



پروژه پایانی درس مبانی امنیت سایبری

پیاده سازی سیستم ورود و ثبت نام بر اساس پروتکل Lamport

شقایق شهبازی ۹۹۳۶۱۳۰۴۰

پیاده‌سازی سیستم لاگین مبتنی بر لمپورت

در این پروژه هدف پیاده سازی یک سیستم احراز اصالت شامل فرآیند ثبت نام کاربر ورود به حساب کاربری خود، براساس پروتکل لمپورت و با استفاده از الگوریتم هش sha256، است که با استفاده از وب فریمورک Django پیاده سازی شده است. در این پروژه بخش هایی وجود دارد، همچون :

- صفحه اصلی: برای هدایت کاربر به صفحات Login یا Sign Up
- صفحه Sign Up: صفحه ای برای ثبت نام کاربر. در این صفحه کاربر با وارد کردن نام کاربری، n که همان تعداد دورهای اولیه هش رمز عبور است و پسوردی که n بار با الگوریتم sha256 هش شده است، اقدام به ثبت نام خود در سیستم می‌کند.
- صفحه Login: صفحه‌ای که در آن کاربری که پیش‌تر ثبت نام کرده است با وارد کردن نام کاربری و پسوردی که n-1 هش شده است اقدام به ورود به سیستم می‌کند و به دنبال آن تعداد n آن یکی کم می‌شود و پسورد نیز با مقدار پسورد ارسال شده آپدیت می‌شود. همچنین در این صفحه قابلیت دیگری در نظر گرفته شده که کاربر با استفاده از آن بتواند از مقدار کنونی n حساب خود آگاه شود. در صورتی که تعداد n تمام شده باشد یعنی برابر با یک بشود کاربر با وارد کردن نام کاربری و پسورد خود به جای ورود به سیستم، به صفحه Set New Password هدایت می‌شود.
- صفحه Set New Password: در این صفحه کاربری که نام کاربری خود را در صفحه Login وارد کرده و تعداد n آن نام کاربری برابر یک است، اقدام به وارد کردن n جدید و پسوردی که n بار هش شده است می‌کند و رکورد مربوط به این کاربر در دیتابیس آپدیت خواهد شد.

نحوه اجرای پروژه:

جهت اجرای پروژه کفایست در ترمینال با واردن کردن دستور زیر اقدام به نصب نیازمندی‌های پروژه کرد:

```
pip install -r requirements.txt
```

سپس با زدن دستور زیر پروژه را اجرا کرد:

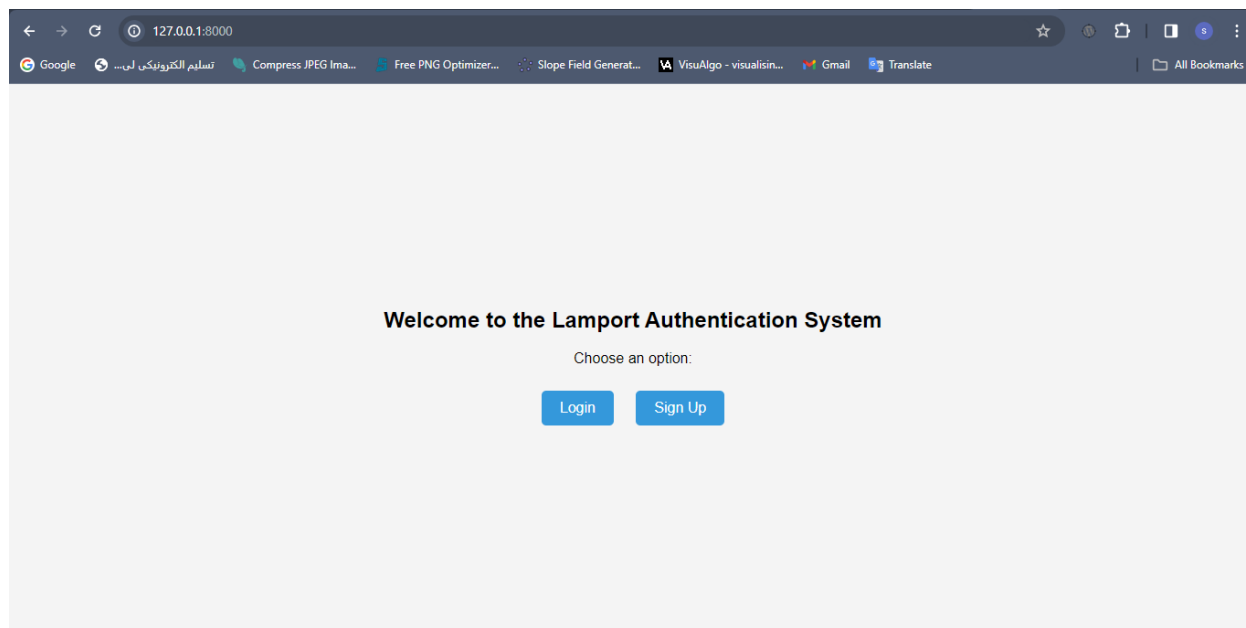
```
Python manage.py runserver
```

و با رفتن به url زیر وارد صفحه اصلی پروژه خواهید شد:

<http://127.0.0.1:8000/>

صفحه Home:

این صفحه، صفحه اصلی پروژه است که به صورت تصویر زیر است:

