



دانشکده مهندسی کامپیوتر

دانشگاه اصفهان

پروژه نهایی مبانی امنیت سایبری

استاد: سیدحسین تهامی

مهروالسادات نوحی

۹۹۳۶۱۳۰۶۱

پاییز ۱۴۰۲

پروژه پایان ترم

پیاده سازی یک سیستم لاگین به صورت کلاینت سرور مبتنی بر وب، با استفاده از پروتکل لمپورت و زنجیره چکیده‌ها

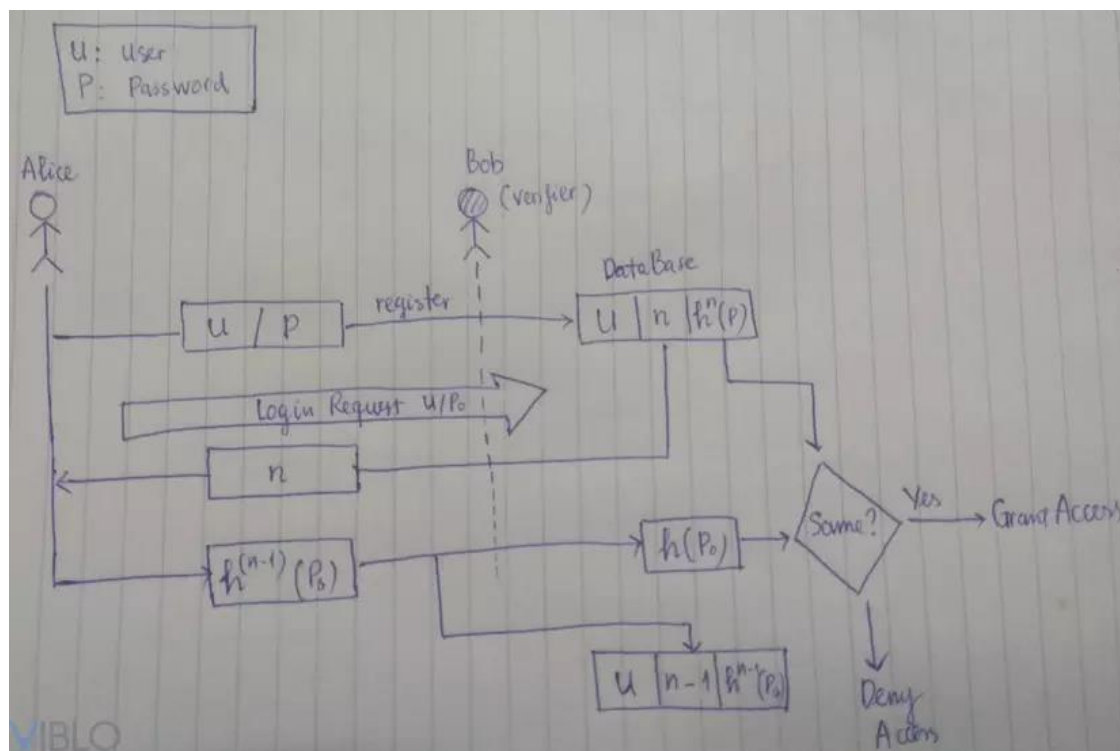
در گام اول پروتکل لمپورت (تأیید هویت رمز عبور ایمن یکبار مصرف) را تعریف می‌کنیم.

