UNIVERSITY OF THE PHILIPPINES VISAYAS
COLLEGE OF ARTS AND SCIENCES
DIVISION OF PHYSICAL SCIENCES AND MATHEMATICS

CMSC 127 Database Systems
2nd Semester AY 2021-2022

Prepared by:
Christi Florence C. Cala-or
Asst.Prof., Faculty of Computer Science

ACTIVITY GUIDE 2
(UNIT 3)

ACADEMIC INTEGRITY

As a student of the University of the Philippines, I pledge to act ethically and uphold the value of honor and excellence. I understand that suspected misconduct on given assignments/examinations will be reported to the appropriate office and if established, will result in disciplinary action in accordance with University rules, policies and procedures. I may work with others only to the extent allowed by the Instructor.

Lab 3.2: PHP and DATABASE (Accessing Database in PHP)

PHP: Hypertext Preprocessor

PHP is a widely used, open-source scripting language and is used in server-side language.

Why PHP?

- It is powerful enough to be at the core of the biggest blogging system on the web (WordPress).
- It is deep enough to run the largest social network (Facebook)!
- It is also easy enough to be a beginner's first server- side language!
- It runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- It supports a wide range of databases

What is a PHP File?

- A PHP file can contain text, HTML, CSS, JavaScript, and PHP code
- The code is executed on the server, and the result is returned to the browser as plain HTML
- It has an extension "**.php**"

What can we do with PHP?

- Generate dynamic page content
- Create, open, read, write, delete, and close files on the server
- Collect form data
- Send and receive cookies
- **Add, delete, modify data in your database**
- Can be used to control user-access
- Can empty data

PHP Variables (can store data of different types)**:**

- String
- Integer
- Float (floating point numbers – also called **double**)
- Boolean
- Array
- Object
- NULL
- Resource

Use the "**.**" Operator to concatenate strings

PHP Statements

- Each statement should end with a semicolon (**;**)
- Syntax for if-else, switch, while, for loop and do-wile is the same with C and Java language

PHP is like C

PHP Functions

Syntax:

```
function functionName(){
    code_to_be_executed;
}
```

Note: A **function name** can start with a letter or underscore (not a number)

Use the **return** statement for a function to return a value when called

PHP Arrays:

Use the function ***array*(elements)** to create arrays. The function ***count*($arrayName)** returns the number of elements of an array.

Example:

```php
<?php
    $cars = array("Volvo", "BMW", "Toyota");
    echo count($cars);
?>
```

echo is println() ??

Types of arrays:

- **Indexed arrays** – arrays with a numeric index

  Example:

```php
<?php
    $cars = array("Volvo", "BMW", "Toyota");
    echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

- **Associative arrays** – arrays with named keys

  Example:

```php
<?php
    $age = array("Peter"=> "35", "Ben"=>"37", "Joe"=>"43");
    echo "Peter is " . $age['Peter'] . " years old.";
?>
```

- **Multidimensional arrays** – arrays containing one or more arrays

  Example:

```php
$cars = array
    (
        array("Volvo",22,18),
        array("BMW",15,13),
        array("Saab",5,2),
        array("Land Rover",17,15)
    );

