**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

**BJECTIVES 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 = "";

$database = "mohsinproject";

// Create Connection

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

//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>

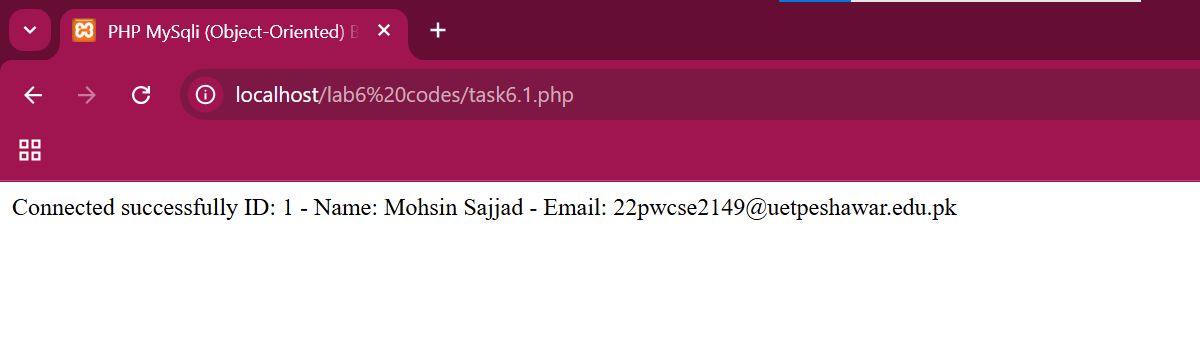
</html>

<?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 = "";

$database = "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();

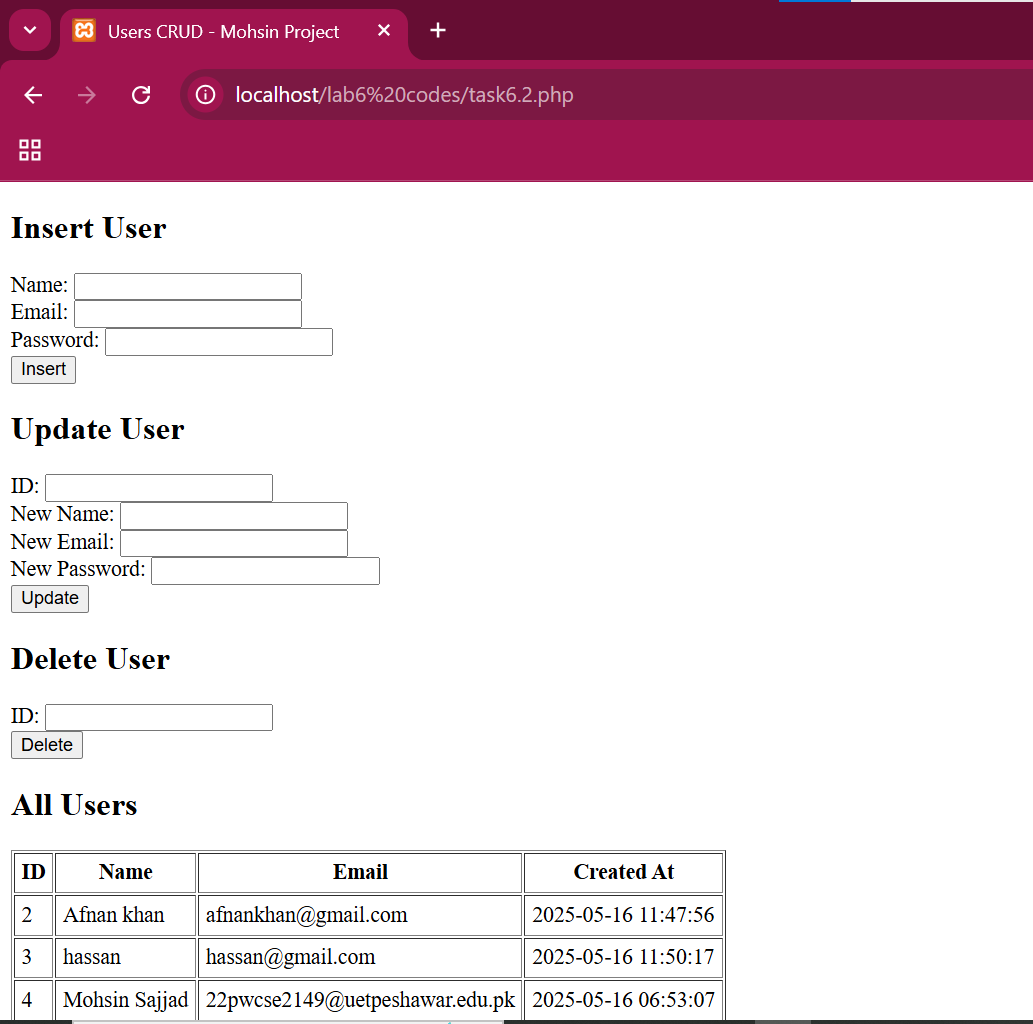
    $conn->close();