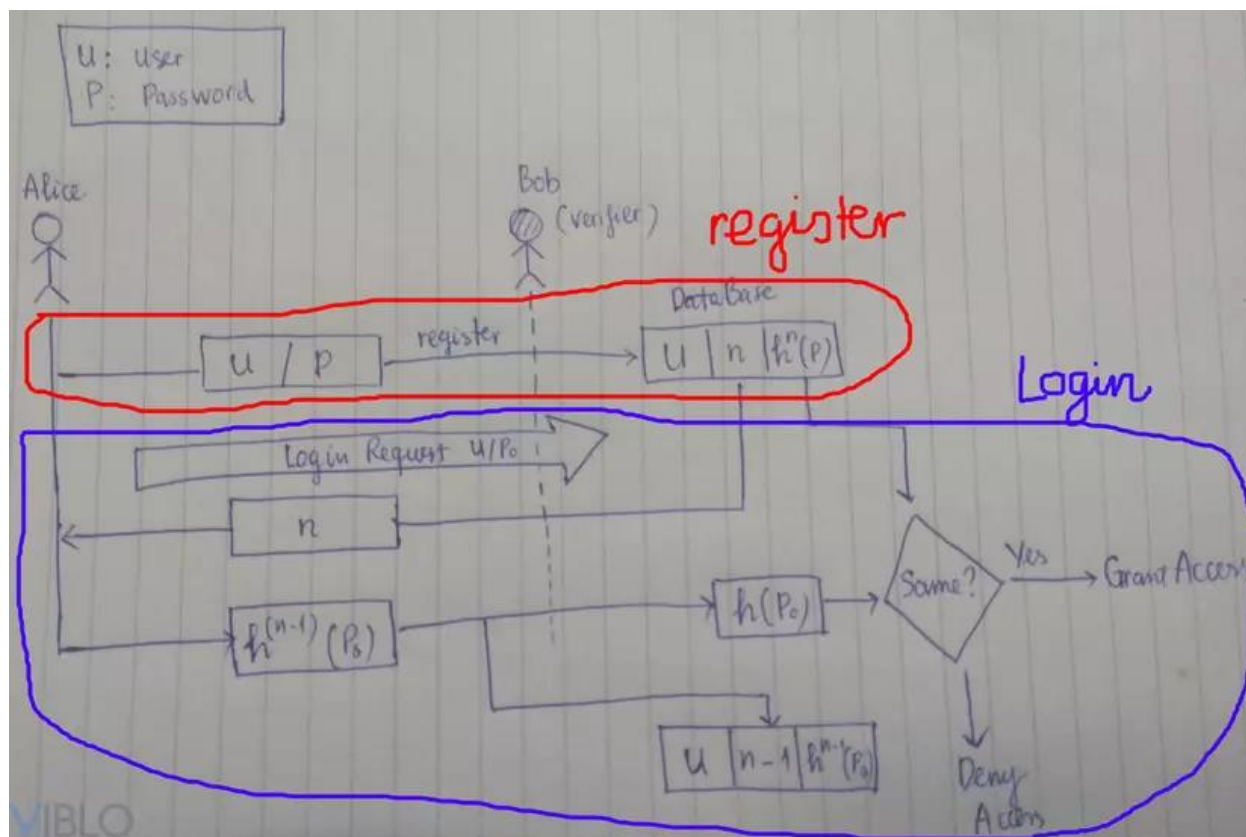
ایده Lamport این است که از رشته‌های مقدار هش متوالی استفاده کند، هر مقدار در این رشته به عنوان رمز عبور استفاده می‌شود و از آخرین عنصر شروع می‌شود (متولد آخرین در رشته هش). بنابراین چالش و پاسخ در هش رمز عبور ارائه شده توسط کاربر در ورود به سیستم $n + 1$ نهفته است، سیستم باید رمز عبور استفاده شده را در n امین بار (بلافاصله قبل) دریافت کند.

مکانیسم روش لمپورت به ۲ مرحله تقسیم می‌شود:

- مرحله ثبت نام
- احراز هویت فاز

ثبت نام فقط یک بار انجام می‌شود، هر بار که کاربر وارد سیستم می‌شود، احراز هویت انجام می‌شود.





فاز ثبت نام:

در تصویر بالا مرحله ثبت نام به شرح زیر انجام می شود:

۱. کاربر (آلیس) نام کاربری (U) و رمز عبور (P) را برای ثبت نام وارد می کند.
۲. سرور (Bob) داده ها را در پایگاه داده وارد می کند، جایی که user_name به صورت ساده plaintext ذخیره می شود.
۳. سرور یک مقدار ثابت n تصادفی ذخیره کنار نام کاربری و رمز در پایگاه داده ذخیره می کند.
۴. سرور از رمز عبور (P)، n بار هش گرفته و در این پایگاه داده ذخیره می کند. به عبارتی $(h^n(P))$ در قسمت رمز عبور ذخیره خواهد شد.

بنابراین مرحله ثبت نام به پایان رسید.

در گام بعد فاز ورود به سیستم و احراز هویت خواهد بود.