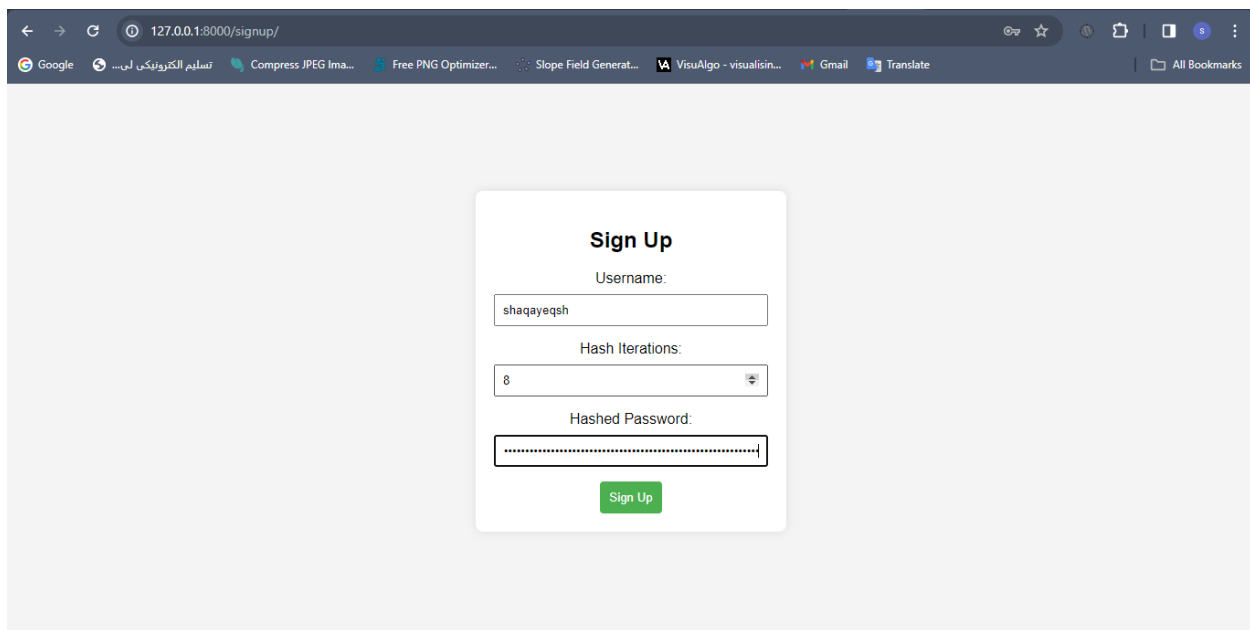


کد مربوط به اجرای این صفحه که تنها فایل Templates/home.html را رندر خواهد کرد:

```
def home(request):  
    return render(request, 'home.html')
```

صفحه Sign Up:

در این صفحه کاربر به ثبت نام حساب کاربری خود می‌کند. در تصویر زیر اقدام به ایجاد کاربری با نام کاربری shaqayeqsh و با تعداد دورهای اولیه هش ۸ و پسورد ۱۴۷ که ۸ بار هش شده است، می‌کنیم:



127.0.0.1:8000/signup/

Google تسلیم الکترونیکی لی... Compress JPEG Ima... Free PNG Optimizer... Slope Field Generat... VisuAlgo - visualisin... Gmail Translate All Bookmarks

Sign Up

Username:

shaqayeqsh

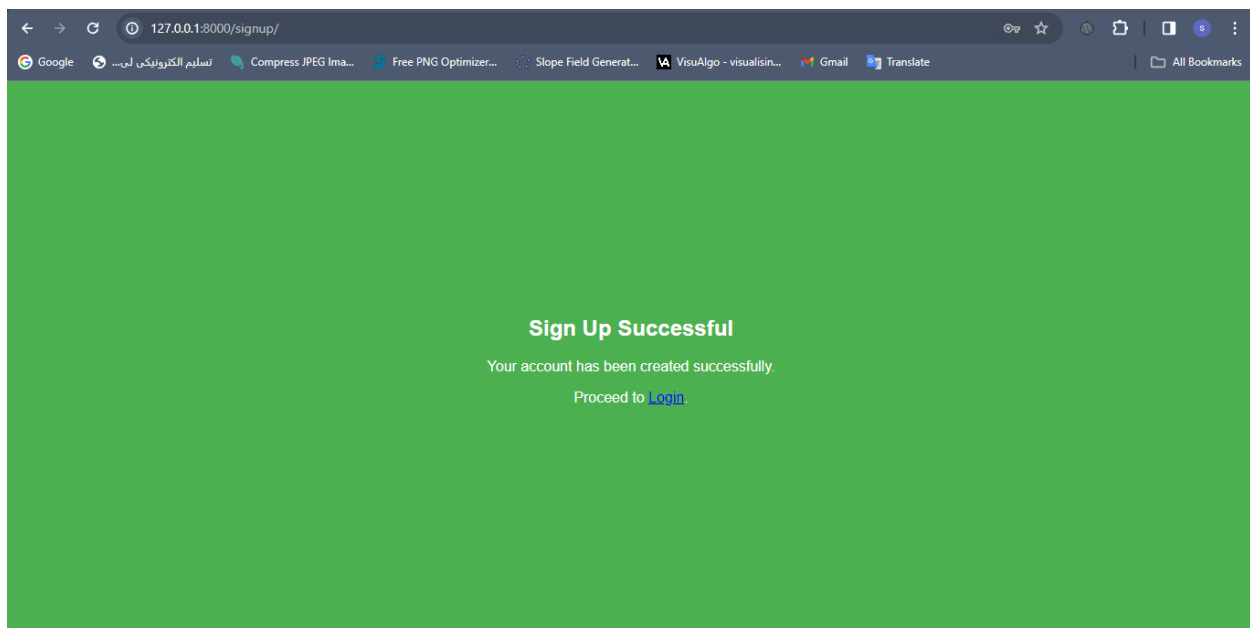
Hash Iterations:

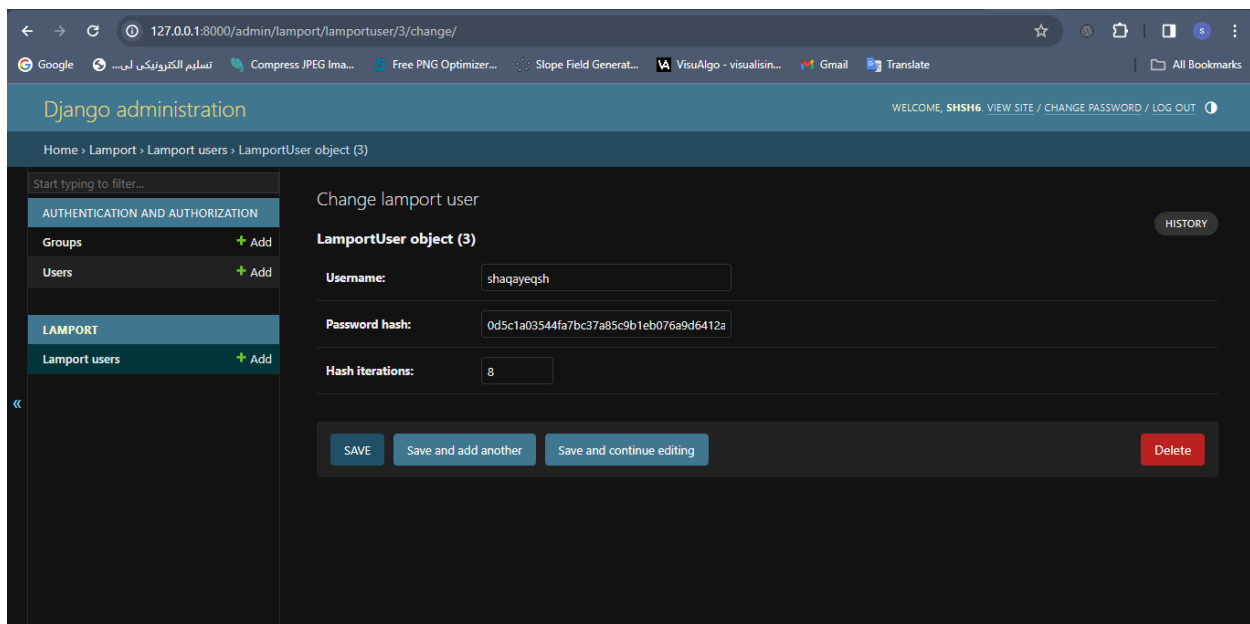
8

Hashed Password:

Sign Up

سپس با زدن بر روی دکمه Sign Up وارد صفحه زیر خواهیم شد و در دیتابیس مشاهده خواهیم که برای این کاربر یک رکورد ایجاد شده است:





کد مربوط به این عملکرد سیستم در تابع signup در فایل view.py به این صورت پیاده شده است:

```
def signup(request):
    if request.method == 'POST':
        username = request.POST['username']
        hash_iterations = int(request.POST['hash_iterations'])
        hashed_password = request.POST['hashed_password']

        LampportUser.objects.create(username=username, hash_iterations=hash_iterations, password_hash=hashed_password)

        # Redirect to a success page or login page after successful signup
        return render(request, 'signup_success.html')

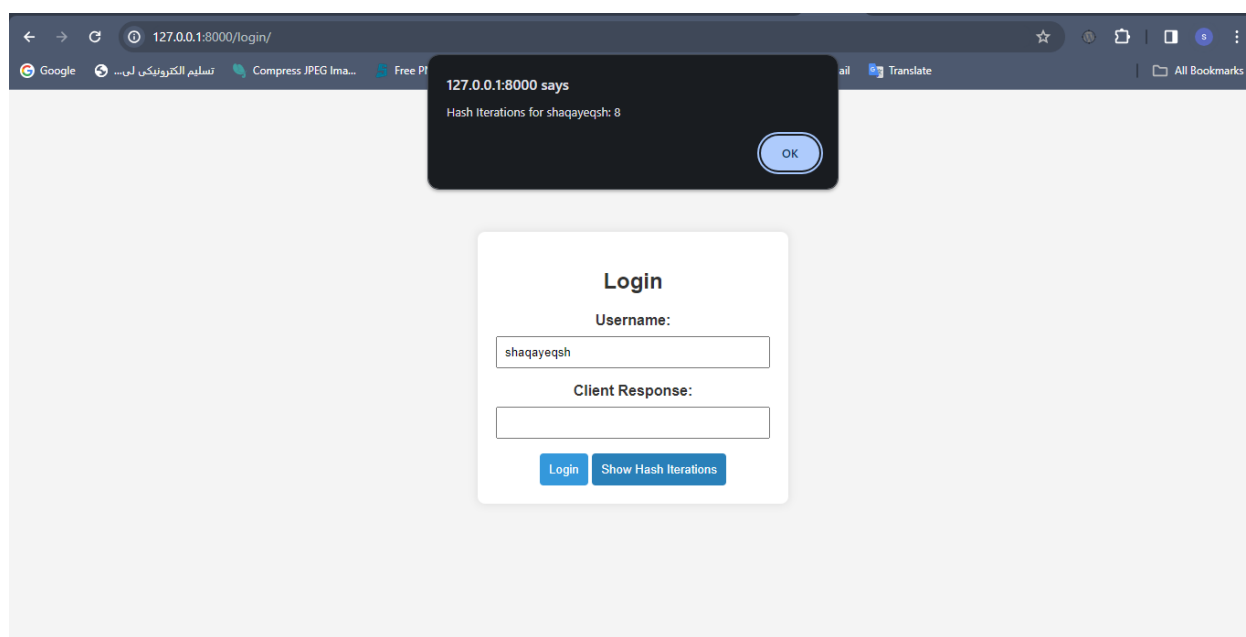
    return render(request, 'signup.html')
```

که در اینجا در صورت ارسال ریکوئست از نوع GET به این URL فایل signup.html رندر خواهد شد و در صورت ارسال ریکوئست از متد POST نام کاربری، تعداد دور هش و رمز هش شده دریافت می‌شود و کاربری از نوع LampportUser ایجاد خواهد شد. فیلدهای این مدل به صورت زیر هستند:

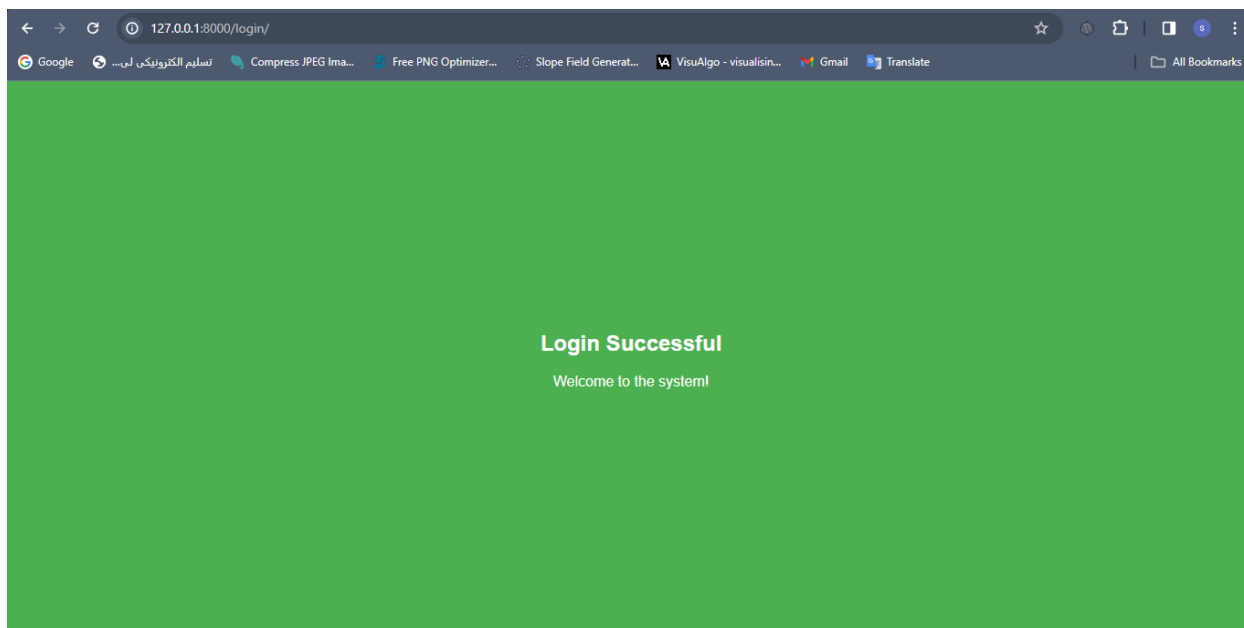
```
class LampportUser(models.Model):
    username = models.CharField(max_length=50, unique=True)
    password_hash = models.CharField(max_length=64) # Assuming SHA-256 for simplicity
    hash_iterations = models.IntegerField(default=100)
```

صفحه Login:

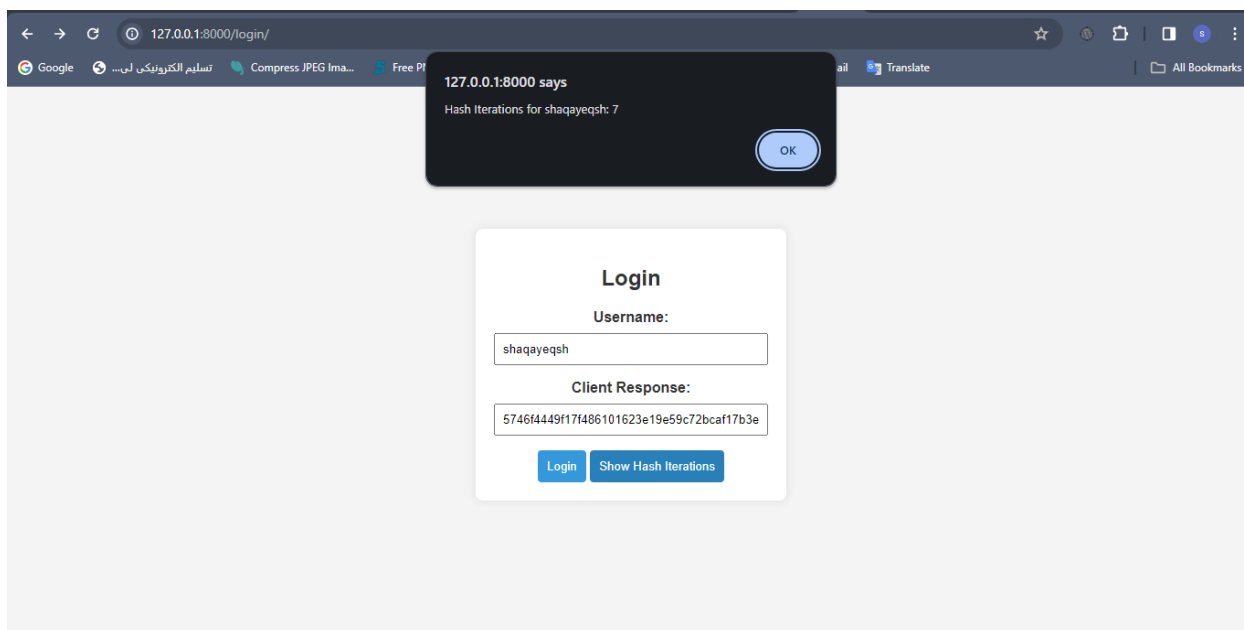
در این صفحه کاربر میتواند وارد حساب کاربری خود شود. همچنین می‌تواند تعداد دور هش خود را ببینید. به این صورت که کاربر در صورتی که نمی‌داند چند دور باید پسورد خود را هش کند و آن را ارسال کند با زدن بر روی دکمه Show Hash Iterations عددی به اون نشان داده خواهد شد و کاربر متوجه می‌شود باید به همان تعداد دور ۱- پسورد خود را هش بگیرد و در صورت نزدن نام کاربری خطایی به کاربر نشان داده خواهد شد که حتما باید پسورد خود را وارد کند. نمونه‌ای از آن برای یوزری که پیش‌تر ایجاد کردیم در تصویر زیر قابل مشاهده است:

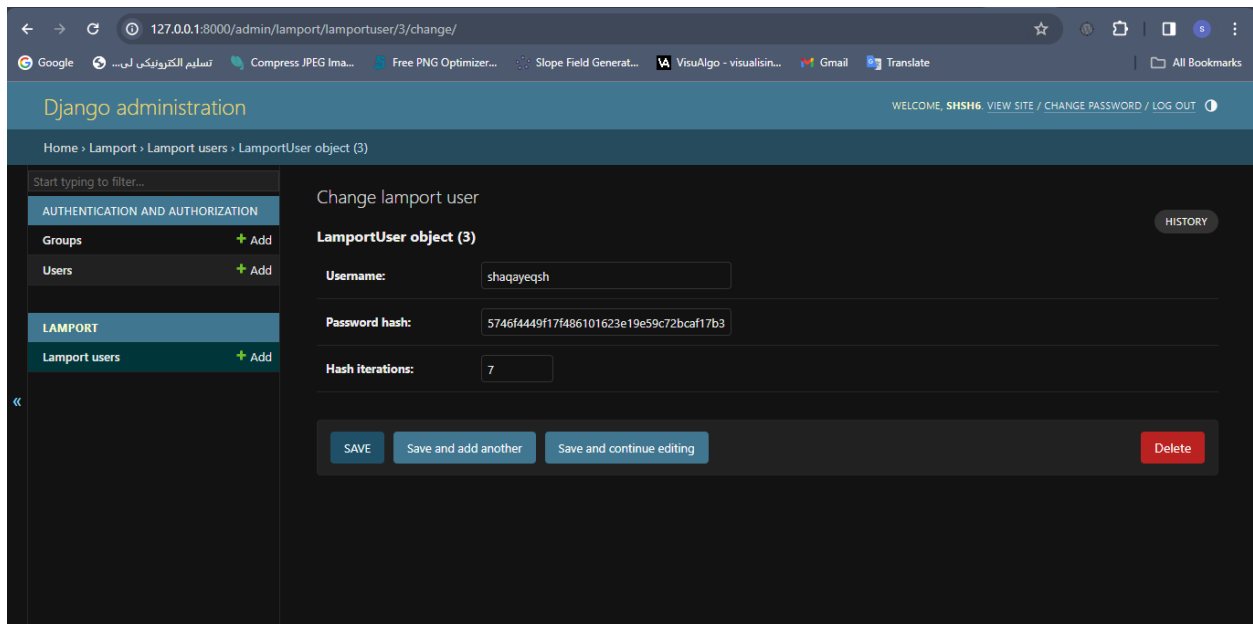


بنابراین متوجه خواهیم شد که باید ۱-۸ یعنی ۷ بار پسورد ۱۴۷ خود را هش بگیریم و برای سرور ارسال کنیم. با ۷ بار هش گرفتن پسورد و ارسال آن به همراه یوزرنیم و با زدن بر روی دکمه لاگین به صفحه زیر هدایت خواهیم شد و وارد سیستم می‌شویم:



و سپس با بازگشت مجدد به صفحه لاگین و زدن بر روی دکمه Show Hash Iterations میبینیم که تعداد دورهای هش یکی کم شده است. و با مشاهده رکورد مربوط به این یوزر درمی یابیم که پسورد یوزر آپدیت شده و تعداد دورهای هش هم یکی کم شده است.





کد مربوط به این عملکرد در تابع login در views.py موجود است:

```
def login(request):
    hash_iterations = None

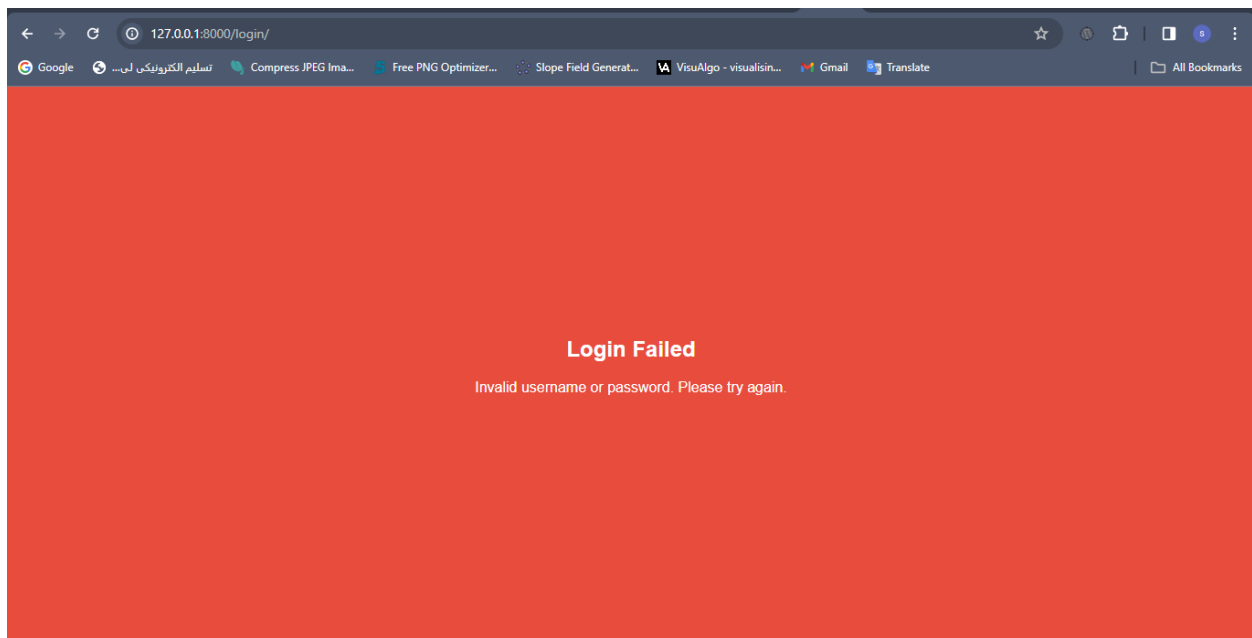
    if request.method == 'POST':
        username = request.POST['username']
        client_response = request.POST['client_response']

        try:
            user = LampportUser.objects.get(username=username)
            hash_iterations = user.hash_iterations
            server_password = user.password_hash
            if server_password == hashlib.sha256(client_response.encode()).hexdigest():
                # Authentication successful, redirect to a success page
                if hash_iterations == 1:
                    return redirect('set_new_password', username=username)
                else:
                    user.hash_iterations -= 1
                    user.password_hash = client_response
                    user.save()
                    return render(request, 'success.html')
            else:
                # Authentication failed, redirect to a failure page
                return render(request, 'failure.html')
        except LampportUser.DoesNotExist:
            # User not found, redirect to a failure page
            return render(request, 'failure.html')

    return render(request, 'login.html')
```


