

# **Programming in PHP**

## **Lab06**



**Spring 2025**

Submitted by: **Mohsin Sajjad**

Registration No: **22pwsce2149**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_

Submitted to:

**Engr. Summeya Salahudin**

Month Day, Year (13 05, 2025)

Department of Computer Systems Engineering  
University of Engineering and Technology, Peshawar

## OBJECTIVES OF THE LAB

This lab aims at the understanding of:

- Type of MySQL Drivers in PHP
- PHP-MySQL Interfaces

### Task 6.1

Complete Example 3 by providing a suitable query, its result, and then freeing result variable.

#### Code:

```
<?php

//step 1. Database Connection + Database Selection
//server name, user, user's password, and database
//must be provided.
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "mohsinproject";

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

//step 2. Check Connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>

<!DOCTYPE html>
<html>
<head>
    <title>PHP MySQLi (Object-Oriented) Basics</title>
</head>
<body>
<?php
    // query code
    //step 3. Perform query or operation on database
    $sql = "SELECT id, name, email FROM users";
    $result = $conn->query($sql);

    //step 4. Show results
    if ($result->num_rows > 0) {
```

```

        // output data of each row
        while($row = $result->fetch_assoc()) {
            echo "ID: " . $row["id"]. " - Name: " . $row["name"]. " - Email: " .
$row["email"]. "<br>";
        }
    } else {
        echo "0 results";
    }

    //step 5. Free result set
    $result->free();
?>
</body>
</html>

<?php
//step 6. Close Database Connection
$conn->close();
?>

```

## Output:



## Task 6.2

Write PHP-MYSQLI Object-Oriented interface script that connects with one of the tables of your DBMS Lab Project (For instance, user table or accounts table or any of your choice). Perform the following: a. Apply the insert command on the table. Take the data from user via HTML Form constructs. b. Apply the update command on the table. Take the data from user via HTML Form constructs. c. Apply the delete query on the table. Take the data from user via HTML Form constructs. d. Apply the select query on the same table to retrieve data. Show the data in HTML Form Constructs.

## CODE:

```

<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "mohsinproject";

```

```

// Create connection
$conn = new mysqli($servername, $username, $password, $database);

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

// INSERT
if (isset($_POST['insert'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);

    $sql = "INSERT INTO users (name, email, password, created_at, updated_at)
            VALUES ('$name', '$email', '$password', NOW(), NOW())";
    $conn->query($sql);
}

// UPDATE
if (isset($_POST['update'])) {
    $id = $_POST['id'];
    $name = $_POST['name'];
    $email = $_POST['email'];
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);

    $sql = "UPDATE users
            SET name='$name', email='$email', password='$password',
updated_at=NOW()
            WHERE id=$id";
    $conn->query($sql);
}

// DELETE
if (isset($_POST['delete'])) {
    $id = $_POST['id'];
    $sql = "DELETE FROM users WHERE id=$id";
    $conn->query($sql);
}
?>

<!DOCTYPE html>
<html>
<head>

```

```

<title>Users CRUD - Mohsin Project</title>
</head>
<body>
    <h2>Insert User</h2>
    <form method="post">
        Name: <input type="text" name="name" required><br>
        Email: <input type="email" name="email" required><br>
        Password: <input type="password" name="password" required><br>
        <input type="submit" name="insert" value="Insert">
    </form>

    <h2>Update User</h2>
    <form method="post">
        ID: <input type="number" name="id" required><br>
        New Name: <input type="text" name="name"><br>
        New Email: <input type="email" name="email"><br>
        New Password: <input type="password" name="password"><br>
        <input type="submit" name="update" value="Update">
    </form>

    <h2>Delete User</h2>
    <form method="post">
        ID: <input type="number" name="id" required><br>
        <input type="submit" name="delete" value="Delete">
    </form>

    <h2>All Users</h2>
    <?php
    $sql = "SELECT id, name, email, created_at FROM users";
    $result = $conn->query($sql);

    echo "<table border='1'
    cellpadding='5'><tr><th>ID</th><th>Name</th><th>Email</th><th>Created
    At</th></tr>";
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            echo "<tr>
                <td>{$row['id']}</td>
                <td>{$row['name']}</td>
                <td>{$row['email']}</td>
                <td>{$row['created_at']}</td>
            </tr>";
        }
    } else {
        echo "<tr><td colspan='4'>No users found</td></tr>";
    }