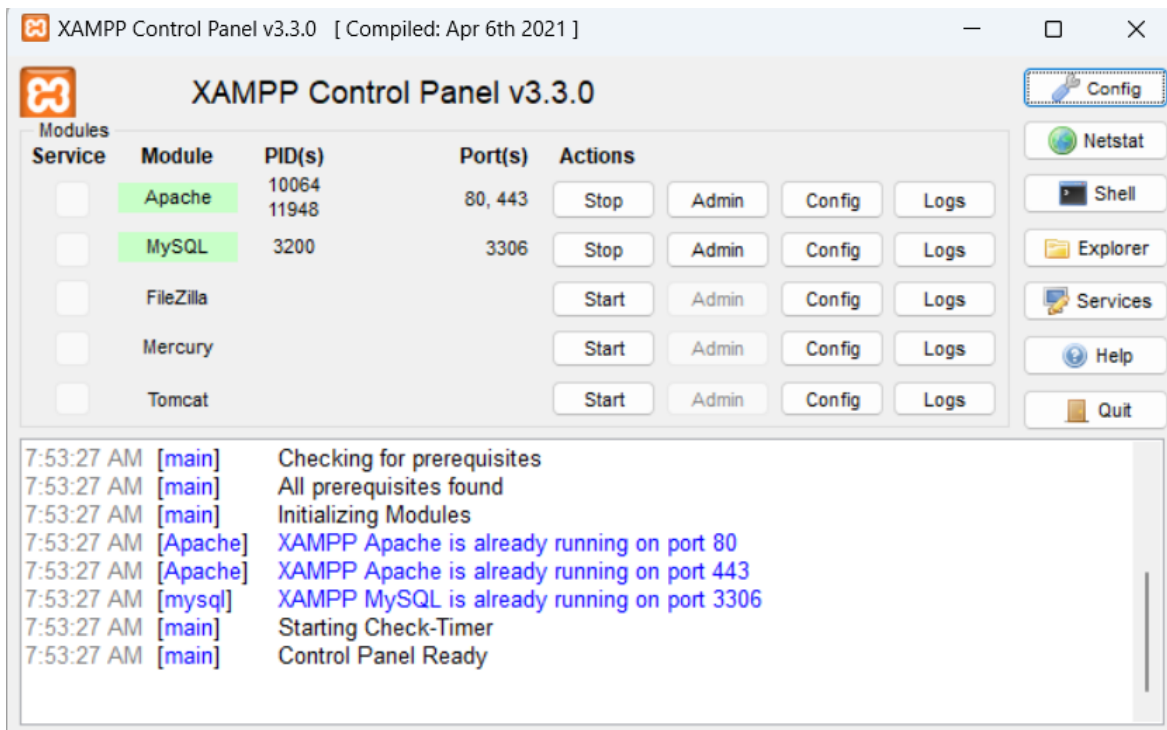
فاز ورود به سیستم:

۱. آلیس (U) `user_name` را به سرور می فرستد.
۲. باب یا سرور پایگاه داده را بررسی می کند تا ببیند `user_name` وجود دارد یا خیر، اگر وجود دارد، سپس n را بر اساس آن برمی گرداند.
۳. در این زمان، فرمی برای آلیس ظاهر می شود تا رمز عبور را وارد کند (P_0)، به این شکل که یک بار $n - 1$ هش می شود و سمت باب فرستاده می شود و n مقدار Bob است که به تازگی برگردانده شده است.
۴. وقتی باب کد هش پسورد را دریافت کرد و دوباره از آن هش می گیرد، $h^n(P_0)$ خواهد داشت.
۵. باب $h^n(P_0)$ را با $h^n(P)$ ذخیره شده در پایگاه داده مقایسه می کند، اگر این دو مقدار مطابقت داشته باشند، با موفقیت وارد شوید و ۲ مقدار $h^{(n-1)}(P_0)$ و $n-1$ را به پایگاه داده، بنابراین رمز عبور ذخیره شده در پایگاه داده به روز می شود.
- همانطور که پس از هر بار ورود، مقدار هش رمز عبور در پایگاه داده تغییر می کند به طوری که در صورت سرقت پایگاه داده، دریافت رمز عبور کاربر بسیار دشوار است.

پیاده سازی:

برای اینکه ما سروری داشته باشیم از `localhost` یا `۱۲۷.۰.۰.۱` سیستم خود استفاده می کنیم. در گام بعد ما نیازمند یک وب سرور هستیم. از آنجایی که زبانی که من استفاده کردم `php` بود از `xampp` استفاده شده که از وب سرور `Apache` و `MySQL` و همچنین یک مفسر برای کامپایل زبان `PHP` می باشد پشتیبانی می کرد که پنل این نرم افزار به صورت زیر است:

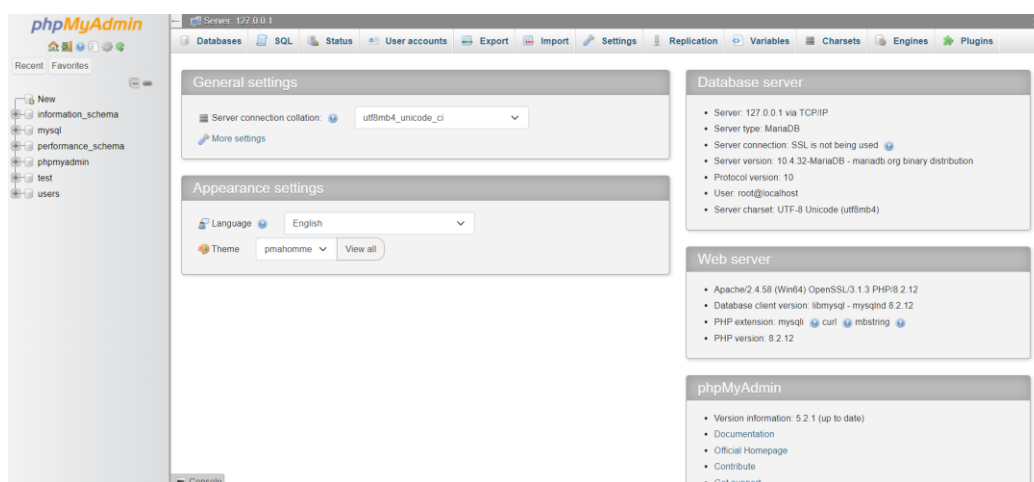
برای کاری که ما نیاز داشتیم سرویس های اول و دوم هست که هنگامی که `start` را می زنیم وقتی رنگ دو این سرویس به سبز درآید یعنی سرویس فعال و آماده پاسخگویی خواهد بود. پورت های موردنظر نیز مشخص هستند.



در اینجا هر دو بخش را توضیح خواهیم داد:

Apache: تمامی مسیرهای پروژه ما و علاوه بر بخش اجرای کدهای php ها مورد نظر روی این اتفاق میفتد و همانند وب سروری که روی سرور اجرا می شود کدهای ما رو اجرا خواهد کرد.

MySQL: برای دسترسی و کنترل پایگاه مدنظر می باشد. و phpMyAdmin را برای بالا می آورد.



در هنگام نصب xampp روی درایو C به صورت پیش فرض خواهد بود. و درگام بعد پوشه htdocs است.

> This PC > Local Disk (C:) >				
Name	Date modified	Type	Size	
Intel	7/11/2023 2:21 PM	File folder		
ja-netfilter	11/15/2023 12:51 AM	File folder		
PerfLogs	5/6/2022 9:24 PM	File folder		
Program Files	1/11/2024 11:46 PM	File folder		
Program Files (x86)	1/11/2024 11:20 PM	File folder		
Users	7/11/2023 10:05 AM	File folder		
Windows	1/12/2024 1:46 AM	File folder		
xampp	1/21/2024 10:25 PM	File folder		

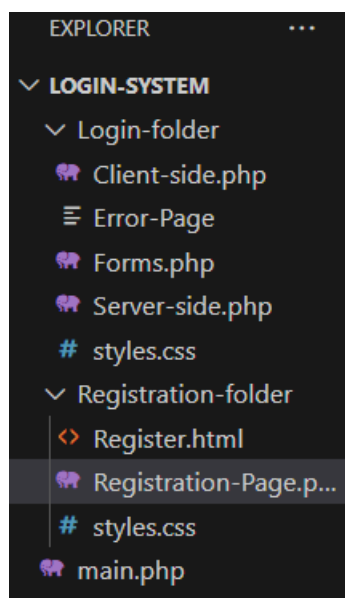
anonymous	1/21/2024 10:22 PM	File folder
apache	1/21/2024 10:23 PM	File folder
cgi-bin	1/21/2024 10:25 PM	File folder
contrib	1/21/2024 10:22 PM	File folder
FileZillaFTP	1/21/2024 10:25 PM	File folder
htdocs	1/21/2024 10:32 PM	File folder
img	1/21/2024 10:22 PM	File folder
install	1/21/2024 10:25 PM	File folder
licenses	1/21/2024 10:22 PM	File folder
locale	1/21/2024 10:22 PM	File folder
mailoutput	1/21/2024 10:22 PM	File folder
mailtodisk	1/21/2024 10:23 PM	File folder
MercuryMail	1/21/2024 10:25 PM	File folder
mysql	1/21/2024 10:23 PM	File folder
perl	1/21/2024 10:23 PM	File folder
php	1/21/2024 10:25 PM	File folder
phpMyAdmin	1/21/2024 10:28 PM	File folder
sendmail	1/21/2024 10:25 PM	File folder
src	1/21/2024 10:22 PM	File folder

در پوشه htdocs کدهای ما اجرا خواهند شد. در این پوشه ما سیستم لاگین خود را باید پیاده سازی کنیم.

