Login & Logout System

Created	@October 14, 2023 7:06 PM

Steps to create a Django Login & Logout system :

1) Whole Project/s urls.py

2) login.html & logout.html

3) settings.py

4) Add the profile in your views

5) create the profile.html template to render the user name there

6) Add the profile link in the URL

7) Update Navbar Depending on user authentication status

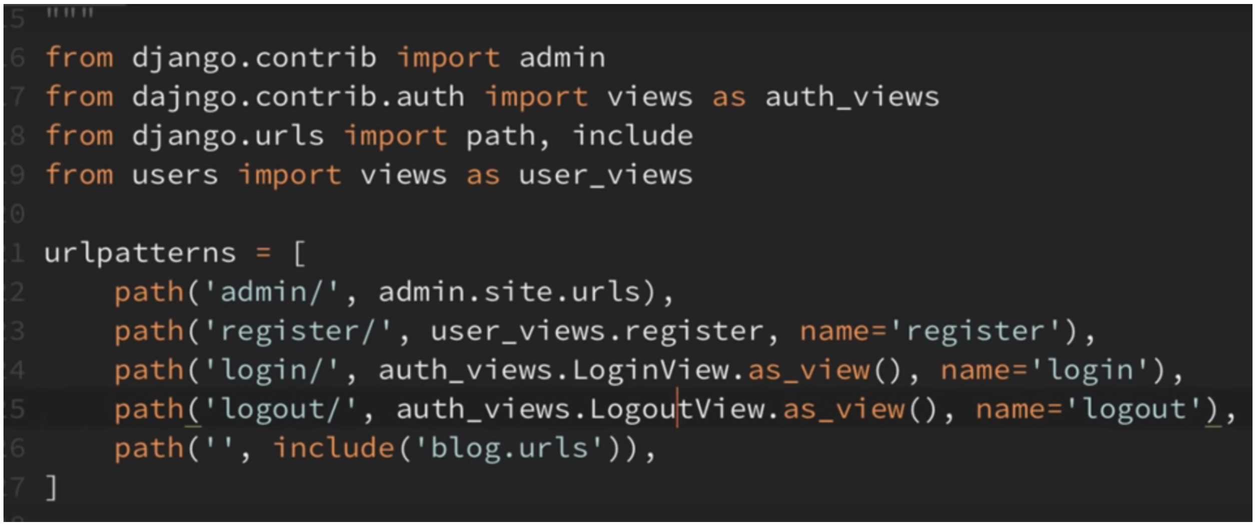
---

1) Whole Project/s urls.py

-use auth to import views as auth_views

-Notice the Capitalized first letters in LoginView & LogoutView in the paths because they are classes that we're gonna create