```

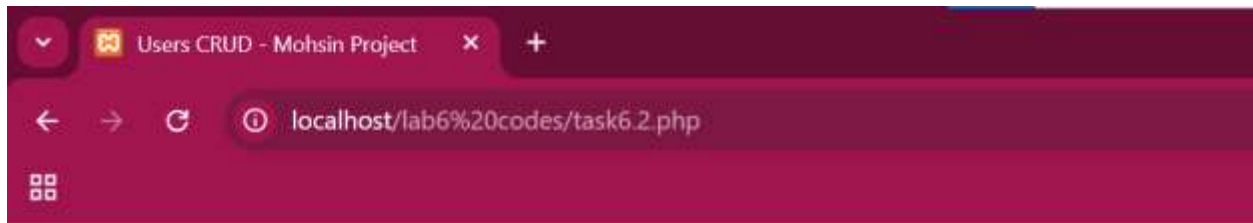
```

    }
    echo "</table>";

    $result->free();
    $conn->close();
    ?>
</body>
</html>

```

## Output:



### Insert User

Name:

Email:

Password:

### Update User

ID:

New Name:

New Email:

New Password:

### Delete User

ID:

### All Users

ID	Name	Email	Created At
2	Afnan khan	afnankhan@gmail.com	2025-05-16 11:47:56
3	hassan	hassan@gmail.com	2025-05-16 11:50:17
4	Mohsin Sajjad	22pwcse2149@uetpeshawar.edu.pk	2025-05-16 06:53:07

## Task 6.3

Perform Task 6.2 using PHP-PDO (PHP Data Objects) interface.

### Code:

```
<?php

$dsn = 'mysql:host=localhost;dbname=mohsinproject;charset=utf8';
$username = 'root';
$password = '';

try {
    $pdo = new PDO($dsn, $username, $password);
    // Enable exceptions
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // INSERT
    if (isset($_POST['insert'])) {
        $name = $_POST['name'];
        $email = $_POST['email'];
        $passwordHash = password_hash($_POST['password'], PASSWORD_DEFAULT);

        $stmt = $pdo->prepare("INSERT INTO users (name, email, password,
created_at, updated_at)
                                VALUES (?, ?, ?, NOW(), NOW())");
        $stmt->execute([$name, $email, $passwordHash]);
    }

    // UPDATE
    if (isset($_POST['update'])) {
        $id = $_POST['id'];
        $name = $_POST['name'];
        $email = $_POST['email'];
        $passwordHash = password_hash($_POST['password'], PASSWORD_DEFAULT);

        $stmt = $pdo->prepare("UPDATE users SET name=?, email=?, password=?,
updated_at=NOW() WHERE id=?");
        $stmt->execute([$name, $email, $passwordHash, $id]);
    }

    // DELETE
    if (isset($_POST['delete'])) {
        $id = $_POST['id'];

        $stmt = $pdo->prepare("DELETE FROM users WHERE id=?");
        $stmt->execute([$id]);
    }
}
```

```

    }

    // SELECT
    $stmt = $pdo->query("SELECT id, name, email, created_at FROM users");
    $users = $stmt->fetchAll(PDO::FETCH_ASSOC);

} catch (PDOException $e) {
    echo "Database error: " . $e->getMessage();
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Users CRUD with PDO</title>
</head>
<body>
    <h2>Insert User</h2>
    <form method="post">
        Name: <input type="text" name="name" required><br>
        Email: <input type="email" name="email" required><br>
        Password: <input type="password" name="password" required><br>
        <input type="submit" name="insert" value="Insert">
    </form>

    <h2>Update User</h2>
    <form method="post">
        ID: <input type="number" name="id" required><br>
        New Name: <input type="text" name="name"><br>
        New Email: <input type="email" name="email"><br>
        New Password: <input type="password" name="password"><br>
        <input type="submit" name="update" value="Update">
    </form>

    <h2>Delete User</h2>
    <form method="post">
        ID: <input type="number" name="id" required><br>
        <input type="submit" name="delete" value="Delete">
    </form>

    <h2>All Users</h2>
    <table border="1" cellpadding="5">
        <tr><th>ID</th><th>Name</th><th>Email</th><th>Created At</th></tr>
        <?php if (!empty($users)) {
            foreach ($users as $user) {

```



```

        echo "<tr>
            <td>{$user['id']}

```

## Output:

**Insert User**

Name:   
 Email:   
 Password:

**Update User**

ID:   
 New Name:   
 New Email:   
 New Password:

**Delete User**

ID:

**All Users**

ID	Name	Email	Created At
2	Afsan Khan	afsan.khan@gmail.com	2025-05-16 11:47:56
3	hassan	hassan@gmail.com	2025-05-16 11:50:17
4	Mohsin Sajjad	22pcsc2149@ustpeduamar.edu.pk	2025-05-16 06:53:07

**Update User**

ID:   
 New Name:   
 New Email:   
 New Password:

**Delete User**

ID:

**All Users**

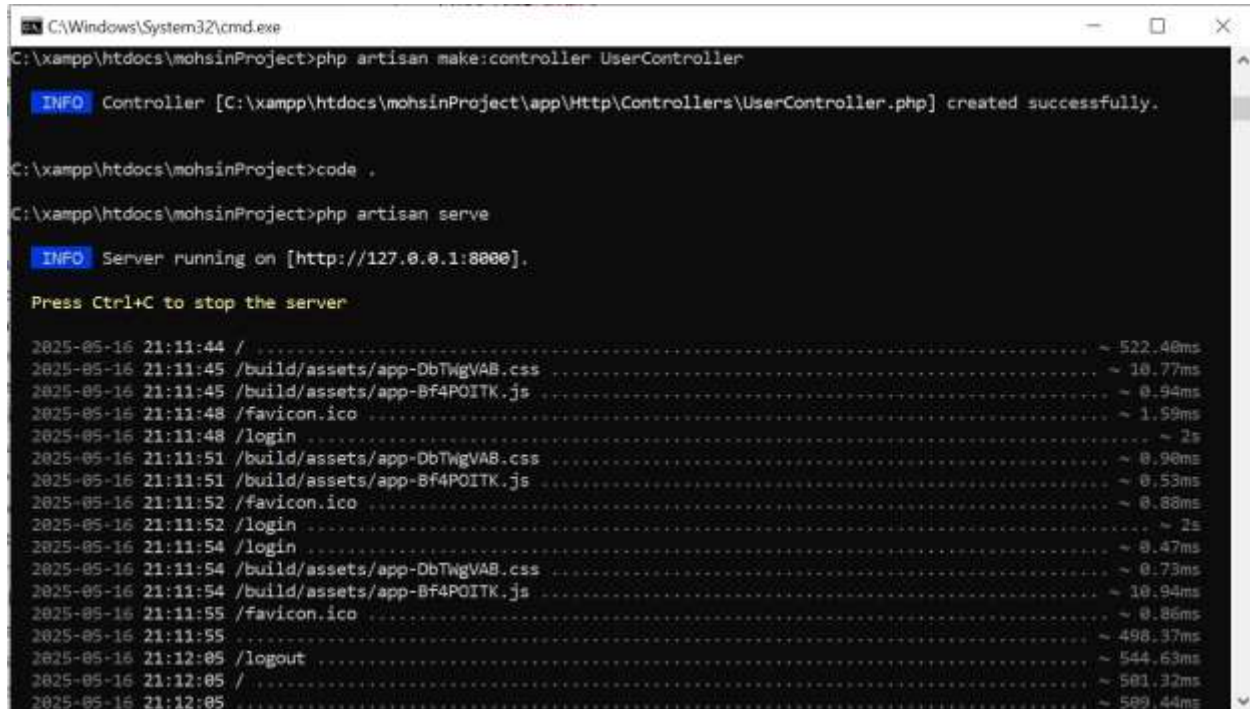
ID	Name	Email	Created At
2	Afsan Khan	afsan.khan@gmail.com	2025-05-16 11:47:56
3	hassan	hassan@gmail.com	2025-05-16 11:50:17
4	Mohsin Sajjad	22pcsc2149@ustpeduamar.edu.pk	2025-05-16 06:53:07
5	mohsin	mohsin@gmail.com	2025-05-16 11:58:06

