Batch: A1 **Roll No.: 16010123012**

Experiment No. 08

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE: Design and Demonstrate database Connectivity between PHP & MYSQL

AIM: Design and Demonstrate database Connectivity between PHP & MYSQL

Expected OUTCOME of Experiment:

CO: Establish a connection between PHP and MySQL, perform basic database operations such as inserting, retrieving, updating, and deleting records.

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.
3. https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5
4. 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 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 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.

Implementation Details:

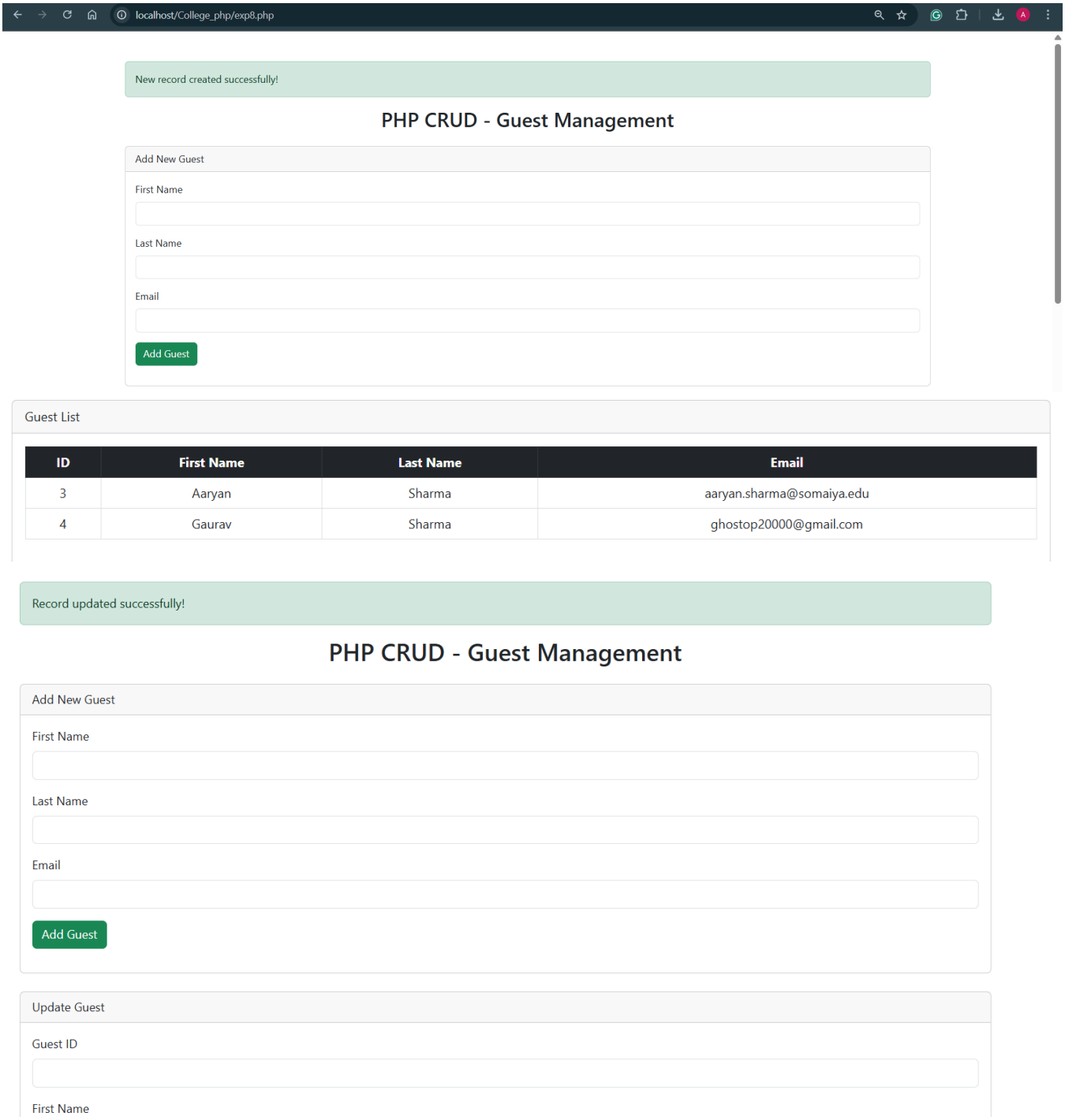
```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "exp8";

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

if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

if (isset($_POST['create'])) {
  $firstname = $_POST['firstname'];
  $lastname = $_POST['lastname'];
  $email = $_POST['email'];

  $stmt = $conn->prepare("INSERT INTO exp8 (firstname, lastname, email) VALUES (?, ?,
?)");
  $stmt->bind_param("sss", $firstname, $lastname, $email);

  if ($stmt->execute()) {
    echo "<div class='alert alert-success'>New record created successfully!</div>";
  } else {
    echo "<div class='alert alert-danger'>Error: " . $stmt->error . "</div>";
  }
  $stmt->close();
}
```