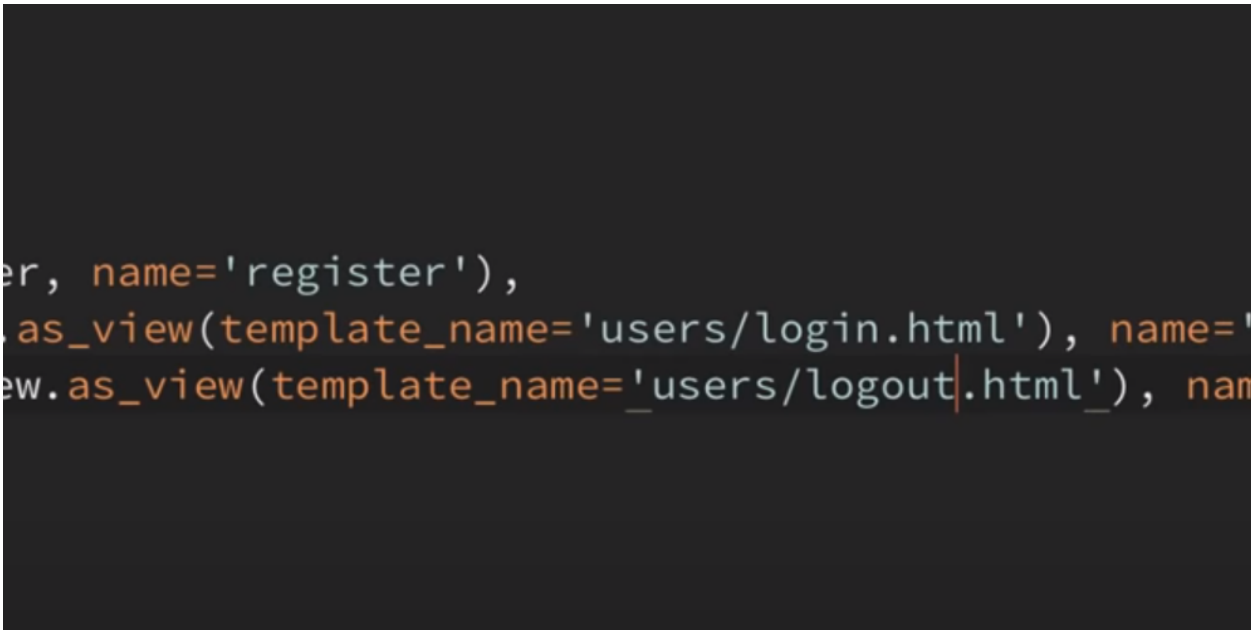
-Don't forget to add .as_view()



```
"""
from django.contrib import admin
from dajngo.contrib.auth import views as auth_views
from django.urls import path, include
from users import views as user_views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('register/', user_views.register, name='register'),
    path('login/', auth_views.LoginView.as_view(), name='login'),
    path('logout/', auth_views.LogoutView.as_view(), name='logout'),
    path('', include('blog.urls')),
]
```

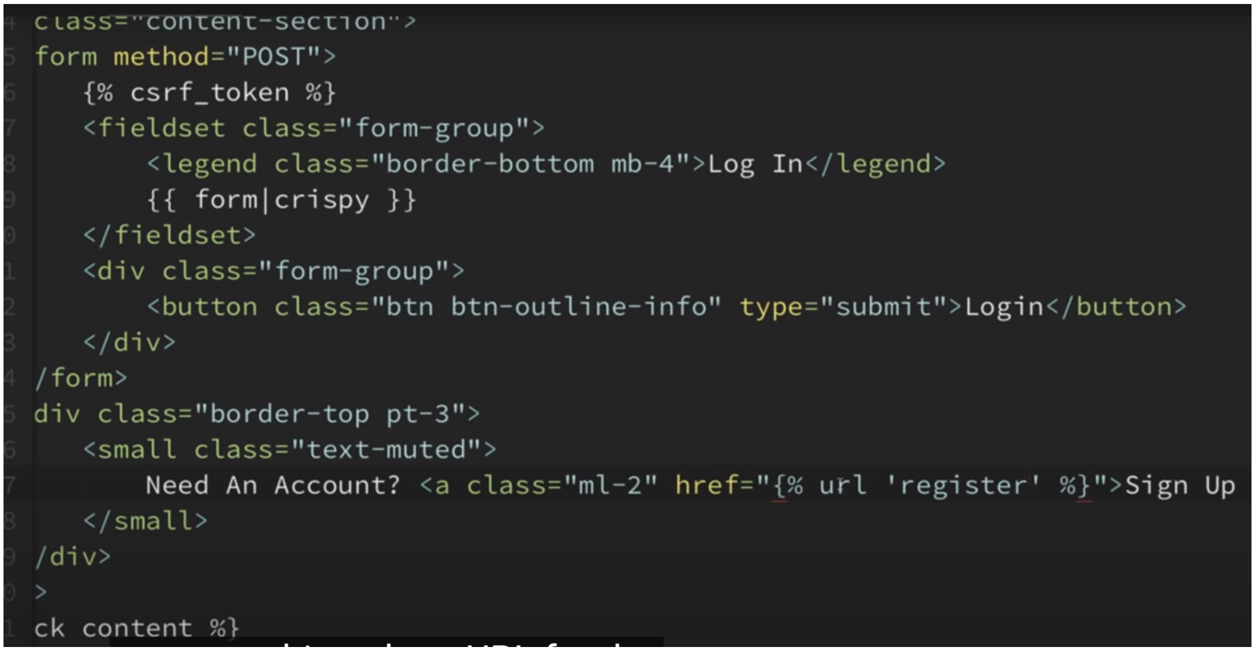
and add your template name in the as_view()



```
er, name='register'),
.as_view(template_name='users/login.html'), name='
ew.as_view(template_name='users/logout.html'), nan
```