Name	Date Modified	Type	Size
dashboard	1/21/2024 10:22 PM	File folder	
img	1/21/2024 10:22 PM	File folder	
Login-System	1/21/2024 10:37 PM	File folder	
webalizer	1/21/2024 10:22 PM	File folder	
xampp	1/21/2024 10:22 PM	File folder	
applications	6/15/2022 8:07 AM	Chrome HTML Do...	4 KB
bitnami.css	6/15/2022 8:07 AM	CSSfile	1 KB
favicon	7/16/2015 7:32 AM	ICO File	31 KB
index	7/16/2015 7:32 AM	PHP Source File	1 KB

طبق توضیحاتی که داده شده محتوای Login-System به صورت زیر خواهد بود.

Login-folder	1/22/2024 5:17 AM	File folder	
Registration-folder	1/21/2024 11:17 PM	File folder	
main	1/22/2024 6:01 AM	PHP Source File	2 KB



محتوای Registration-Folder شامل صفحات فرم‌های Register.html, styles.css است و فایل پردازش اطلاعات Registration.php که هر کدام قرار داده خواهد شد.

```
EXPLORER
  LOGIN-SYSTEM
    Login-folder
      Client-side.php
      Error-Page
      Forms.php
      Server-side.php
      styles.css
    Registration-folder
      Register.html
      styles.css
      main.php

Register.html X
  Registration-folder > Register.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Registration Form</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10   <form action="Registration-Page.php" method="POST">
11     <div id="align">
12       <div id="container">
13         <p class="head">Registration</p>
14         <div class="input-container">
15           <label for="username">Username</label>
16           <input type="text" name="username" class="inpt" required>
17           <label for="password">Password</label>
18           <input type="password" name="password" class="inpt" required>
19           <div class="rem-forgot">
20             </div>
21           <button type="submit" value="submit" class="btn">Register</button>
22         </div>
23       </div>
24     </div>
25   </form>
26 </body>
```

```
EXPLORER
  LOGIN-...
    Login-folder
      Client-side.php
      Error-Page
      Forms.php
      Server-side.php
      styles.css
    Registration-folder
      Register.html
      styles.css
      main.php

# styles.css X
  Registration-folder > # styles.css > #input-container_btn
1 @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');
2 body{
3   font-family: 'Poppins', sans-serif;
4   background-image: url(https://naseemul1.github.io/img_host/peakpx.jpg);
5   background-size: cover;
6   background-position: center center;
7   background-attachment: fixed;
8   margin: 0;
9   padding: 0;
10  color: white;
11 }
12
13 #align{
14   height: 100vh;
15   display: flex;
16   justify-content: center;
17   align-items: center;
18 }
19
20 #container{
21   border: 2px solid white;
22   border-radius: 10px;
23   width: 320px;
24   height: 340px;
25   display: flex;
26   flex-direction: column;
```



```

1 <?php
2 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
3     $username = $_POST['username'];
4     $password = $_POST['password'];
5     $random_number = 3;
6     $hashed_password = hash('sha256', $password);
7     for ($i = 0; $i < $random_number-1; $i++) {
8         $hashed_password = hash('sha256', $hashed_password);
9     }

```

در فایل پردازش اطلاعات از طریق متد POST اطلاعات توسط کاربر گرفته شده و سرور یک یک مقدار تصادفی ثابت کنار اطلاعات کاربر در دیتابیس ذخیره میکند. همچنین هنگام ذخیره در ثبت نام سرور به اندازه تعداد عدد تصادفی از پسورد هاش گرفته و آن را ذخیره میکند.

```

$conn = mysqli_connect($db_host, $db_user, $db_password, $db_name);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// Prepare and execute a SQL query to insert data into the users table
$query = "INSERT INTO userstable (username, password, random_number) VALUES ('$username', '$hashed_

if ($conn->query($query) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $query . "<br>" . $conn->error;
}

// Redirect to login.php
header('Location: http://localhost/login-system/Login-folder/Forms.php');
exit();
$conn->close();
}
?>

```

در تصویر بالا نحوه اتصال به دیتابیس از طریق php آورده شده است. که ترکیبی از php و SQL کوئری میباشد. برای تست یک کاربر را وارد پایگاه داده میکنیم. در ابتدا دیتابیس خالی میباشد.

☐ Profiling
 [\[Edit inline \]](#)
[\[Edit \]](#)
[\[Explain SQL \]](#)
[\[Create PH](#)





id	username	password	random_number

هنگام ورود به صفحه ثبت نام باید از url مربوطه و تنظیم شده در htdocs استفاده کنیم. مسیر زیر معادل localhost میباشد.

