**Hamza Mehmood  
Sp-21110  
Web Engeering  
Mam Nabeela**

**Assignment 03**

You are tasked with developing a secure sign-up and sign-in system for a social media platform using PHP and MySQL. The system needs to handle user registration, authentication, and session management. The platform requires users to create accounts with a unique username, valid email address, and a strong password. Once registered, users should be able to log in securely using their credentials. After successful authentication, users should have access to their personalized dashboard.

Tasks to be accomplished:

**signup.php:**

* Contains an HTML form for user registration, including fields for username, email, password, and file upload for the profile picture.
* Validates user input (both client-side and server-side) and checks the uploaded file's type and size.
* Sends validated data, including the uploaded profile picture, to **register.php** for processing.

**register.php:**

* Receives POST data from **signup.php.**
* Establishes a connection to the MySQL database (e.g., db\_connection.php).
* Performs server-side validation (e.g., checking for existing usernames or emails, validating password strength, etc.).
* Hashes the password securely (using PHP's password\_hash function) and stores user data in the database table, e.g., users\_table.

**login.php:**

* Contains the HTML form for user login.
* Validates user input and sends data to **authenticate.php.**

**authenticate.php:**

* Receives POST data from login.php.
* Validates user credentials against the data stored in the users\_table in the MySQL database.
* If authentication succeeds, starts a session using session\_start() and sets session variables (e.g., $\_SESSION['user\_id']) to maintain user login state.
* Redirects the authenticated user to their dashboard (**dashboard.php**) or displays an error message for unsuccessful login attempts.

**dashboard.php:**

* Contains the user's personalized dashboard.
* Checks for an active session using session\_start() and $\_SESSION variables to ensure that only authenticated users access the dashboard.
* Displays user-specific content or functionality based on the logged-in user's information.

**logout.php:**

* Destroys the active session using session\_start() and session\_destroy().
* Redirects the user to the login page (**login.php**) after logout.

**db\_connection.php:**

* Contains PHP code to establish a connection to the MySQL database.
* Includes credentials (e.g., hostname, username, password, database name) to connect to the MySQL database.

==============================================

Good Luck!

**db\_connection.php**

**<?php**

**$host = "your\_database\_host";**

**$username = "your\_database\_username";**

**$password = "your\_database\_password";**

**$database = "your\_database\_name";**

**// Create connection**

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

**// Check connectionif ($conn->connect\_error) {**

**die("Connection failed: " . $conn->connect\_error);**

**}**

**?>**

**signup.php**

<!DOCTYPE html>

<html>

<head>

<title>Sign Up</title>

</head>

<body>

<h2>Sign Up</h2>

<form action="register.php" method="post" enctype="multipart/form-data">

Username: <input type="text" name="username" required><br>

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

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

Profile Picture: <input type="file" name="profile\_picture" accept="image/\*" required><br>

<input type="submit" value="Sign Up">

</form>

</body>

</html>

**register.php:**

<?php

include 'db\_connection.php';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$username = $\_POST["username"];

$email = $\_POST["email"];

$password = $\_POST["password"];

// Additional server-side validation goes here

// Hash the password

$hashed\_password = password\_hash($password, PASSWORD\_DEFAULT);

// Check if the username or email already exists in the database

$check\_query = "SELECT \* FROM users\_table WHERE username='$username' OR email='$email'";

$result = $conn->query($check\_query);

if ($result->num\_rows == 0) {

// Insert user data into the database

$insert\_query = "INSERT INTO users\_table (username, email, password) VALUES ('$username', '$email', '$hashed\_password')";

$conn->query($insert\_query);

// Redirect to login page or any other page

header("Location: login.php");

exit();

} else {

echo "Username or email already exists.";

}

}

$conn->close();

?>

**login.php:**

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

</head>

<body>

<h2>Login</h2>

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

Username: <input type="text" name="username" required><br>

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

<input type="submit" value="Login">

</form>

</body>

</html>

**authenticate.php:**

<?php

session\_start();

include 'db\_connection.php';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$username = $\_POST["username"];

$password = $\_POST["password"];

// Retrieve user data from the database

$query = "SELECT \* FROM users\_table WHERE username='$username'";

$result = $conn->query($query);

if ($result->num\_rows == 1) {

$row = $result->fetch\_assoc();

// Verify password

if (password\_verify($password, $row["password"])) {

// Set session variables

$\_SESSION["user\_id"] = $row["id"];

// Redirect to dashboard or any other page

header("Location: dashboard.php");

exit();

} else {

echo "Invalid password.";

}

} else {

echo "Invalid username.";

}

}

$conn->close();

?>

**dashboard.php:**

<?php

session\_start();

// Check if user is not logged in, redirect to login page

if (!isset($\_SESSION["user\_id"])) {

header("Location: login.php");

exit();

}

// Additional dashboard content goes here

echo "Welcome to your dashboard!";

?>

**logout.php:**

<?php

session\_start();

session\_destroy();

// Redirect to login page after logout

header("Location: login.php");

exit();

?>

**Table:**

**CREATE TABLE users\_table (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**username VARCHAR(50) UNIQUE NOT NULL,**

**email VARCHAR(100) UNIQUE NOT NULL,**

**password VARCHAR(255) NOT NULL**

**);**