در اینجا در صورت GET بودن متد ریکوئست ارسال شده فایل login.html رندر خواهد شد و در صورت POST بودن نام کاربری دریافت خواهد شد و طبق آن شی مربوط به آن کاربر از دیتابیس دریافت خواهد شد سپس از پسوردی که کاربر وارد کرده یک دور هش گرفته می‌شود و با مقدار موجود در دیتابیس مقایسه می‌شود و در صورت برابر بودن چک می‌شود که آیا تعداد دورهای هش آن کاربر تمام شده یا نه. که در صورت تمام شدن یعنی برابر یک بودن، کاربر به صفحه مربوط به ایجاد پسورد جدید هدایت می‌شود و در غیر این صورت یکی از تعداد دورهای هش کم می‌شود و به همراه پسورد جدیدی که کاربر ارسال کرده در دیتابیس برای آن یوزر ذخیره خواهد شد و به صفحه success.html هدایت می‌شود

و درنیز در صورت برابر نبودن این دو پسورد کاربر به صفحه failure.html هدایت می‌شود که در به صورت زیر است:



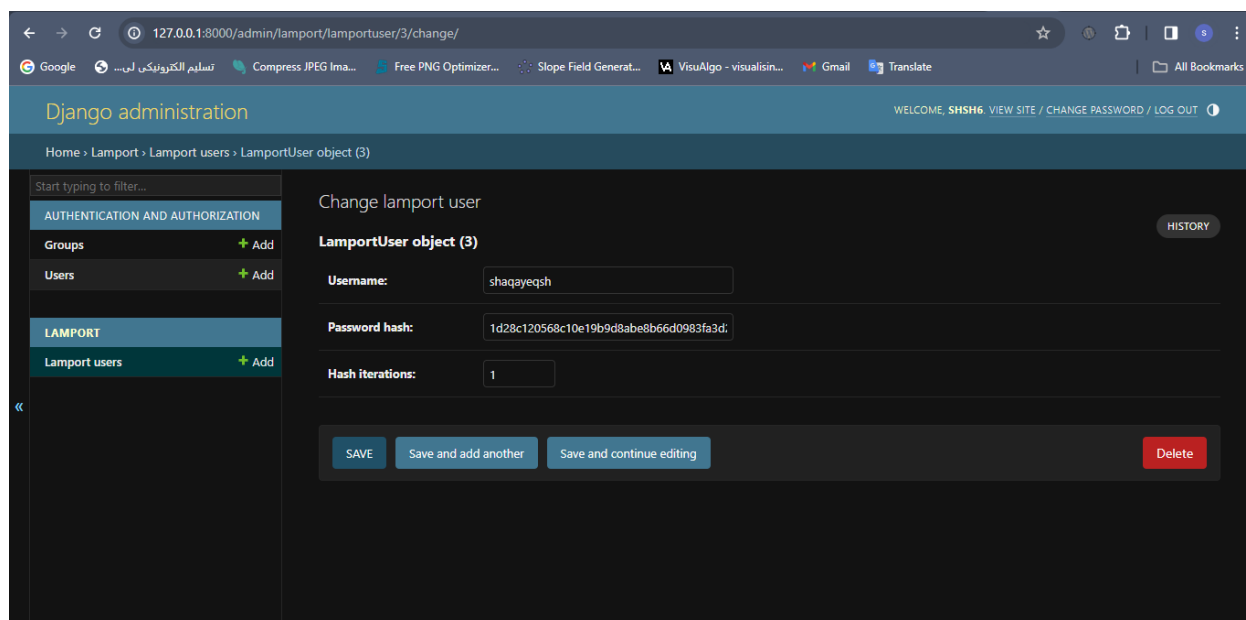
همچنین پیاده سازی مربوط به عملکرد نمایش تعداد دورهای هش کاربر در تابع get_hash_iterations در views.py پیاده شده است که با گرفتن یوزرنیم کاربر، به صورت پارامتر در url، شی مربوط به آن را از دیتابیس دریافت می‌کند و مقدار تعداد دورهای هش آن را باز می‌گرداند:

```
def get_hash_iterations(request):
    username = request.GET.get('username', '')
    try:
        user = LamportUser.objects.get(username=username)
        hash_iterations = user.hash_iterations
        return JsonResponse({'hash_iterations': hash_iterations})
    except LamportUser.DoesNotExist:
        return JsonResponse({'error': 'User not found'})
```

صفحه Set New Password:

در این صفحه کاربر که یوزرنیم خود پیش‌تر در صفحه لاگین وارده کرده اقدام به وارد کردن یک پسورد جدید به همراه تعداد دور هش‌های آن می‌کند و زمانی کاربر به این صفحه هدایت خواهد شد که در هنگام لاگین تعداد دورهای هش‌های آن در دیتابیس برابر یک باشد.

