Create new User (social media)

Form.py

```
class UserRegistrationForm(forms.ModelForm):

    password=forms.CharField(widget=forms.PasswordInput(attrs={'placeholder':'enter password here....'}))
    confirm_password = forms.CharField(widget=forms.PasswordInput(attrs={'placeholder': 'confirm password....'}))

    class Meta:
        model=User
        fields=(
            'username',
            'first_name',
            'last_name',
            'email',
        )

    def clean_confirm_password(self):
        print('clean_confirm_password')
        password = self.cleaned_data.get('password')
        confirm_password = self.cleaned_data.get('confirm_password')

        if password != confirm_password:
            print('password missmatch')
            raise forms.ValidationError("password missmatch")

        return confirm_password
```

Views.py

```
def registration(request):

    if request.method == "POST":

            formm = UserRegistrationForm(request.POST)

            if formm.is_valid():
                new_user=formm.save(commit=False)
                new_user.set_password(formm.cleaned_data['password'])
                new_user.save()

                # after successful signup automatic login

                user = authenticate(request, username=formm.cleaned_data['username'], password=formm.cleaned_data['password'])

                if user is not None:

                    if user.is_active:
                        login(request, user)
                        print('user login')
                        return redirect('post_list')

                else:
                    return HttpResponse('User name or password not matched')
    else:
        formm=UserRegistrationForm()

        context={
            'form':formm,
        }

        return render(request,’registration/register.html',context)
```

Urls.py

```
url(r'^register/$', views.registration, name=‘register'),
```

register.html

```
{% extends 'blog/base.html' %}

{% block content %}

    <form method="post">

        {% csrf_token %}
        {{ form.as_p }}

        <input type="submit" value="Register" />

    </form>

{% endblock %}
```

Login (socialmedia)

Views.py

```
from django.contrib.auth import authenticate, login, logout

def user_login(request):

    if request.method == 'POST':

        formm = UserLoginForm(request.POST)

        if formm.is_valid():
            username = request.POST['username']
            password = request.POST['password']
            user = authenticate(request, username=username, password=password)

            if user is not None:

                if user.is_active:
                    login(request,user)
                    print('user login')
                    return redirect('post_list')
                else:
                    return HttpResponse('user not active')

            else:
                return HttpResponse('User name or password not matched')
    else:

        formm=UserLoginForm()

        context={
            'form':formm,
        }

        return render(request,'blog/login.html',context)
```

Urls.py

```
path('', views.post_list,name=‘post_list'),
```

Forms.py

```
class UserLoginForm(forms.Form):
        username=forms.CharField(label="")
        password=forms.CharField(label=“",widget=forms.PasswordInput)
```

Logout

Views.py

```
from django.contrib.auth import authenticate, login, logout

def user_logout(request):
    logout(request)
    return redirect(‘post_list')
```

check user login or not in html

```
{% if user.is_authenticated %}

                <span>
                       Hi, {{ request.user | capfirst }}, <a href="logout">Logout</a>
                </span>

{% else %}

                   <a href="login" >Login</a>

{% endif %}
```

check user login or not in views.py

views.py

```
from django.contrib.auth.decorators import login_required

@login_required(login_url='/')
def CreatePost(request):
```

Password Reset (email) (socialmedia)

Study link

https://www.youtube.com/watch?v=Nn0sYcp3pOs&list=PLKILtxhEt4-RT-GkrDkJDLuRPQfSK-6Yi&index=25

https://simpleisbetterthancomplex.com/tutorial/2016/09/19/how-to-create-password-reset-view.html

1. Setting.py

```
EMAIL_BACKEND='django.core.mail.backends.console.EmailBackend'
```

2. urls.py

```
#password reset url

url(r'^password-reset/$', auth_views.password_reset, name='password_reset'),

url(r'^password-reset/done/$', auth_views.password_reset_done, name='password_reset_done'),

url(r'^password-reset/confirm/(?P<uidb64>[\w-]+)/(?P<token>[\w-]+)/$', auth_views.password_reset_confirm, name='password_reset_confirm'),

url(r'^password-reset/complete/$', auth_views.password_reset_complete, name='password_reset_complete'),
```

OR Just write bottom line

```
from django.conf.urls import include

url('^', include('django.contrib.auth.urls')),
```

3. Create registration folder in template folder and in registration folder create following files

registration/password_reset_form.html

```
{% extends 'blog/base.html' %}


{% block content %}

  <h3>Forgot password</h3>

  <form method="post">

    {% csrf_token %}

    {{ form.as_p }}
    <button type="submit">Submit</button>

  </form>

{% endblock %}
```

registration/password_reset_subject.txt

```
TestSite password reset
```

registration/password_reset_email.html

```
{% autoescape off %}

To initiate the password reset process for your {{ user.get_username }} TestSite Account,

click the link below:


{{ protocol }}://{{ domain }}{% url 'password_reset_confirm' uidb64=uid token=token %}


If clicking the link above doesn't work, please copy and paste the URL in a new browser

window instead.


Sincerely,

The TestSite Team

{% endautoescape %}
```

registration/password_reset_done.html

```
{% extends 'blog/base.html' %}


{% block content %}

  <p>

    We've emailed you instructions for setting your password, if an account exists with the email you entered.

    You should receive them shortly.

  </p>

  <p>

    If you don't receive an email, please make sure you've entered the address you registered with,
    and check your spam folder.
  </p>
{% endblock %}
```

registration/password_reset_confirm.html

```
{% extends 'blog/base.html' %}

{% block content %}
  {% if validlink %}
    <h3>Change password</h3>
    <form method="post">
      {% csrf_token %}
      {{ form.as_p }}
      <button type="submit">Change password</button>
    </form>
  {% else %}
    <p>
      The password reset link was invalid, possibly because it has already been used.
      Please request a new password reset.
    </p>
  {% endif %}
{% endblock %}
```

registration/password_reset_complete.html

```
{% extends 'blog/base.html' %}
{% block content %}
  <p>
    Your password has been set. You may go ahead and <a href="{% url 'user_login' %}">sign in</a> now.
  </p>
{% endblock %}
```