C:\xampp\htdocs\Login-System

→ ↻ localhost/login-system/

Index of /login-system

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 Login-folder/	2024-01-22 05:17	-	
 Registration-folder/	2024-01-21 23:17	-	
 main.php	2024-01-22 06:01	1.3K	

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80

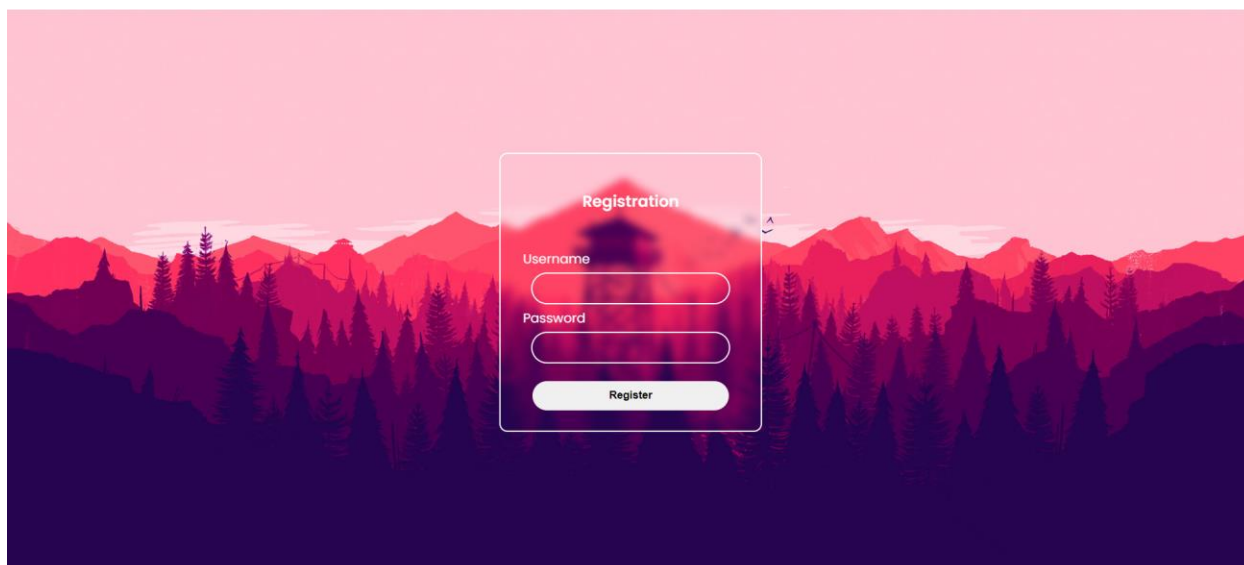
وارد بخش ثبت نام میشویم.

Index of /login-system/Registration-folder

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 Register.html	2024-01-22 00:33	1.0K	
 Registration-Page.php	2024-01-22 08:10	1.1K	
 styles.css	2024-01-22 00:33	1.8K	

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80

و وارد ثبت نام میشویم.



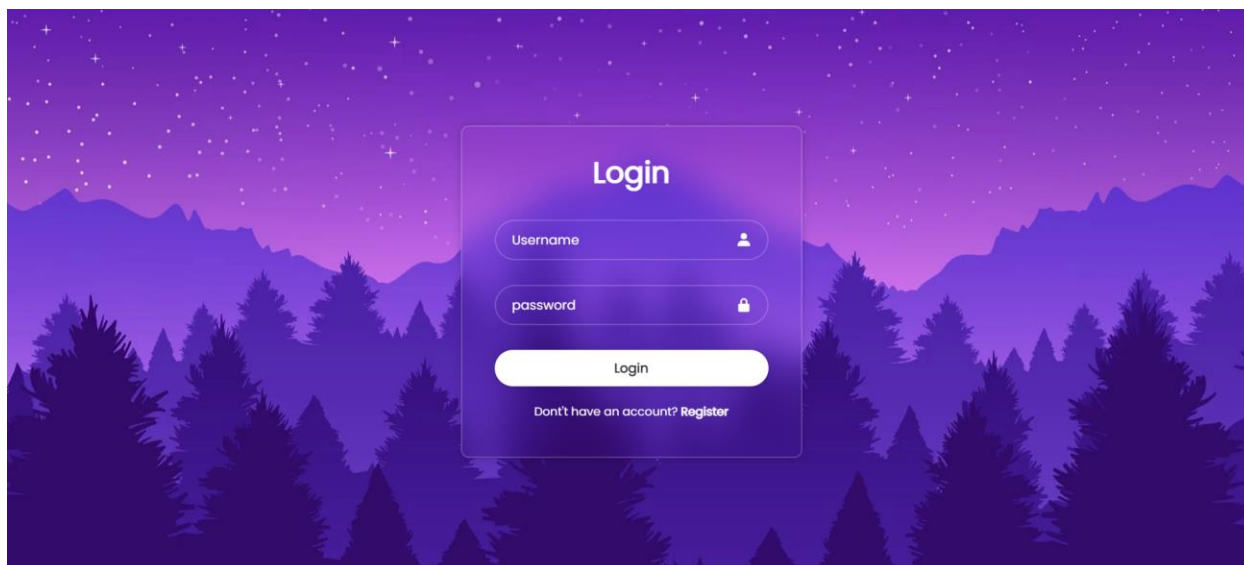
به عنوان نمونه یوزرنیم را mahroonoohi و پسورد را ۱ در نظر میگیریم. وقتی دکمه سابمیت میزنیم دیتایس ایدیت میشودو ثبت نام کاربر با موفقیت انجام شد.

