**Batch: C1**

**Roll No.:1601012221**

**Experiment No. 08**

**TITLE: Design and Demonstrate database Connectivity between PHP & MYSQL**

**AIM:** Design and Demonstrate database Connectivity between PHP & MYSQL

## Expected OUTCOME of Experiment:

**CO5: Apply database operations by integrating SQL queries and session variables.**

## Books/ Journals/ Websites referred:

1. Head First HTML5 Programming published by Shroff/O'Reilly in 2011. 2. HTML, XHTML, and CSS Bible, 5th Edition By [Steven Schafe.](http://shop.oreilly.com/product/9780470523964.do)

1. <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>
2. <https://www.w3schools.com/php/php_mysql_connect.asp>

**Pre Lab/ Prior Concepts:**

# PHP

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. This tutorial helps you to build your base with PHP.

The different server side technologies should be considered in the design of web pages. The technology used can be PHP, ASP, JSP, ASP.NET etc.

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

* + PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
  + PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
  + It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
  + PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
  + PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
  + PHP is forgiving: PHP language tries to be as forgiving as possible.
  + PHP Syntax is C-Like.

***Common uses of PHP***

* + PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
  + PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
  + You add, delete, modify elements within your database through PHP.
  + Access cookies variables and set cookies.
  + Using PHP, you can restrict users to access some pages of your website.
  + It can encrypt data.

***Characteristics of PHP***

Five important characteristics make PHP's practical nature possible −

* + Simplicity
  + Efficiency
  + Security
  + Flexibility
  + Familiarity

*"Hello World" Script in PHP*

To get a feel for PHP, first start with simple PHP scripts. Since "Hello, World!" is an essential example, first we will create a friendly little "Hello, World!" script.

As mentioned earlier, PHP is embedded in HTML. That means that in amongst your normal HTML (or XHTML if you're cutting-edge) you'll have PHP statements like this −

<html>

<head>

<title>Hello World</title>

</head>

<body>

<?php echo "Hello, World!";?>

</body>

</html>

It will produce following result −

Hello, World!

If you examine the HTML output of the above example, you'll notice that the PHP code is not present in the file sent from the server to your Web browser. All of the PHP present in the Web page is processed and stripped from the page; the only thing returned to the client from the Web server is pure HTML output.

All PHP code must be included inside one of the three special markup tags ATE are recognised by the PHP Parser.

<?php PHP code goes here ?>

<? PHP code goes here ?>

<script language="php"> PHP code goes here </script>

A most common tag is the <?php...?> and we will also use the same tag in our tutorial.

From the next chapter we will start with PHP Environment Setup on your machine and then we will dig out almost all concepts related to PHP to make you comfortable with the PHP language.

***Installation of Php:***

In order to develop and run PHP Web pages three vital components need to be installed on your computer system.

* + Web Server − PHP will work with virtually all Web Server software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server. Download Apache for free here − <https://httpd.apache.org/download.cgi>
  + Database − PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database. Download MySQL for free here

− <https://www.mysql.com/downloads/>

* + PHP Parser − In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web Browser. This tutorial will guide you how to install PHP parser on your computer.

The [INSERT INTO](https://www.tutorialrepublic.com/sql-tutorial/sql-insert-statement.php) statement is used to insert new rows in a database table. Syntax Insert Data Into MySQL

1. The SQL query must be quoted in PHP
2. String values inside the SQL query must be quoted
3. Numeric values must not be quoted
4. The word NULL must not be quoted

The INSERT INTO statement is used to add new records to a MySQL table: INSERT INTO table\_name (column1, column2, column3,...)

VALUES (value1, value2, value3,...)

***Inserting Data into a MySQL Database Table***

Now that you've understood how to create database and tables in MySQL. In this experiment you will learn how to execute SQL query to insert records into a table.

The [INSERT INTO](https://www.tutorialrepublic.com/sql-tutorial/sql-insert-statement.php) statement is used to insert new rows in a database table.

Let's make a SQL query using the INSERT INTO statement with appropriate values, after that we will execute this insert query through passing it to the PHP mysqli\_query () function to insert data in table.

**Problem Statement:**

1. Set up a MySQL database using PHP and MySQL connectivity, and create a table named "students" with fields: Name, Rollno, Div, and Address.