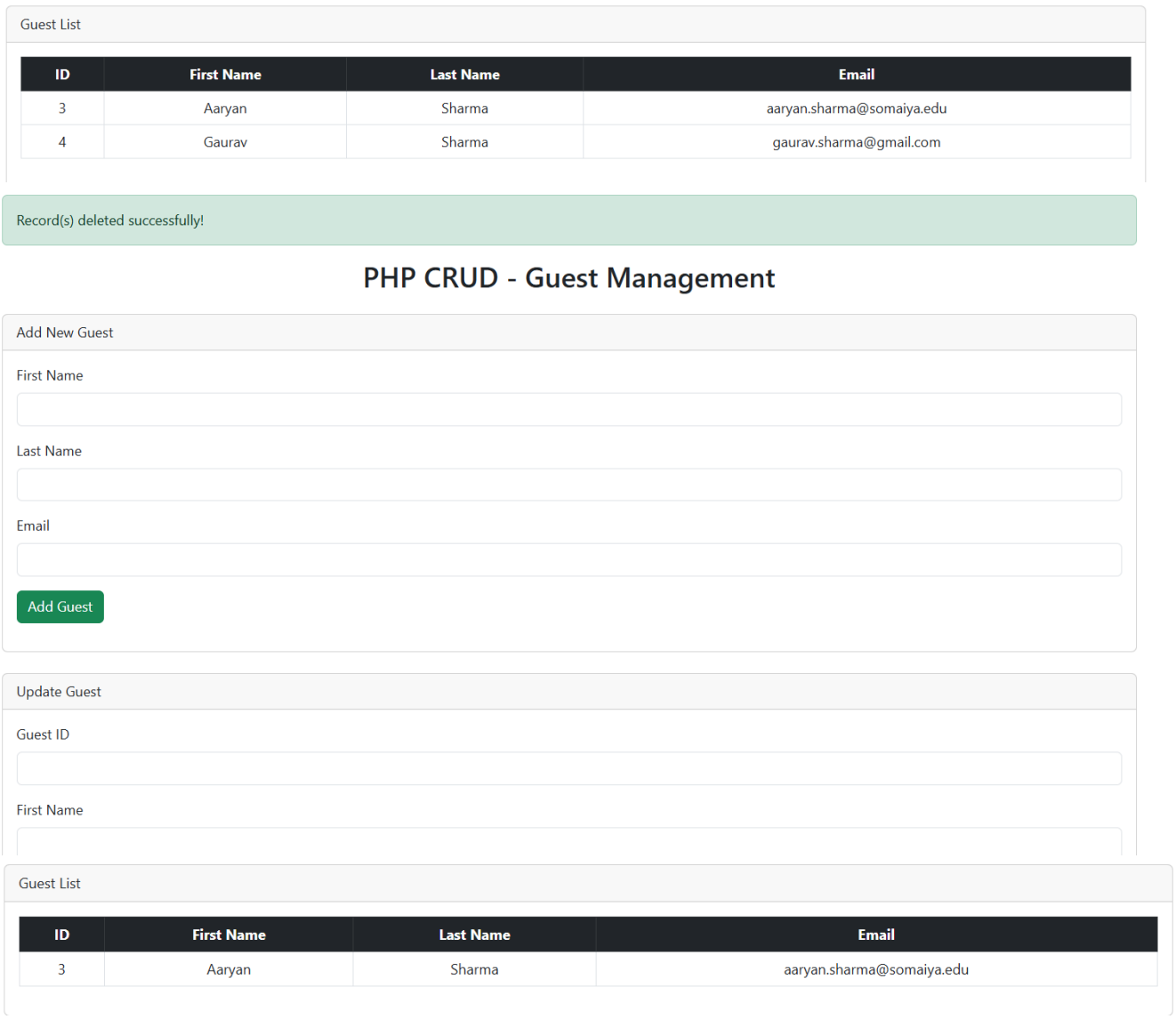
```php
if (isset($_POST['update'])) {
  $id = $_POST['id'];
  $firstname = $_POST['firstname'];
  $lastname = $_POST['lastname'];
  $email = $_POST['email'];

  $stmt = $conn->prepare("UPDATE exp8 SET firstname=?, lastname=?, email=? WHERE
id=?");
  $stmt->bind_param("sssi", $firstname, $lastname, $email, $id);

  if ($stmt->execute()) {
    echo "<div class='alert alert-success'>Record updated successfully!</div>";
  } else {
    echo "<div class='alert alert-danger'>Error updating record: " . $stmt->error .
"</div>";
  }
  $stmt->close();
}

if (isset($_POST['delete'])) {
  $firstname = $_POST['firstname'];

  if (!empty($firstname)) {
    $stmt = $conn->prepare("DELETE FROM exp8 WHERE firstname=?");
    $stmt->bind_param("s", $firstname);

    if ($stmt->execute()) {
      echo "<div class='alert alert-success'>Record(s) deleted successfully!</div>";
    } else {
      echo "<div class='alert alert-danger'>Error deleting record: " . $stmt->error .
"</div>";
    }
    $stmt->close();
  } else {
    echo "<div class='alert alert-warning'>Please enter a valid first name!</div>";
  }
}

$sql = "SELECT * FROM exp8";
$result = $conn->query($sql);
```

```
?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>PHP CRUD - MyGuests</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>

<body class="container mt-5">
  <h2 class="text-center mb-4">PHP CRUD - Guest Management</h2>

  <div class="card mb-4">
    <div class="card-header">Add New Guest</div>
    <div class="card-body">
      <form method="POST">
        <div class="mb-3">
          <label class="form-label">First Name</label>
          <input type="text" name="firstname" class="form-control" required>
        </div>
        <div class="mb-3">
          <label class="form-label">Last Name</label>
          <input type="text" name="lastname" class="form-control" required>
        </div>
        <div class="mb-3">
          <label class="form-label">Email</label>
          <input type="email" name="email" class="form-control" required>
        </div>
        <button type="submit" name="create" class="btn btn-success">Add Guest</button>
      </form>
    </div>
  </div>

  <div class="card mb-4">
    <div class="card-header">Update Guest</div>