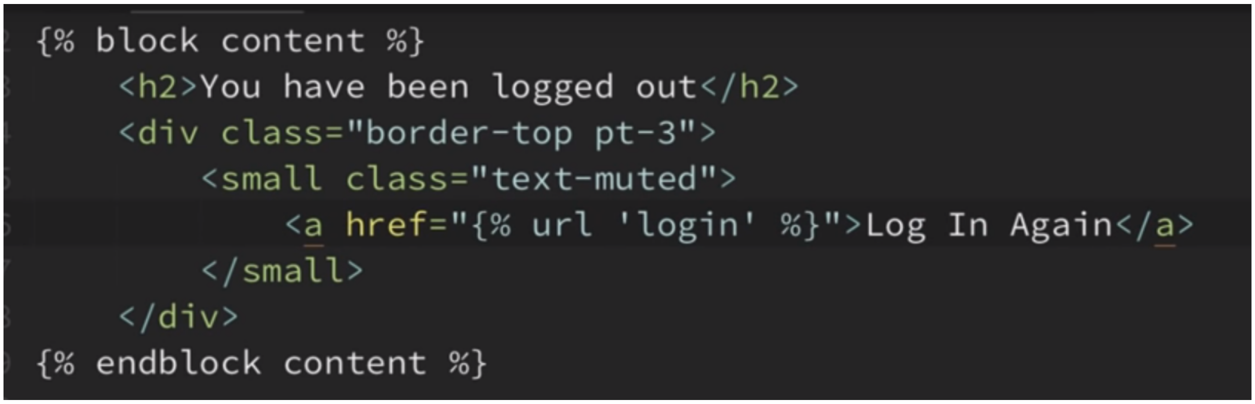
2) In your login.html page :

copy paste the register.html code that we did before and just change the button names to Login and Sign up Instead of submit and login



```
class="content-section">
form method="POST">
   {% csrf_token %}
   <fieldset class="form-group">
       <legend class="border-bottom mb-4">Log In</legend>
       {{ form|crispy }}
   </fieldset>
   <div class="form-group">
       <button class="btn btn-outline-info" type="submit">Login</button>
   </div>
/form>
div class="border-top pt-3">
   <small class="text-muted">
       Need An Account? <a class="ml-2" href="{% url 'register' %}">Sign Up
   </small>
/div>
>
ck content %}
```