مقدار کنونی یوزر در دیتابیس با تعداد دور هش یک:

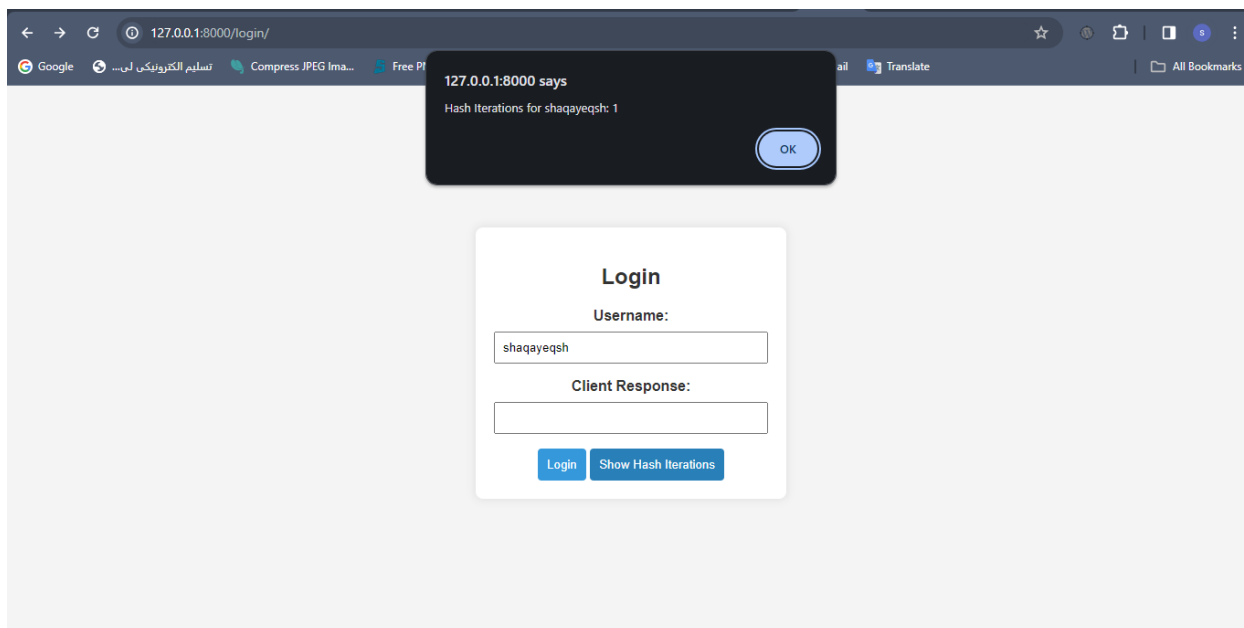


The screenshot shows the Django administration interface for a user named 'shagayeqsh'. The page title is 'Change lampport user'. The form contains the following fields:

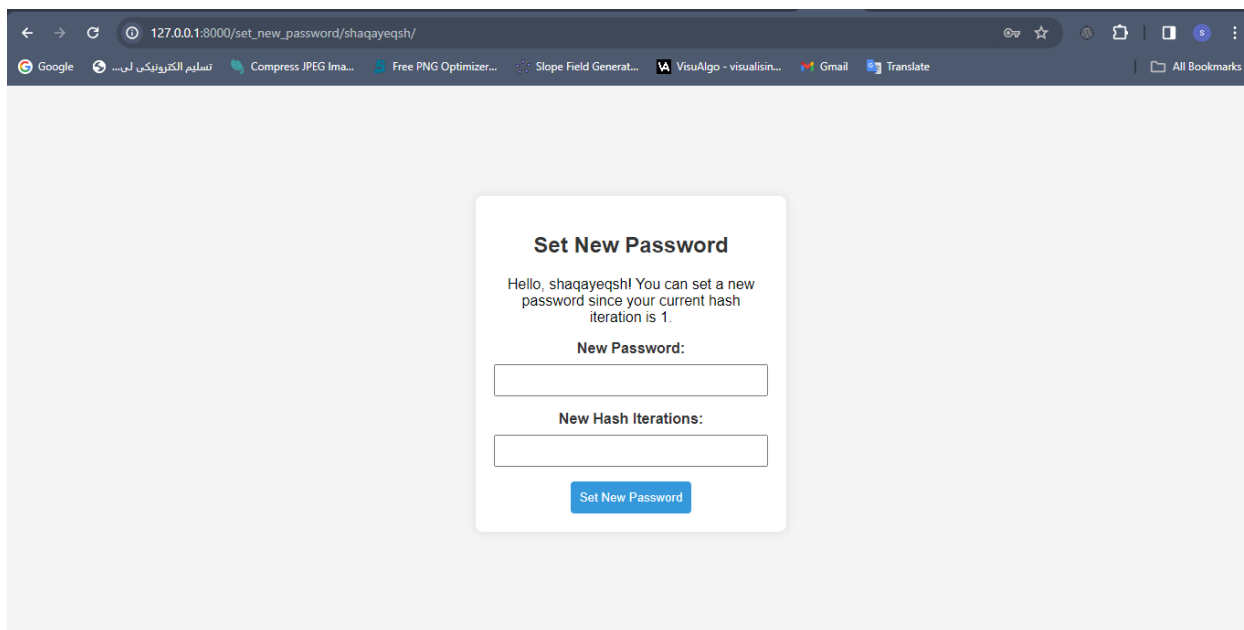
- Username:** shagayeqsh
- Password hash:** 1d28c120568c10e19b9d8abe8b66d0983fa3d
- Hash iterations:** 1

At the bottom of the form, there are four buttons: 'SAVE', 'Save and add another', 'Save and continue editing', and 'Delete'.