id	username	password	random_number
4	mahroonoohi	d6a804981ea7ce374acc21c9a8bf82f50b684b0ea4bdf8b26a...	3

به اندازه ۳ بار از پسورد هاش گرفته شده است و ثبت شده است.

id	username	password	random_number
4	mahroonoohi	d6a804981ea7ce374acc21c9a8bf82f50b684b0ea4bdf8b26a...	3

در گام بعد میرویم سراغ لاگین و احراز اصالت اصلی انجام خواهد بود. در ادامه صفحه ورود آورده شده است.



در واقع کدهای اجرایی این به صورت زیر است.

```

10  <form action="Client-side.php" method="POST">
11  <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet' href='styles.c
12  <body>
13  <div class="wrapper">
14  <form action="">
15  <h1>Login</h1>
16  <div class="input-box">
17  <input type="text" name="username" placeholder="Username" required>
18  <i class='bx bxs-user'></i>
19  </div>
20  <div class="input-box">
21  <input type="password" name="password" placeholder="password" required>
22  <i class='bx bxs-lock-alt'></i>
23  </div>
24  <button type="submit" class="btn">Login</button>
25  <div class="register-link">
26  <p>Don't have an account? <a href="http://localhost/login-system/Registration-folder/f
27  Register
28  </a></p>
29  </div>
30  </form>
31  </div>
32  </body>
33  </form>
34  </body>
35  </html>

```

کاری که این صفحه میکند این است که اطلاعات کاربر گرفته و برای سرور ارسال میکند و به عنوان واسط صرفاً عمل میکند.

در گام بعد اطلاعات به صفحه Client-Side.php برای پردازش فرستاده میشود. در اینجا مشخصات گرفته شد و باید اطلاعات همزمان از پایگاه داده خوانده شود. تا چک کند آیا همچنین کاربری اصلاً وجود دارد یا خیر .

Login-folder > Client-side.php

```
1 <?php
2 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
3     $username = $_POST['username'];
4     $password = $_POST['password'];
5     $random_number=0;
6
7
8     $db_host = 'localhost';
9     $db_user = 'root';
10    $db_password = '';
11    $db_name = 'users';
12
13    $conn = mysqli_connect($db_host, $db_user, $db_password, $db_name);
14    if (!$conn) {
15        die("Connection failed: " . mysqli_connect_error());
16    }
17    $query = "SELECT * FROM userstable";
18    $result = mysqli_query($conn, $query);
19
```

```
if ($result) {
    $isuser=false;
    while ($row = mysqli_fetch_assoc($result)) {
        $userId = $row['id'];
        $usernameadb = $row['username'];
        $passwordadb = $row['password'];
        $random_number=$row['random_number'];
        if($username==$usernameadb){
            $isuser=TRUE;
            if($random_number==1){
                session_start();
                $_SESSION['hashpassword_db'] = $passwordadb;
                $_SESSION['hashpassword'] = $password;
                $_SESSION['username'] = $usernameadb;
                $_SESSION['data'] = $random_number-1;
                break;
            }
            if($random_number==0){
                echo("Your Session is finished After this timer, you will be automatically Redirected");
                $timerDuration = 5;
                sleep($timerDuration);
                header('Location:http://localhost/login-system/Registration-folder/Register.html');
                exit();
            }
        }
    }
}
```

```

        $hashed_password = hash('sha256', $password);
        for ($i = 0; $i < $random_number - 2; $i++) {
            $hashed_password = hash('sha256', $hashed_password);
        }
        session_start();
        $_SESSION['hashpassword_db'] = $passworddb;
        $_SESSION['hashpassword'] = $hashed_password;
        $_SESSION['username'] = $username;
        $_SESSION['data'] = $random_number-1;
        break;
    }
}
if($isuser==false){
    header('Location: http://localhost/login-system/Login-folder/Error-Page');
    exit();
}
header('Location: http://localhost/login-system/Login-folder/Server-side.php');
exit();
mysqli_free_result($result);
} else {
    echo "Error: " . mysqli_error($conn);
}
?>

