University of Colombo School of Computing
SCS 1311 - Internet and Web Technologies
Lab Sheet – 09

Task 1: Establishing a Database Connection

1. Open your XAMPP/WAMP control panel and start Apache and MySQL.
2. Create a PHP script to establish a connection to the MySQL database.
3. Use MySQLi (both object-oriented and procedural) and PDO for database connections.

Example Code:

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

// MySQLi Object-Oriented
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
$conn->close();
?>
```

Task 2: Creating a Database and Tables

1. Write a PHP script to create a new database example_db.
2. Create a table persons with fields: id (Primary Key, Auto Increment), first_name, last_name, age.

Example Code:

```
<?php
$conn = new mysqli("localhost", "root", "");
$conn->query("CREATE DATABASE example_db");
$conn->close();
?>

<?php
$conn = new mysqli("localhost", "root", "", "example_db");
```

```
$sql = "CREATE TABLE persons (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(30),
    last_name VARCHAR(30),
    age INT
    )";
$conn->query($sql);
$conn->close();
?>
```

Task 3: Insert Data into Table

1. Insert at least five records into the persons table using an SQL query.
2. Insert data dynamically using an HTML form and PHP script.

Example SQL Query:

```
$sql = "INSERT INTO persons (first_name, last_name, age) VALUES ('John', 'Doe', 25)";
$conn->query($sql);
```

Example Form:

```
<form action="insert.php" method="post">
    First Name: <input type="text" name="first_name"><br>
    Last Name: <input type="text" name="last_name"><br>
    Age: <input type="number" name="age"><br>
    <input type="submit" value="Submit">
</form>
```

Processing Script (insert.php):

```
<?php
$conn = new mysqli("localhost", "root", "", "example_db");
$sql = "INSERT INTO persons (first_name, last_name, age) VALUES (?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ssi", $_POST['first_name'], $_POST['last_name'], $_POST['age']);
$stmt->execute();
$stmt->close();
$conn->close();
echo "Record inserted successfully";
?>
```

Task 4: Retrieving Data

1. Write a PHP script to fetch and display all records from the persons table.
2. Display the results in an HTML table.

Example Code:

```
<?php
$conn = new mysqli("localhost", "root", "", "example_db");
$result = $conn->query("SELECT * FROM persons");
if ($result->num_rows > 0) {
   echo "<table border='1'><tr><th>First Name</th><th>Last Name</th><th>Age</th></tr>";
   while ($row = $result->fetch_assoc()) {
      echo "<tr><td>" . $row['first_name'] . "</td><td>" . $row['last_name'] . "</td><td>" .
$row['age'] . "</td></tr>";
   }
   echo "</table>";
} else {
   echo "No records found";
}
$conn->close();
?>
```

Task 5: Updating and Deleting Records

1. Write a PHP script to update a person’s age based on their name.
2. Write a PHP script to delete a record from the table.

Example Update Code:

```
$sql = "UPDATE persons SET age = 30 WHERE first_name = 'John'";
$conn->query($sql);
```

Example Delete Code:

```
$sql = "DELETE FROM persons WHERE first_name = 'John'";
$conn->query($sql);
```

Task 6: Using Prepared Statements to Prevent SQL Injection

1. Modify the insertion script to use prepared statements.
2. Implement prepared statements for updating and deleting records.

Example Prepared Statement for Update:

```
$stmt = $conn->prepare("UPDATE persons SET age = ? WHERE first_name = ?");
$stmt->bind_param("is", $new_age, $first_name);
```

```
$stmt->execute();
$stmt->close();
```

Task 7: Implementing SQL Joins

1. Create another table orders with fields: order_id (Primary Key), customer_id (Foreign Key), order_date.
2. Write a PHP script to join persons and orders using INNER JOIN.

Example SQL Query:

```
$sql = "SELECT persons.first_name, persons.last_name, orders.order_date FROM persons
INNER JOIN orders ON persons.id = orders.customer_id";
$result = $conn->query($sql);
```

Submission Instructions

- Upload all PHP files and screenshots of successful outputs to the LMS.
- Ensure proper indentation and comments in the code.