CreateAccount.php*, *Login.php*, *ViewAllEmployees.php*, and *SelectAccount.php*

NOTE: *Admin.php*, *UpdateAccount.php*, and *UpdateComplete.php* will be accessed by clicking on the link for *SelectAccount.php****.***

**CreateAccount.php**

Create a form that will create employee accounts in the ‘Employee’ table of your database.

Details:

1. Use ‘input’ tags to accept the information listed in Task 3. Your form must populate all the columns of the ‘Employee’ table to create an employee account.

1. Using the above form, you must create three types of employee accounts (i) IT Specialist, (ii) IT Developer and (iii) Manager. While creating an ‘IT Specialist’ account, Admin Code may be set as ‘222’ (Designation = ‘IT Specialist’, Admin Code = 111). While creating an ‘IT Developer’ account, Admin Code may be set as ‘111’ (Designation = ’ITDeveloper’, Admin Code = 111). Again, while creating a ‘Manager’ account, it is MANDATORY to set the Admin Code as ‘999’ (Designation = ’Manager’, Admin Code = 999). You need to create two **IT Specialist** accounts, two **IT Developer** accounts, and one **Manager** account.
2. After creating every account, the user needs to be redirected to the ‘**ViewAllEmployees.php**’ page.

Sample Screenshot for ***CreateAccount.php*** is as follows:

|  |
| --- |
|  |

**Login.php**

The form ***Login.php*** allows the user to log into your application.

Details:

1. Create a form to accept the employee’s EmailAddress and Password as credentials to your site. Use an SQL Query to determine if the person has an account.
2. If the user has an account, redirect the user to ‘ViewAllEmployees.php’. Display an error if the user cannot log into the system.

Sample Screenshot for ***Login.php*** is as follows:

|  |
| --- |
|  |

**ViewAllEmployees.php**

This page pulls information from the Database and displays the information to the user.

Details:

1. If the employee tries to navigate to this page without having logged in – the employee should be redirected to the login page.
2. If the employee has successfully logged into the application and is directed to the page, then display the following information
   1. HTML table containing all rows of the ‘Employee’ table of your database**.** The HTML table must also contain a header row that identifies the column name of the database table.

Sample Screenshot for ***ViewAllEmployees.php*** is as follows:

|  |
| --- |
|  |

**Note**: You do NOT need to display the values of ‘Password’ and ‘AdminCode’ fields of the ‘Employee’ table**.**

**SelectAccount.php**

The form **SelectAccount.php** allows the Manager to select an employee whose information (that is previously stored into Employee table) needs to be updated.

Details:

1. If anyone tries to navigate to this page, the person should be redirected to the **Admin.php** page. Create a form to accept the employee’s EmailAddress, Password and Admin Code as credentials to your site. Only the Manager (Admin Code =999) will be able to access this page. Use an SQL Query to determine if the person has an account and he is a Manager.

Sample Screenshot for ***Admin.php*** is as follows:

|  |
| --- |
|  |

1. If the Manager has successfully logged into the application, he/she will be directed to the ***SelectAccount.php*** page that will display the following form to select an employee whose information needs to be updated:

|  |
| --- |
|  |

1. After clicking on the ‘Edit Employee’ button, the Manager should be redirected to the form ***UpdateAccount.php*** which allows the Manager to update the information of the corresponding employee.

Sample Screenshot for ***UpdateAccount.php*** is as follows:

|  |
| --- |
|  |

1. After clicking on the ‘Update Record’ button, the modified information of the corresponding employee should be updated into the Employee table of the database and a success/error message will be displayed in ***UpdateComplete.php***.

Sample Screenshot for ***UpdateComplete.php*** is as follows:

|  |
| --- |
|  |

**NOTE**: All the sample screenshots must include common header, footer and menu files.

# Task 4

Create Lab 9 submission folder ‘**Lab9**’ and copy ***CreateAccount.php***, ***Login.php***, ***ViewAllEmployees.php***, ***SelectAccount.php, Admin.php***, ***UpdateAccount.php***, ***UpdateComplete.php, Header.php, Footer.php, Menu.php*** and any other required files(e.g**. css file**) into this folder**.**

Create **Lab9.zip** file by compressing the '**Lab9'** folder.

To hand in your lab go to Brightspace and navigate to Course Content 🡪 Labs.

Then click on ‘Lab 9 – PHP and MySQL’ link.

Upload **Lab9.zip** on Brightspace.

**IMPORTANT NOTE**:

**You MUST demo the lab and explain the code to the Lab Professor to get marks.**