```

تمام گام های زیر انجام میشود:

۱. آلیس (U) user_name را به سرور می فرستد.
۲. باب یا سرور پایگاه داده را بررسی می کند تا ببیند user_name وجود دارد یا خیر، اگر وجود دارد، سپس n را بر اساس آن برمی گرداند.
۳. در این زمان، فرمی برای آلیس ظاهر می شود تا رمز عبور را وارد کند (PO)، PO به این شکل که یک بار n - 1 هش می شود و سمت باب فرستاده می شود و n مقدار Bob است که به تازگی برگردانده شده است.

```

1 <?php
2 session_start();
3 if (isset($_SESSION['data'])) {
4     $getrandom = $_SESSION['data'];
5     echo "Received data: $getrandom";
6     $hashpassword_db = $_SESSION['hashpassword_db'];
7     $hashed_password = $_SESSION['hashpassword'];
8     $Final_hashed_password = hash('sha256', $hashed_password);
9 > if($Final_hashed_password==$hashpassword_db){...
19 $newPassword = $hashed_password;
20 $newRandom = $getrandom;
21 $newUsername = $_SESSION['username'];
22 $sql = "UPDATE userstable SET password = '$newPassword', random_number = '$newRandom' WHERE use
23
24 if (mysqli_query($conn, $sql)) {
25     header('Location: http://localhost/login-system/main.php');
26     exit();
27 }
28 else {
29     header('Location: http://localhost/login-system/Login-folder/Error-Page');
30     exit();
31 }
32 }
33 ?>

```

۴. وقتی باب کد هش پسورد را دریافت کرد و دوباره از آن هش می‌گیرد ، $h^n(P0)$ خواهد داشت.

۵. باب $h^n(P0)$ را با $h^n(P)$ ذخیره شده در پایگاه داده مقایسه می‌کند، اگر این دو مقدار مطابقت داشته باشند، با موفقیت وارد شوید و ۲ مقدار $h^{(n-1)}(P0)$ و $n-1$ را به پایگاه داده، بنابراین رمز عبور ذخیره شده در پایگاه داده به روز می‌شود.

در زیر تمام حالت های مختلف تست میکنیم:

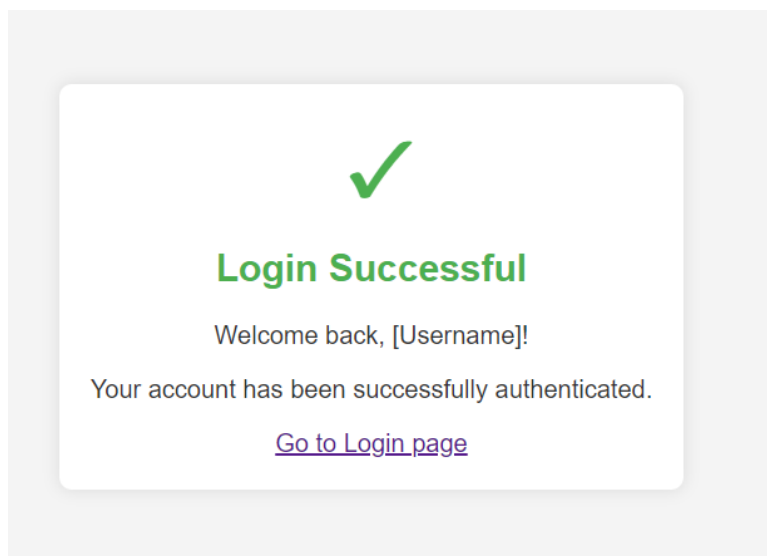
زمانی که کاربر مشخصات درست خواهد داشت قبل از ورود:

	id	username	password	random_number
<input type="checkbox"/>	4	mahroonoohi	d6a804981ea7ce374acc21c9a8bf82f50b684b0ea4bdf8b26a...	3

بعد از ورود:

	id	username	password	random_number
<input type="checkbox"/>	4	mahroonoohi	e0bc614e4fd035a488619799853b075143deea596c477b8dc0...	2

اطلاعات ابدیت شد.



اگر کاربر اطلاعات غلط وارد کند :

Login Failed

The provided username and password do not match our records. Please try again.

[Go back to login](#)

هنگامی که عدد تصادفی به * میرسد مجدد به صفحه ثبت نام میرویم:

<div><div>←T→</div><div>▼ idusernamepasswordrandom_number</div></div>				
<input type="checkbox"/>	 Edit	 Copy	 Delete	4 mahroonoohi 10