|  |  |
| --- | --- |
| 实验名称： | 实验4：Session 和数据库 |
| 班级： | 信息安全224 |
| 学号： | 2022132003 |

1. **实验过程：**

（实验的具体过程，可以分成两个部分，第一个部分写思路，第二个部分写具体的实现代码。）

1. 实验思路：

(1) 编写登录页面（index.php）

步骤：

创建HTML表单：使用HTML编写一个包含用户名和密码输入字段的表单，并设置表单的action属性为处理登录的PHP文件。

表单提交：表单使用POST方法提交数据。

验证用户输入：在login\_process.php中，获取用户输入的用户名和密码，并与预设或数据库中的值进行比较。

登录验证：如果用户名和密码正确，则设置会话变量（如$\_SESSION['username']）并重定向到welcome.php页面；否则，显示错误消息并允许用户重新输入。

(2) 编写welcome页面

步骤：

检查登录状态：在welcome.php页面的开始，检查会话变量$\_SESSION['username']是否存在。

显示内容：如果已登录，显示用户的基本信息；否则，重定向到index.php页面。

(3) 设计MySQL用户表并完成注册页面

步骤：

设计用户表：在MySQL数据库中创建一个用户表，包含字段如id、username、password（使用哈希加密）等。

创建注册表单：编写一个HTML表单，用于收集用户注册信息。

处理注册请求：在PHP文件中接收表单数据，检查用户名是否已存在，如果不存在，则将用户信息（包括加密后的密码）插入到用户表中，并重定向到welcome.php页面。

(4) 实现个人信息修改页面

步骤：

检查登录状态：在个人信息修改页面开始时，检查用户是否已登录。

创建表单：创建一个HTML表单，用于收集用户修改后的信息，包括密码（需要输入两次以确认）。

处理修改请求：在PHP文件中接收表单数据，验证新密码的两次输入是否一致，然后更新数据库中的用户信息。

重定向：修改成功后，重定向到welcome.php页面。

(5) 使用session和include\_once进行用户验证和页面保护

步骤：

初始化session：在所有需要验证用户登录状态的页面开始时，使用session\_start()函数初始化会话。

创建session验证文件：创建一个PHP文件（如session\_check.php），用于检查用户是否已登录。使用include\_once语句在其他页面中引用此文件。

页面保护：在需要登录才能访问的页面开始处，包含session\_check.php文件。如果用户未登录，则重定向到index.php页面。

通过这些步骤，您可以实现一个基本的用户登录、注册、个人信息修改和页面保护的功能。

1. 代码实现：

重要代码片段说明：

index.php

<?php

session\_start();

include\_once 'db\_config.php';

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

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

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

$sql = "SELECT \* FROM users WHERE username = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param('s', $username);

$stmt->execute();

$result = $stmt->get\_result();

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

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

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

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

header('Location: welcome.php');

exit();

} else {

$error = "Invalid username or password.";

}

} else {

$error = "Invalid username or password.";

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Login</title>

</head>

<body>

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

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<br>

<button type="submit">Login</button>

</form>

<?php if (isset($error)) echo "<p>$error</p>"; ?>

</body>

</html>

welcome.php

<?php

session\_start();

include\_once 'db\_config.php';

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

header('Location: index.php');

exit();

}

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

$sql = "SELECT \* FROM users WHERE id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param('i', $user\_id);

$stmt->execute();

$user = $stmt->get\_result()->fetch\_assoc();