    ?>

</body>

</html>

**Output:**

****

**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']}</td>

                        <td>{$user['name']}</td>

                        <td>{$user['email']}</td>

                        <td>{$user['created\_at']}</td>

                      </tr>";

            }

        } else {

            echo "<tr><td colspan='4'>No users found</td></tr>";

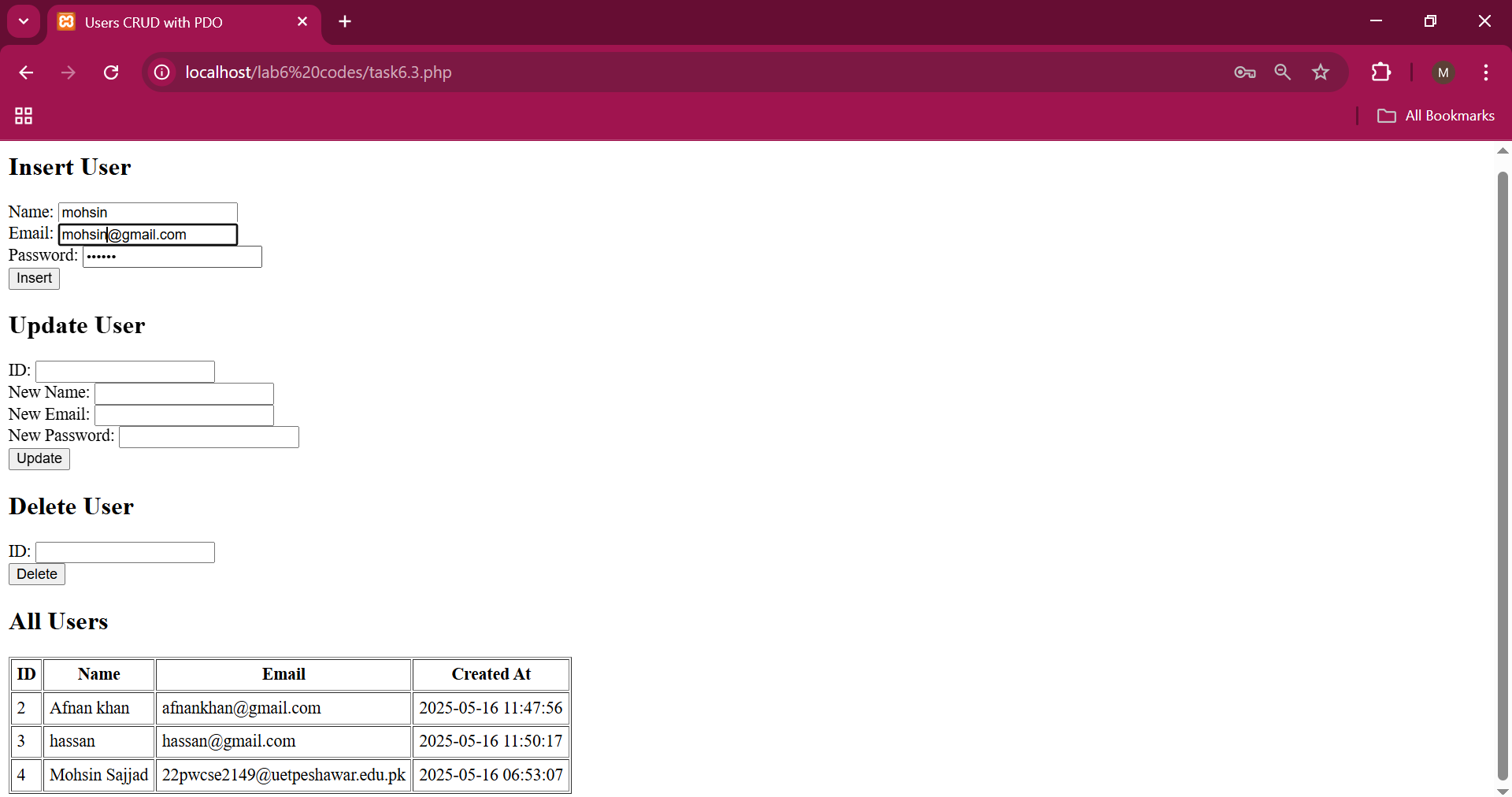
        } ?>

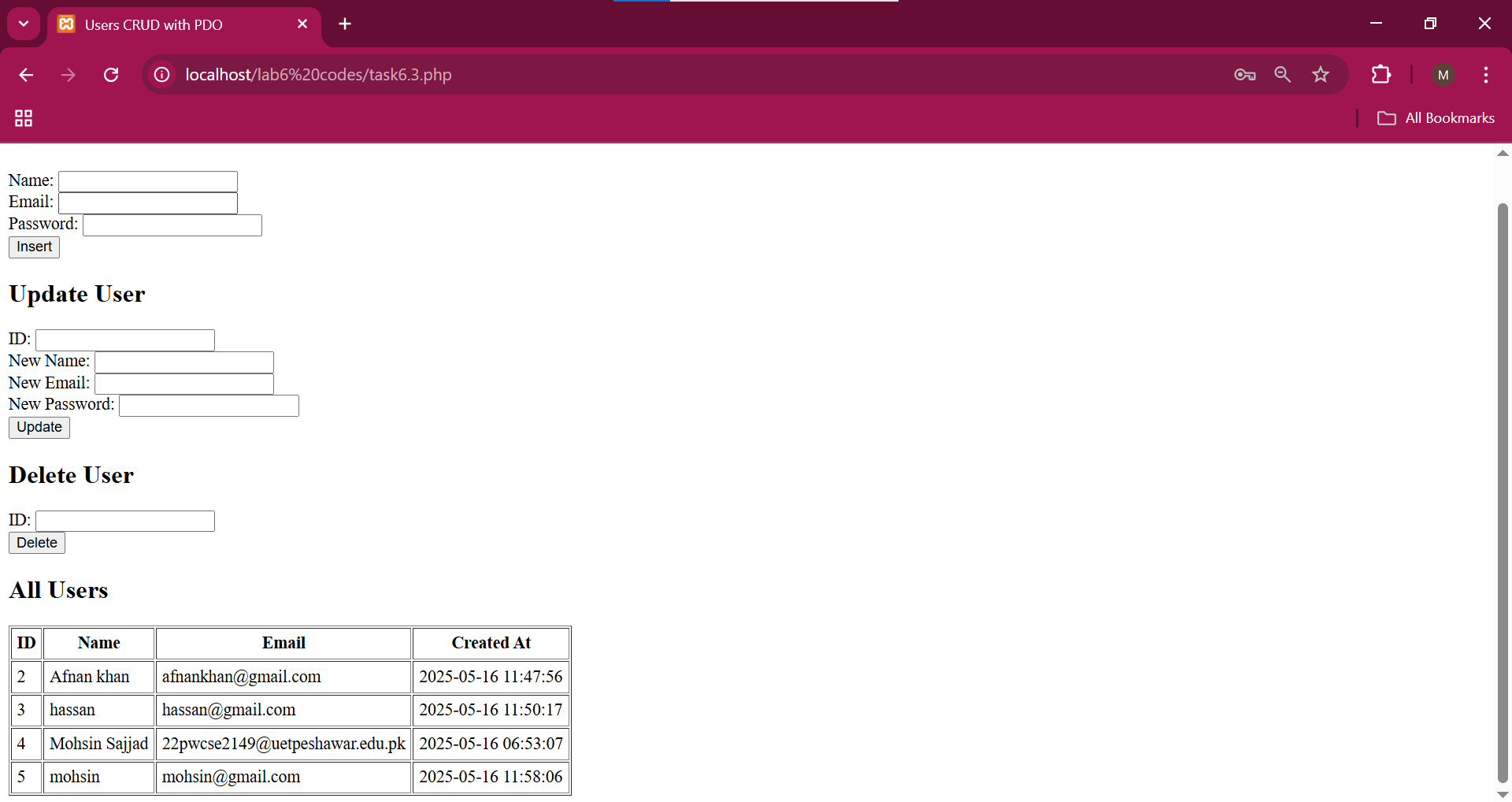
    </table>

</body>

</html>

**Output:**

****

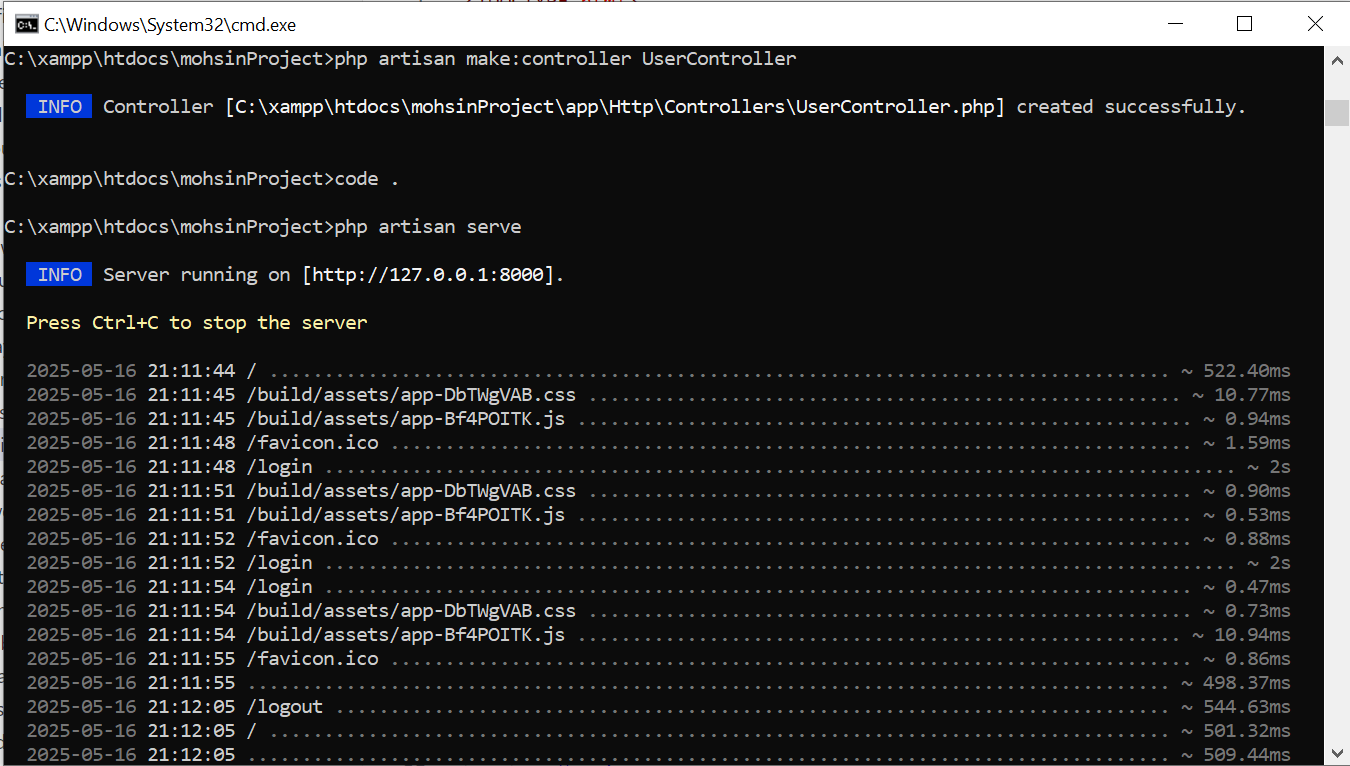
****

**Task 6.4**

Perform Task 6.2 using LARAVEL