<?php

$servername = "localhost:3308";

$username = "root";

$password = "";

$dbname = "databse\_school";

$conn = mysqli\_connect($servername, $username, $password);

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

$result = mysqli\_query($conn, "SELECT SCHEMA\_NAME FROM INFORMATION\_SCHEMA.SCHEMATA WHERE SCHEMA\_NAME = '$dbname'");

if (mysqli\_num\_rows($result) == 0) {

if (mysqli\_query($conn, "CREATE DATABASE $dbname")) { echo "Database created successfully!<br>";

} else {

echo "Error creating database: " . mysqli\_error($conn) . "<br>";

}

}

mysqli\_select\_db($conn, $dbname);

$sql = "CREATE TABLE IF NOT EXISTS students ( Name VARCHAR(100) NOT NULL,

Rollno INT(10) NOT NULL PRIMARY KEY, Division VARCHAR(10) NOT NULL, Address VARCHAR(255) NOT NULL

)";

if (mysqli\_query($conn, $sql)) {

echo "Table 'students' has been created successfully!";

} else {

echo "Error creating table: " . mysqli\_error($conn);

}

mysqli\_close($conn);

1. Develop an HTML form with fields for Name, Rollno, Div, and Address, styled using CSS.

<!DOCTYPE html>

<html>

<head>

<style>

body {

font-family: Arial, sans-serif;

}

form {

width: 500px; margin: 0 auto;

}

label {

display: block; margin-top: 10px;

}

input, select { width: 100%; padding: 10px; margin-top: 5px;

}

button {

background-color: #4CAF50; color: white;

padding: 10px;margin-top: 10px; border: none; cursor: pointer;

}

</style>

</head>

<body>

<form action="expp9.php" method="post">

<label for="name">Name:</label>

<input type="text" name="name" id="name" required>

<label for="rollno">Rollno:</label>

<input type="number" name="rollno" id="rollno" required>

<label for="div">Div:</label>

<select name="div" id="div" required>

<option value="A">A</option>

<option value="B">B</option>

<option value="C">C</option>

<option value="D">D</option>

</select>

<label for="address">Address:</label>

<input type="text" name="address" id="address" required>

<button type="submit" name="submit">Submit</button>

</form>

</body>

</html>

1. Write PHP code to dynamically insert form values into the "students" table in the MySQL database.

<?php

$servername = "localhost:3308";

$username = "root";

$password = "";

$dbname = "database\_school";

$conn = mysqli\_connect($servername, $username, $password, $dbname);

if (!$conn) {

    die("Connection failed: " . mysqli\_connect\_error());

}

$create\_table\_sql = "CREATE TABLE IF NOT EXISTS students (

    ID INT AUTO\_INCREMENT PRIMARY KEY,

    Name VARCHAR(255) NOT NULL,

    Rollno VARCHAR(20) NOT NULL,

    Division VARCHAR(10) NOT NULL,

    Address VARCHAR(255) NOT NULL

)";

if (mysqli\_query($conn, $create\_table\_sql)) {

    echo "Table 'students' created successfully or already exists.<br>";

} else {

    echo "Error creating table: " . mysqli\_error($conn) . "<br>";

}

if (isset($\_POST['submit'])) {

    $name = mysqli\_real\_escape\_string($conn, $\_POST['name']);

    $rollno = mysqli\_real\_escape\_string($conn, $\_POST['rollno']);

    $div = mysqli\_real\_escape\_string($conn, $\_POST['div']);

    $address = mysqli\_real\_escape\_string($conn, $\_POST['address']);

    $insert\_sql = "INSERT INTO students (Name, Rollno, Division, Address) VALUES ('$name', '$rollno', '$div', '$address')";

    if (mysqli\_query($conn, $insert\_sql)) {

        echo "New record inserted successfully";

    } else {

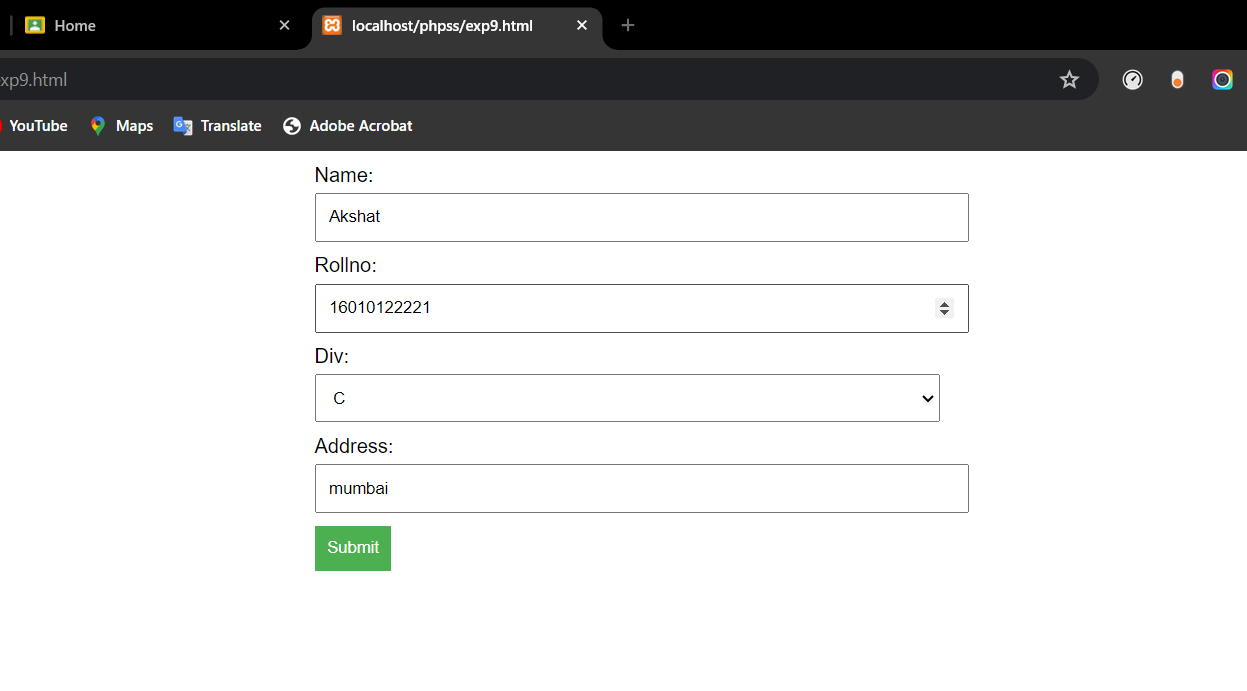
        echo "Error: " . $insert\_sql . "<br>" . mysqli\_error($conn);

