**User Authentication System in PHP**

**Introduction**

In this assignment, we will explore a basic **user authentication system** using PHP and MySQL. This system includes functionalities such as **user registration (signup), login, session management, and logout**. The purpose of this assignment is to understand how web applications handle user authentication securely.

**1. Database Connection (config.php)**

**Purpose:**

This file is responsible for establishing a connection to the **MySQL database** so that the application can interact with stored user data.

**Key Features:**

* Uses mysqli\_connect() to establish a connection.
* Checks for connection errors and terminates if there is a failure.

**Code Explanation:**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "user\_db";

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

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

?>

✅ **This ensures that PHP can communicate with the database.**

**2. User Signup (signup.php)**

**Purpose:**

Allows new users to create an account by entering their **name, email, and password**. The password is **hashed** before storing it in the database for security.

**Code Explanation:**

<?php

include 'config.php';

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

$name = $\_POST['name'];

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

$password = password\_hash($\_POST['password'], PASSWORD\_DEFAULT); // Encrypt password

$sql = "INSERT INTO users (name, email, password) VALUES ('$name', '$email', '$password')";

if ($conn->query($sql) === TRUE) {

echo "Registration successful. <a href='login.php'>Login here</a>";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

?>

✅ **Key Features:**

* Accepts user input and hashes the password using password\_hash().
* Stores user details securely in the database.
* Redirects users to login after successful registration.

**3. User Login (login.php)**

**Purpose:**

Verifies user credentials and starts a **session** if authentication is successful.

**Code Explanation:**

<?php

session\_start();

include("config.php");

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

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

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

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

$result = mysqli\_query($conn, $query);

$user = mysqli\_fetch\_assoc($result);

if ($user && password\_verify($password, $user['password'])) {

$\_SESSION['user\_id'] = $user['id'];

$\_SESSION['email'] = $user['email'];

header("Location: dashboard.php");

exit();

} else {

echo "Invalid email or password!";

}

}

?>

✅ **Key Features:**

* Checks email and password against the database.
* Uses password\_verify() to compare hashed passwords.
* Starts a session and redirects to dashboard.php if login is successful.

**4. User Dashboard (dashboard.php)**

**Purpose:**

Displays a protected page that only **logged-in users** can access.

**Code Explanation:**

<?php

session\_start();

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

header("Location: login.php");

exit();

}

echo "Welcome, " . $\_SESSION['email'] . "!";

echo "<br><a href='logout.php'>Logout</a>";

?>

✅ **Key Features:**

* Prevents unauthorized access using session\_start().
* Displays logged-in user's email.
* Provides a **logout** option.

**5. User Logout (logout.php)**

**Purpose:**

Ends the user's session and redirects them to the login page.

**Code Explanation:**

<?php

session\_start();

session\_destroy();

header("Location: login.php");

exit();

?>

✅ **Key Features:**

* Destroys session data and logs out the user.
* Redirects users back to the **login page**.

**Conclusion**

This assignment covered a simple **user authentication system** using PHP and MySQL. The system securely stores user passwords, manages sessions, and restricts access to certain pages. This approach can be expanded to include features like **email verification, password reset, and user roles**.

✅ **Key Takeaways:**

* Securely storing passwords using password\_hash() and password\_verify().
* Preventing unauthorized access using **sessions**.
* Managing user authentication with **database queries**.