## Task 6.4

Perform Task 6.2 using LARAVEL

### Step 1:

Code .



```
C:\Windows\System32\cmd.exe
C:\xampp\htdocs\mohsinProject>php artisan make:controller UserController

INFO Controller [C:\xampp\htdocs\mohsinProject\app\Http\Controllers\UserController.php] created successfully.

C:\xampp\htdocs\mohsinProject>code .

C:\xampp\htdocs\mohsinProject>php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

2025-05-16 21:11:44 / ..... ~ 522.40ms
2025-05-16 21:11:45 /build/assets/app-DbTvgVAB.css ..... ~ 10.77ms
2025-05-16 21:11:45 /build/assets/app-Bf4POITK.js ..... ~ 0.94ms
2025-05-16 21:11:48 /favicon.ico ..... ~ 1.59ms
2025-05-16 21:11:48 /login ..... ~ 2s
2025-05-16 21:11:51 /build/assets/app-DbTvgVAB.css ..... ~ 0.90ms
2025-05-16 21:11:51 /build/assets/app-Bf4POITK.js ..... ~ 0.53ms
2025-05-16 21:11:52 /favicon.ico ..... ~ 0.88ms
2025-05-16 21:11:52 /login ..... ~ 2s
2025-05-16 21:11:54 /login ..... ~ 0.47ms
2025-05-16 21:11:54 /build/assets/app-DbTvgVAB.css ..... ~ 0.73ms
2025-05-16 21:11:54 /build/assets/app-Bf4POITK.js ..... ~ 10.94ms
2025-05-16 21:11:55 /favicon.ico ..... ~ 0.86ms
2025-05-16 21:11:55 ..... ~ 498.37ms
2025-05-16 21:12:05 /logout ..... ~ 544.63ms
2025-05-16 21:12:05 / ..... ~ 501.32ms
2025-05-16 21:12:05 ..... ~ 509.44ms
```

### Step 2:

app/Http/Controllers/UserController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Models\User;
```

```
class UserController extends Controller
```

```
{
```

```
    public function index()
```

```
    {
```

```
        $users = User::all();
```

```
        return view('users.index', compact('users'));
```

```
    }
```

```

public function store(Request $request)
{
    User::create($request->validate([
        'name' => 'required',
        'email' => 'required|email',
        'password' => 'required|min:6',
    ]));
    return redirect()->back()->with('success', 'User created!');
}

public function update(Request $request, $id)
{
    $user = User::findOrFail($id);
    $user->update($request->only('name', 'email'));
    return redirect()->back()->with('success', 'User updated!');
}

public function destroy($id)
{
    User::destroy($id);
    return redirect()->back()->with('success', 'User deleted!');
}
}

```

### Step 3:

**routes/web.php**

**Add** these routes at the bottom of the file:

#### **CODE:**

```
use App\Http\Controllers\UserController;
```

```
Route::get('/users', [UserController::class, 'index']);
```

```
Route::post('/users', [UserController::class, 'store']);
```

```
Route::post('/users/update/{id}', [UserController::class, 'update']);
```

```
Route::post('/users/delete/{id}', [UserController::class, 'destroy']);
```

## Step 4:

`index.blade.php`

**Path:** `resources/views/users/index.blade.php`

### CODE:

```
<!DOCTYPE html>

<html>

<head>

    <title>Laravel CRUD - Users</title>

</head>

<body>

    <h1>Laravel CRUD: Users</h1>

    <h2>Add User</h2>

    <form method="POST" action="/users">

        @csrf

        Name: <input type="text" name="name" required><br>

        Email: <input type="email" name="email" required><br>

        Password: <input type="password" name="password" required><br>

        <button type="submit">Add User</button>

    </form>

    <h2>Update User</h2>

    <form method="POST" action="/users/update/1">

        @csrf

        Name: <input type="text" name="name" required><br>

        Email: <input type="email" name="email" required><br>

        <button type="submit">Update User with ID 1</button>

    </form>
```

```
<h2>Delete User</h2>
```

```
<form method="POST" action="/users/delete/1">
```

```
  @csrf
```

```
  <button type="submit">Delete User with ID 1</button>
```

```
</form>
```

```
<h2>All Users</h2>
```

```
<table border="1" cellpadding="5">
```

```
  <tr><th>ID</th><th>Name</th><th>Email</th></tr>
```

```
  @foreach ($users as $user)
```

```
    <tr>
```

```
      <td>{{ $user->id }}</td>
```

```
      <td>{{ $user->name }}</td>
```

```
      <td>{{ $user->email }}</td>
```

```
    </tr>
```

```
  @endforeach
```

```
</table>
```

```
</body>
```

```
</html>
```

Laravel CRUD - Users

WhatsApp | Secure and Reliab...

localhost:8000/users

GmailMapsImported From IEImported From IE (1)classroom.google.c...YouTube@Fitywap Fitywa...Watch Indian Movie...All bookmarks

## Laravel CRUD: Users

### Add User

Name:

mohsin

Email:

kpkawarsajjad@gmail.com

Password:

\*\*\*\*\*

Add User

### Update User

Name:

Email:

Update User with ID 1

### Delete User

Delete User with ID 1

### All Users

ID	Name	Email
1	Mohsin Sajjad	kpkawarsajjad@gmail.com
2	mohsin	mohsin@gmail.com