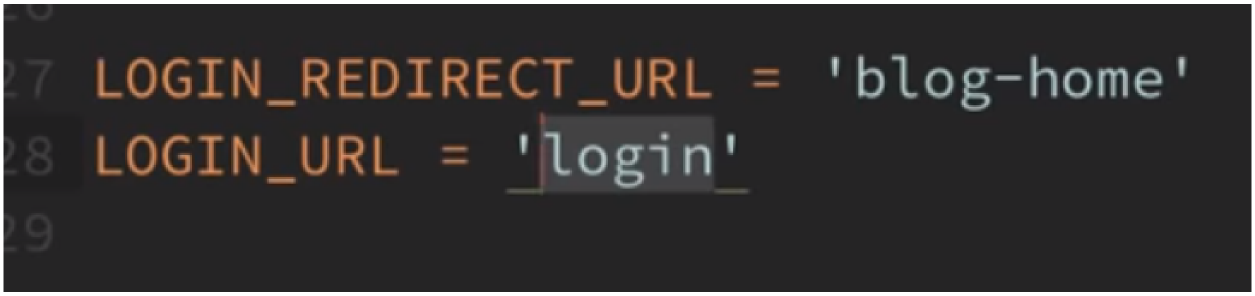
Logout template logout.html



```
{% block content %}
    <h2>You have been logged out</h2>
    <div class="border-top pt-3">
        <small class="text-muted">
            <a href="{% url 'login' %}">Log In Again</a>
        </small>
    </div>
{% endblock content %}
```

3) settings.py

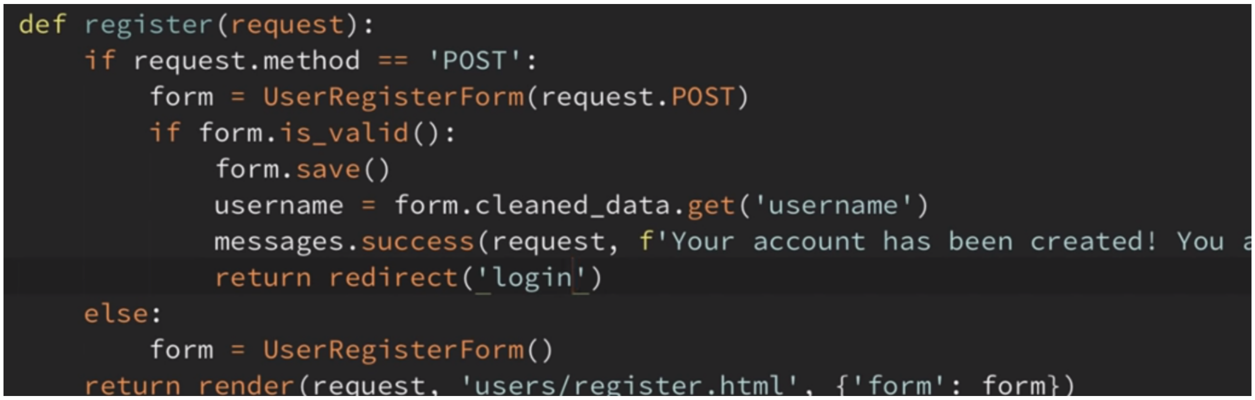
write your home page url name here so when you login, your browser directs you into your home page



```
LOGIN_REDIRECT_URL = 'blog-home'
LOGIN_URL = 'login'
```

-you can no change your view so when a user signs up for the first time, they’re directed to the login page

return redirect(’login’)

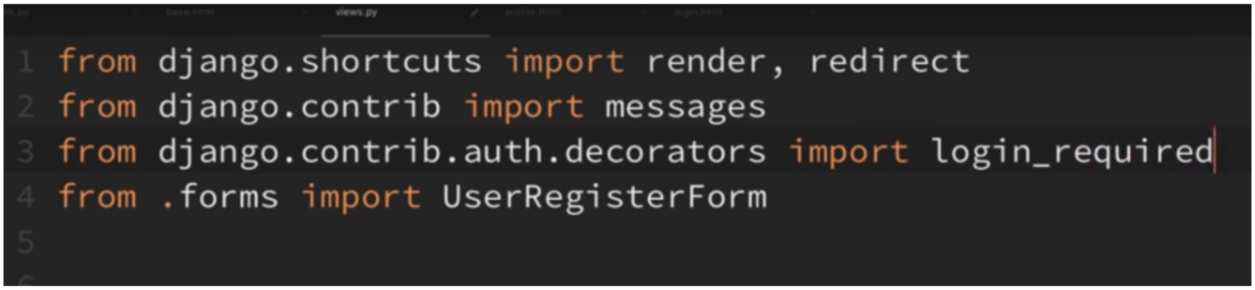


```
def register(request):
    if request.method == 'POST':
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get('username')
            messages.success(request, f'Your account has been created! You a
            return redirect('login')
    else:
        form = UserRegisterForm()
    return render(request, 'users/register.html', {'form': form})
```