?>

re.php

**<?php**

**session\_start();**

**include\_once 'db\_config.php';**

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

**$username = $\_POST['username'];**

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

**$confirm\_password = $\_POST['confirm\_password'];**

**if ($password !== $confirm\_password) {**

**$error = "Passwords do not match.";**

**} else {**

**$sql = "SELECT \* FROM users WHERE username = ?";**

**$stmt = $conn->prepare($sql);**

**$stmt->bind\_param('s', $username);**

**$stmt->execute();**

**$result = $stmt->get\_result();**

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

**$error = "Username is already taken.";**

**} else {**

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

**$sql = "INSERT INTO users (username, password) VALUES (?, ?)";**

**$stmt = $conn->prepare($sql);**

**$stmt->bind\_param('ss', $username, $hashed\_password);**

**if ($stmt->execute()) {**

**$\_SESSION['user\_id'] = $conn->insert\_id;**

**header('Location: welcome.php');**

**exit();**

**} else {**

**$error = "Registration failed.";**

**}**

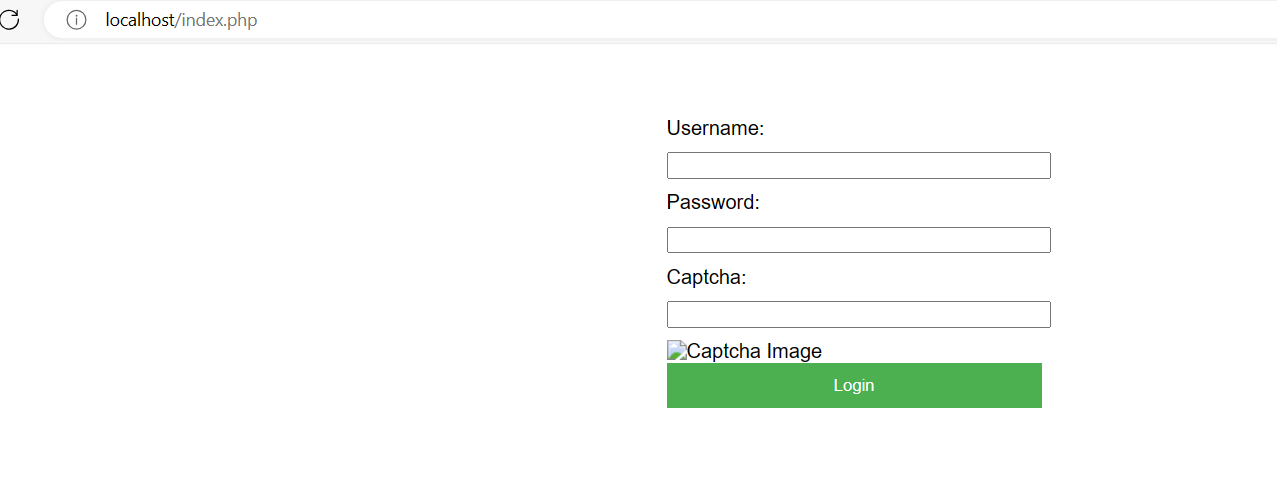
**}**

**}**

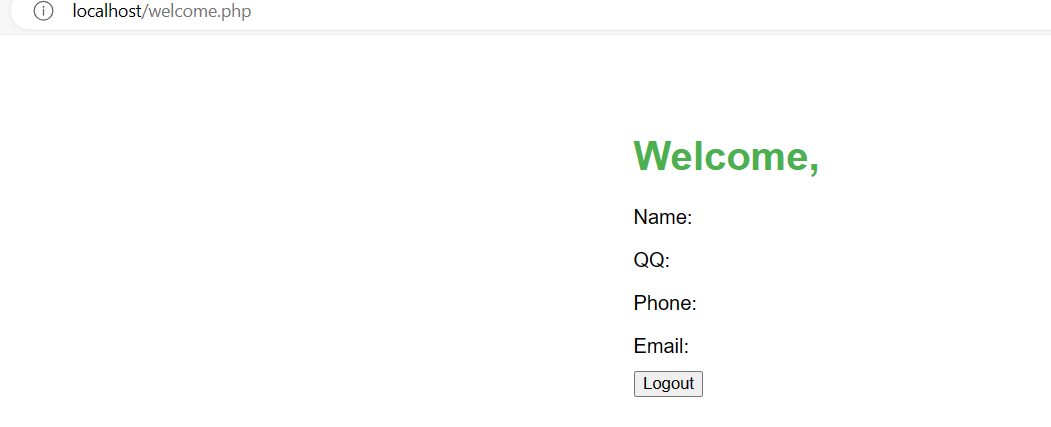
**}**

**?>**

1. **实验结果**
2. 编写登录页面（index.php），实现用户名和密码的登录：

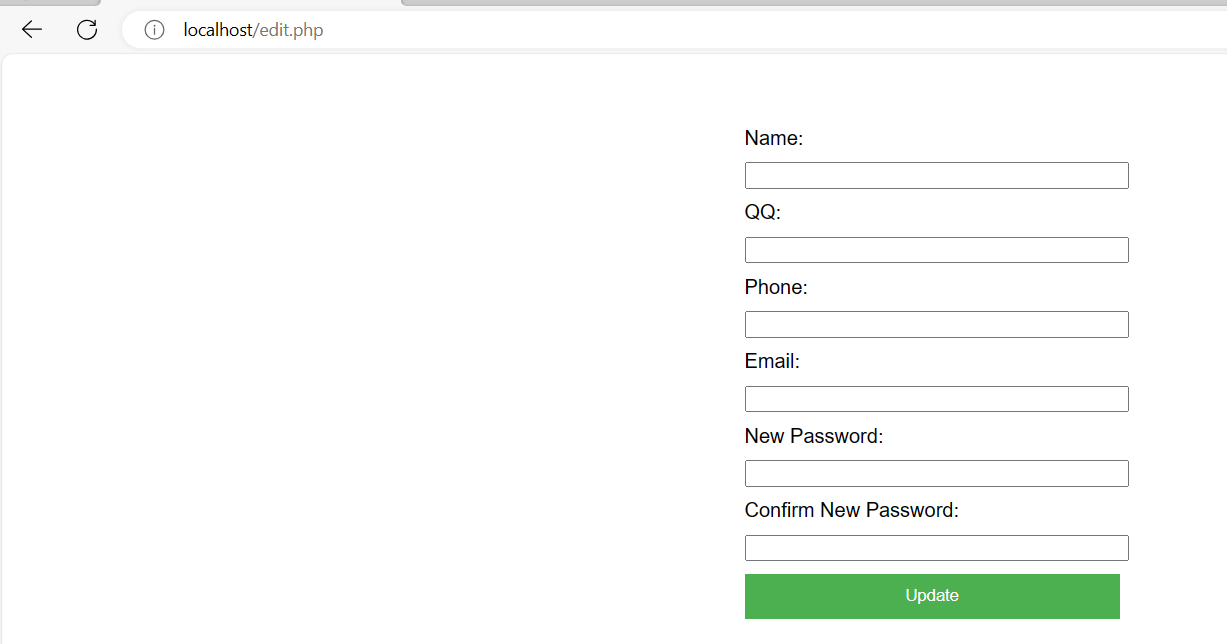


1. 编写一个简单的 welcome 页面，用户成功登录后进入该页面。该页面可以是用户基本信息的显示页面：



3. 在 MySQL 中设计一个用户表（参考步骤 4 中的个人信息），完成一个用户注册页面。可以判断用户名是否被占用，如果被占用进行提示，如果没有被占用， 注册成功后重定向到 welcome 页面进行显示:

4.实现一个页面，用于登录用户修改自己的个人信息，个人信息包括登录名、姓名、QQ 号、电话、邮箱、头像、密码等（照片可以不用设置）。修改页面如下：



1. **总结与思考**

这次实验的目的是深入理解Session在Web应用中的作用以及与数据库之间的交互。在实验过程中，我尝试将Session与数据库进行结合，实现用户数据的存储与检索。通过Session，我可以轻松地跟踪用户的活动，并根据需要将数据保存到数据库中。同时，我也可以从数据库中检索用户的数据，并在用户访问不同页面时，通过Session来恢复这些数据，从而为用户提供个性化的服务。

总的来说，这次实验让我对Session和数据库有了更深入的了解，并提升了我的实践能力和解决问题的能力。我相信，在未来的学习和工作中，这些经验和知识将会对我产生积极的影响。