<?php
    echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2].".<br>";
    echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2].".<br>";
    echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2].".<br>";
    echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2].".<br>";
?>
```

PHP and MySQL

PHP5 and later can work with a MySQL database using:

- MySQLi extension (the ***i*** stands for improved)
- PDO (PHP Data Objects)

Earlier versions of PHP used the MySQL extension. However, this extension was deprecated in 2012.

Instructions:

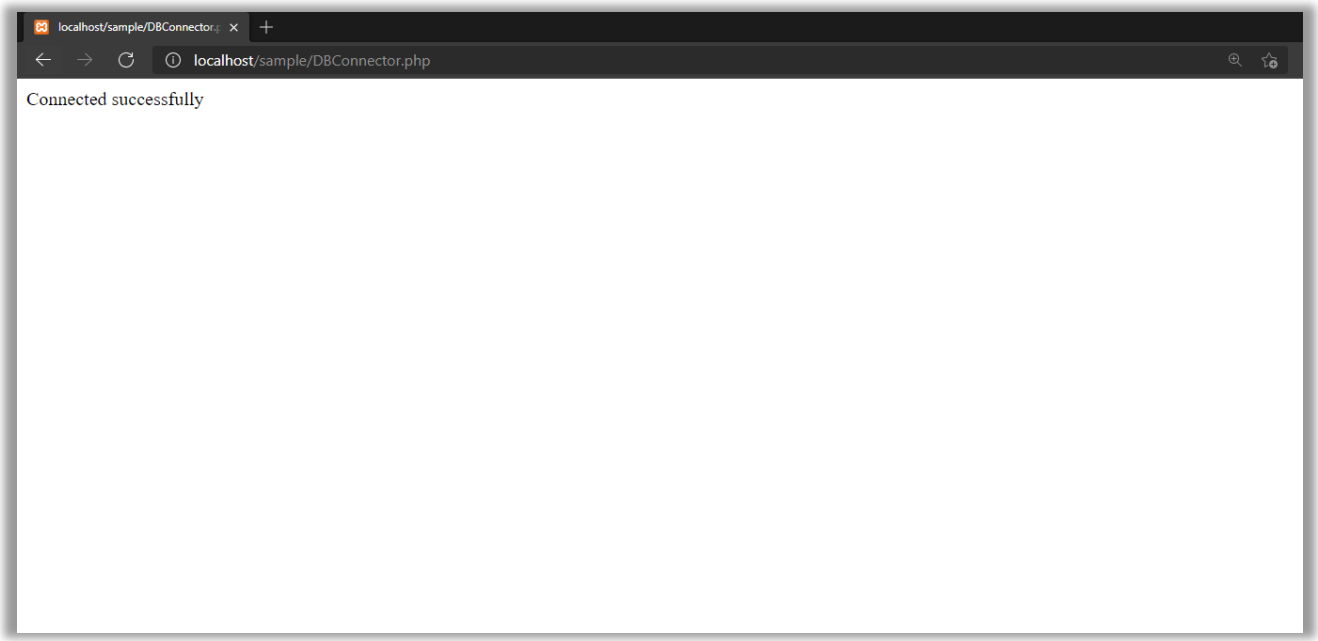
1. Open a Connection to MySQL. Type the code below in your text editor and save as **DBConnector.php** in **C:\xampp\htdocs\sample** directory (*create the folder **sample** inside* **C:\xampp\htdocs**).

```php
<?php

$servername = "localhost";
$username = "root"; //default username
$password = "";     //default password
$dbname = "sample";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

echo "Connected successfully <br/>";
?>
```

Before we can access data in the MySQL database, we need to be able to connect to the server.

2. Make sure that you **START** Apache and MySQL (in XAMPP), then access the **DBConnector.php** file in a web browser using the url **localhost/sample/DBConnector.php**

A message **Connect Successfully** should be displayed in your browser.



3. Type the code below in your text editor and save as **display.php** in **C:\xampp\htdocs\sample** directory (*it's the same directory where you saved your DBConnector.php)*

```php
<?php

include 'DBConnector.php';      //importsincludes all the statemens of the file specified

$sql = "SELECT * FROM employee";    //the query string is assigned into a variable named $sql
$result = $conn -> query($sql);     //executes the query and puts the resulting data into a variable called $result

if($result -> num_rows > 0){    //the function num_rows() checks if there are more than zero rows returned
    //output data of each row
    while($row = $result -> fetch_assoc()){     //the function fetch_assoc() puts all the results into an associative array that we can loop

        echo "<pre/>";      //white spaced and format of the next data to be echo/printed will be preserved
        print_r($row);      //prints the content of a variable in an array form

        echo "EmpID: ".$row["EmpID"]."<br>".    //<br/> is an HTML element that represents NEWLINE
            "- Name: " .$row["EmpName"].
            "- Age: " .$row["Age"].
            "- Salary: ".$row["Salary"].
            "- HireDate: ".$row["HireDate"].
            "<br/><br/>";
    }
}else{
    echo "0 results";
}