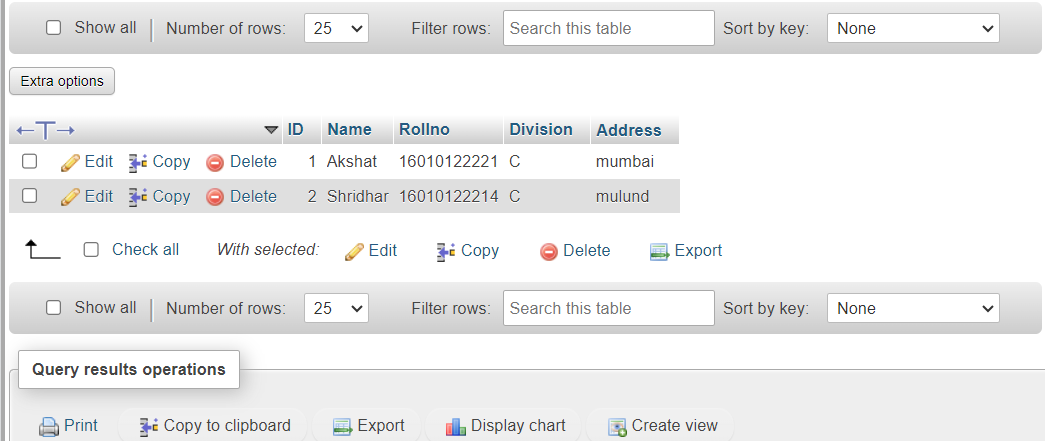
    }

}

mysqli\_close($conn);

?>





**Post Lab Descriptive Questions:**

# What are the advantages of server side technologies that you used?

Advantages of server-side technologies:

* + Scalability: Handle increased traffic and user demands by distributing the workload across multiple servers.
  + Security: Centralize sensitive operations and data, implementing authentication, access controls, and encryption.
  + Performance: Offload resource-intensive tasks to the server, improving application performance.
  + Code Protection: Keep server-side code hidden, protecting intellectual property and business logic.
  + Cross-Platform Compatibility: Develop applications compatible with various browsers and devices.
  + Data Integrity: Centralized data management ensures consistency and reduces the risk of erroneous modifications.
  + Maintenance and Updates: Centralized updates simplify deployment and ensure all clients have the latest version.
  + Integration and Extensibility: Integrate with external systems and extend functionality through APIs and plugins.

# Which function is used for database connectivity in PHP?

Answer: Code with output screenshots

<?php

$servername = "localhost:3308";

$username = "root";

$password = "";

$dbname = "school";

$conn = mysqli\_connect($servername, $username, $password);

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

$result = mysqli\_query($conn, "SELECT SCHEMA\_NAME FROM INFORMATION\_SCHEMA.SCHEMATA WHERE SCHEMA\_NAME = '$dbname'");

if (mysqli\_num\_rows($result) == 0) {

if (mysqli\_query($conn, "CREATE DATABASE $dbname")) { echo "Database created successfully!<br>";

} else {

echo "Error creating database: " . mysqli\_error($conn) . "<br>";

}

}



1. How would you redirect the page in PHP? Action method

<form action="insert.php" method="post">

<label for="name">Name:</label>

<input type="text" name="name" id="name" required>

<label for="rollno">Rollno:</label>

<input type="number" name="rollno" id="rollno" required>

<label for="div">Div:</label>

<select name="div" id="div" required>

<option value="A">A</option>

<option value="B">B</option>

<option value="C">C</option>

<option value="D">D</option>

</select>

<label for="address">Address:</label>

<input type="text" name="address" id="address" required>

<button type="submit" name="submit">Submit</button>

</form

Answer: Code with output screenshots