**Step 1:**

**Code .**

****

**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>

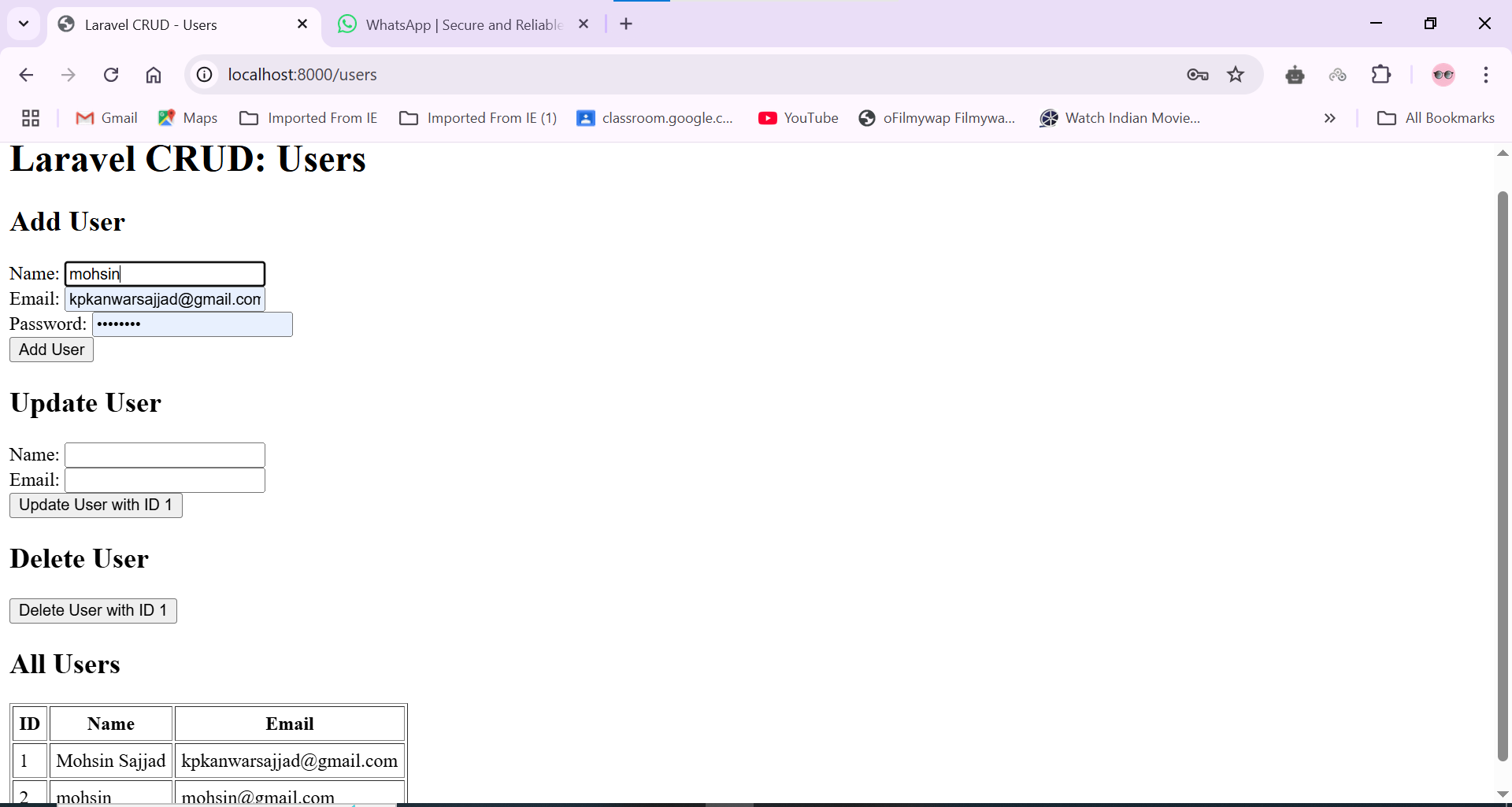
</tr>

@endforeach

</table>

</body>

</html>

****