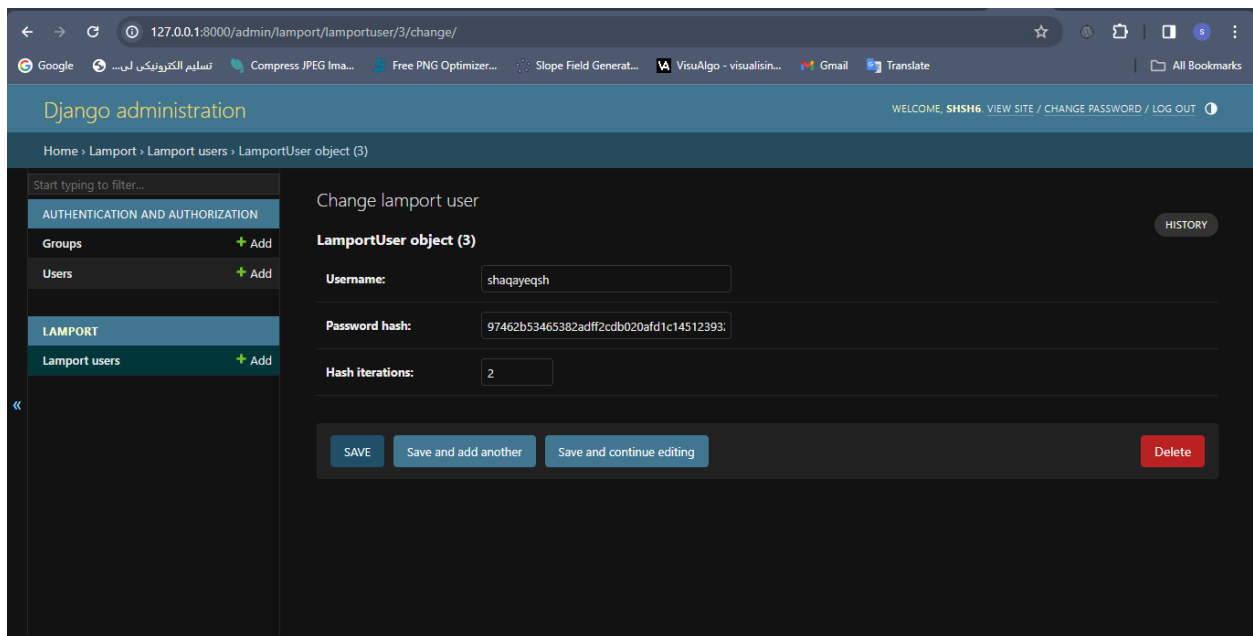
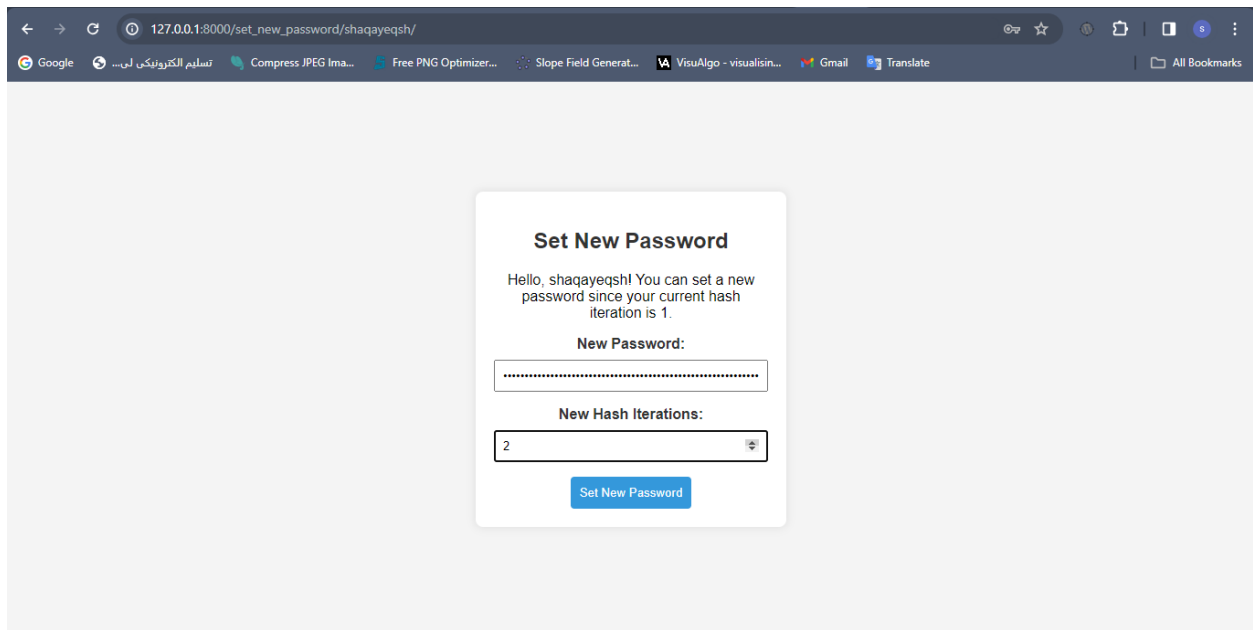
تعداد دورهای هش:



هدایت به صفحه ایجاد پسورد جدید:



وارد کردن پسورد جدید و ورود به سیستم و آپدیت رکورد مربوط به یوزر در دیتابیس:



پیاده سازی مربوط به این عملکرد در تابع `set_new_password` در `views.py` انجام گرفته است:

```

def set_new_password(request, username):
    user = LamportUser.objects.get(username=username)

    if request.method == 'POST':
        new_password = request.POST['new_password']
        new_hash_iterations = int(request.POST['new_hash_iterations'])

        # Update user record with new password and hash iteration
        user.password_hash = hashlib.sha256(new_password.encode()).hexdigest()
        user.hash_iterations = new_hash_iterations
        user.save()

    return render(request, 'success.html', {'hash_iterations': new_hash_iterations})

return render(request, 'set_new_password.html', {'username': username, 'current_hash_iterations': user.hash_iterations})

```