    <div class="card-body">
      <form method="POST">
```

```
        <div class="mb-3">
          <label class="form-label">Guest ID</label>
          <input type="number" name="id" class="form-control" required>
        </div>
        <div class="mb-3">
          <label class="form-label">First Name</label>
          <input type="text" name="firstname" class="form-control" required>
        </div>
        <div class="mb-3">
          <label class="form-label">Last Name</label>
          <input type="text" name="lastname" class="form-control" required>
        </div>
        <div class="mb-3">
          <label class="form-label">Email</label>
          <input type="email" name="email" class="form-control" required>
        </div>
        <button type="submit" name="update" class="btn btn-primary">Update
Guest</button>
      </form>
    </div>
  </div>

  <div class="card mb-4">
    <div class="card-header">Delete Guest by Name</div>
    <div class="card-body">
      <form method="POST">
        <div class="mb-3">
          <label class="form-label">Enter First Name</label>
          <input type="text" name="firstname" class="form-control" required>
        </div>
        <button type="submit" name="delete" class="btn btn-danger">Delete
Guest</button>
      </form>
    </div>
  </div>

  <div class="card">
    <div class="card-header">Guest List</div>
    <div class="card-body">
      <table class="table table-bordered text-center">
```

```
        <thead class="table-dark">
          <tr>
            <th>ID</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
          </tr>
        </thead>
        <tbody>
          <?php
          if ($result->num_rows > 0) {
            while ($row = $result->fetch_assoc()) {
              echo "<tr>
                    <td>{$row['id']}</td>
                    <td>{$row['firstname']}</td>
                    <td>{$row['lastname']}</td>
                    <td>{$row['email']}</td>
                  </tr>";
            }
          } else {
            echo "<tr><td colspan='4' class='text-center'>No records found</td></tr>";
          }
          $conn->close();
          ?>
        </tbody>
      </table>
    </div>
  </div>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></sc
ript>
</body>
</html>
```