$conn -> close();   //closes the Database connection
?>
```

4.Access the **display.php** file in a browser using the url **localhost/sample/display.php**

display.php displays all the data stored in table ***employee***

Connected successfully

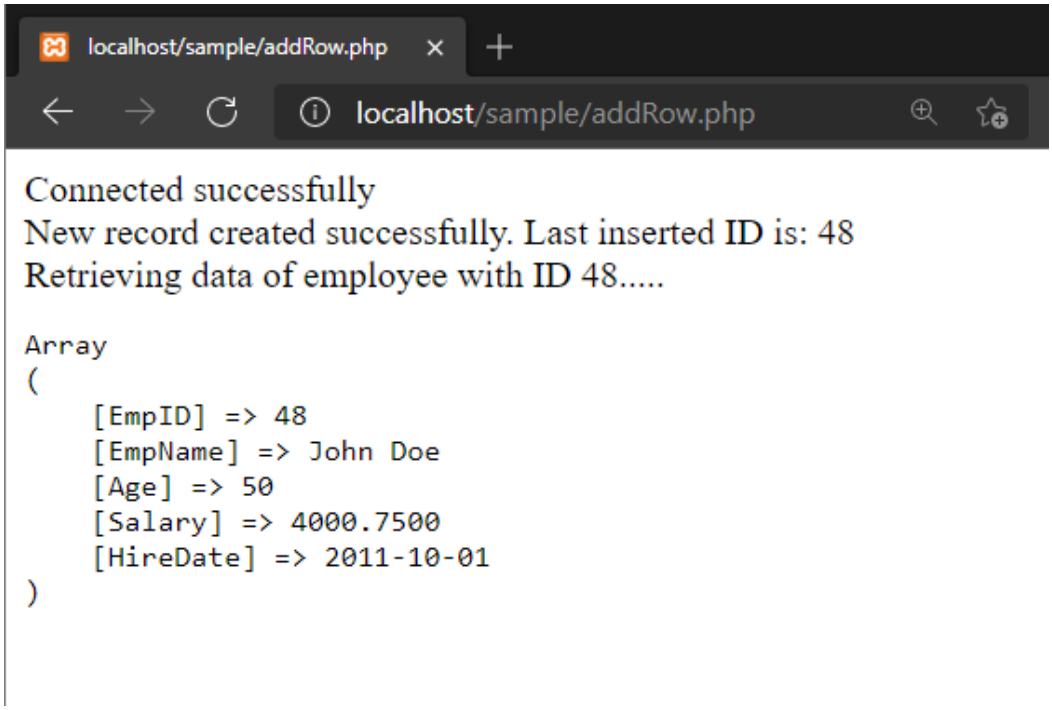
```
Array
(
    [EmpID] => 2
    [EmpName] => River Dale
    [Age] => 35
    [Salary] => 20000.0000
    [HireDate] => 2019-06-01
)
EmpID: 2
- Name: River Dale- Age: 35- Salary: 20000.0000- HireDate: 2019-06-01


Array
(
    [EmpID] => 3
    [EmpName] => Anthony Peterson
    [Age] => 38
    [Salary] => 21000.5000
    [HireDate] => 2019-07-01
)
EmpID: 3
- Name: Anthony Peterson- Age: 38- Salary: 21000.5000- HireDate: 2019-07-01


Array
(
    [EmpID] => 4
    [EmpName] => Robert Brickson
    [Age] => 40
    [Salary] => 19000.7500
    [HireDate] => 2019-10-01
)
EmpID: 4
- Name: Robert Brickson- Age: 40- Salary: 19000.7500- HireDate: 2019-10-01


Array
(
    [EmpID] => 5
```

5. In your text editor again, type the code below save as **addRow.php** in **C:\xampp\htdocs\sample** directory (*it's the same directory where you saved your DBConnector.php and display.php)*

```php
<?php

include 'DBConnector.php';

$sql = "INSERT INTO `employee` (`EmpID`, `EmpName`,`Age`, `Salary`, `HireDate`)
        VALUES (NULL, 'John Doe', '50', '4000.75', '2011-10-01');";
$result = $conn -> query($sql);

if($conn -> query($sql) == TRUE){
    $last_id = $conn -> insert_id;  //retrieving the ID of the newy inserted row
    echo "New record created successfully. Last inserted ID is: " .$last_id . "<br/>";

    echo "Retrieving data of employee with ID ".$last_id. ".....<br/>";

    $query = "SELECT * FROM `employee` WHERE `EmpID` = '$last_id';";
    $result = $conn -> query($query);
    echo "<pre />";
    print_r($result -> fetch_assoc());

}else{
    echo "Error: ".$sql . "<br/>" . $conn -> error ;
}

$conn -> close();
?>
```

6. Access the **addRow.php** file in a browser using the url **localhost/sample/addRow.php**

addRow.php will add a row of data in the *employee* table and show he ID of that new row along with its other attributes.



Take a screenshot of the entire screen of your laptop/desktop (should include the Taskbar of your desktop – **do not crop your screenshot**) for every output in your browser of **DBConnector.php, display.php, and addRow.php.**

Put those three (3) screenshots in a document file that you will submit as an output for this Laboratory exercise.

Submission Instructions:

Save and submit your file as **LastnameFirstLetterOfFirstName_Lab3.2.pdf.**

Please take note of the use of **Camel Case** for your filename. **Those who are not following instructions will have a 20% deduction on their Laboratory exercise**.

Upload your file in the UPV LMS (on or before May 02, 2022 11:59 PM).