localhost/College_php/exp8.php
New record created successfully!
PHP CRUD - Guest Management
Add New Guest
First Name
Last Name
Email
Add Guest
Guest List
ID
First Name
Last Name
Email
3
Aaryan
Sharma
aaryan.sharma@somaiya.edu
4
Gaurav
Sharma
ghostop20000@gmail.com
Record updated successfully!
PHP CRUD - Guest Management
Add New Guest
First Name
Last Name
Email
Add Guest
Update Guest
Guest ID
First Name



Steps for execution of the code:

1. **Create a MySQL Database and Table**
   Open phpMyAdmin
   Create a new database named exp8
2. **Configure the PHP script and run the application**
   a. Write the code
   b. Run the application in a browser using XAMPP

3. **Perform CRUD Operations**
   Create a New Guest:
   a. Fill in the First Name, Last Name, and Email fields
   b. Click "Add Guest" to insert the record into the database
   c. A success message will appear if the record is inserted
   Update Guest Details:
   a. Enter the Guest ID (found in the table)
   b. Modify the First Name, Last Name, or Email fields
   c. Click "Update Guest" to update the details
   Delete a Guest by First Name:
   a. Enter the First Name of the guest you want to delete
   b. Click "Delete Guest" to remove the record
   View Guest List:
   The table at the bottom displays all guests stored in the database

Conclusion:
I have successfully implemented database connectivity in PHP using MySQL. Through this experiment, I learned how to establish a secure connection, execute CRUD operations, and handle data efficiently. This experiment gave me hands-on experience in integrating PHP with databases, which is essential for developing dynamic web applications.

Post Lab Descriptive Questions:

1) What are the advantages of server-side technologies that you used?

In this experiment, I used PHP as the server-side scripting language and MySQL as the database, which provided several advantages. PHP efficiently processes user requests and handles input on the server before sending responses to the client, reducing the workload on the client-side. MySQL allowed me to store, retrieve, and manipulate data dynamically, ensuring smooth data management. One of the key benefits of server-side scripting is enhanced security, as sensitive operations such as authentication and database transactions remain hidden from the client. Another advantage of using PHP is its ability to generate dynamic content. It processes logic and data before sending the final HTML output, making web pages more interactive and responsive to user actions. Additionally, PHP and MySQL are cross-platform compatible, meaning they work seamlessly across various operating systems such as Windows, macOS, and Linux, and are supported by popular web servers like Apache and Nginx. Scalability is

another strong point of these technologies, as they can efficiently handle large volumes of data and multiple user requests without performance issues. Moreover, I integrated Bootstrap with PHP to enhance the user interface, making the web application more visually appealing and mobile-friendly. Overall, using server-side technologies like PHP and MySQL enabled me to build a secure, efficient, and dynamic web application that effectively processes data while maintaining an optimized user experience.

2) Which function is used for database connectivity in PHP?

```php
<?php
$server = "localhost";
$user = "root";
$pass = "";
$db = "test_db";

$conn = mysqli_connect($server, $user, $pass, $db);

if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}
?>
```

3) How would you redirect the page in PHP?

```php
<?php
header("Location: home.php");
exit();
?>
```

Date: 29 / 03 / 2025 **Signature of faculty in-charge**