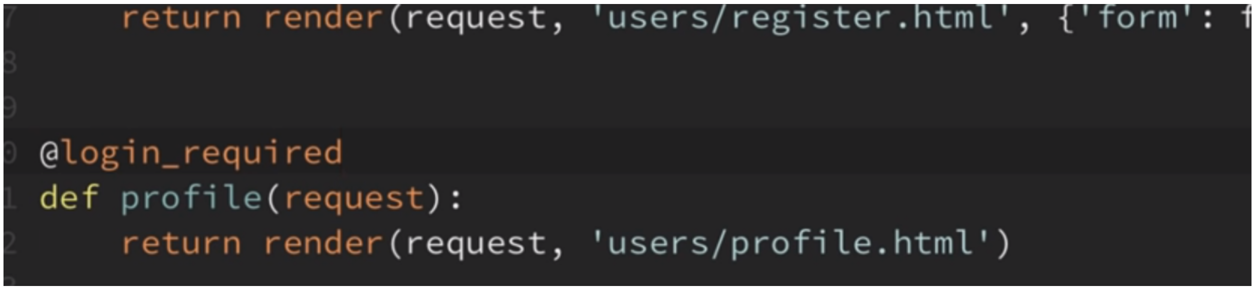
4) Add the profile in your views

- **this profile template is gonna display the user name of the logged on user**
- **import the login_required decorator**

  decorator : adds functionaly to an existing function, and in this case, it adds functionality in the profile view where the user must be logged in to view this page



```python
from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from .forms import UserRegisterForm
```

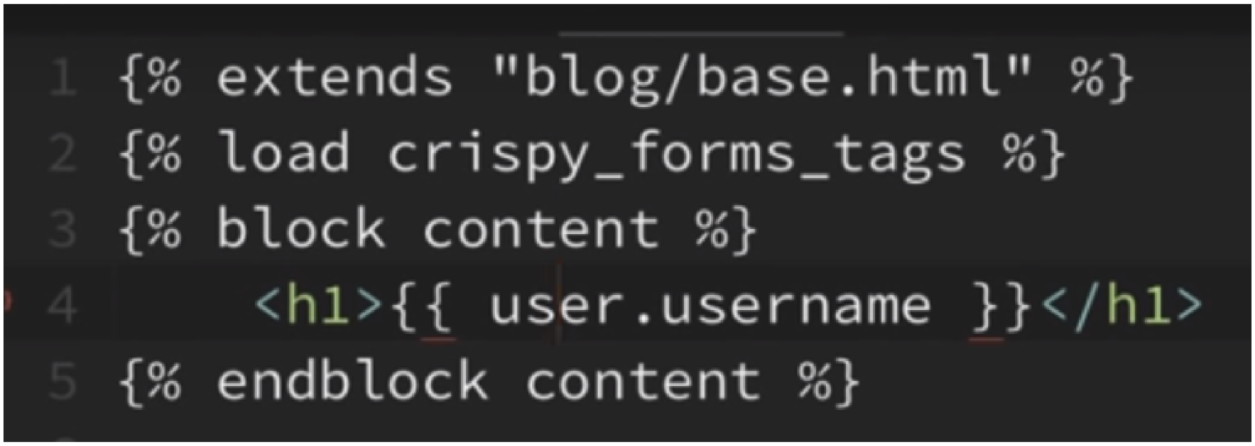


```python
    return render(request, 'users/register.html', {'form': f


@login_required
def profile(request):
    return render(request, 'users/profile.html')
```

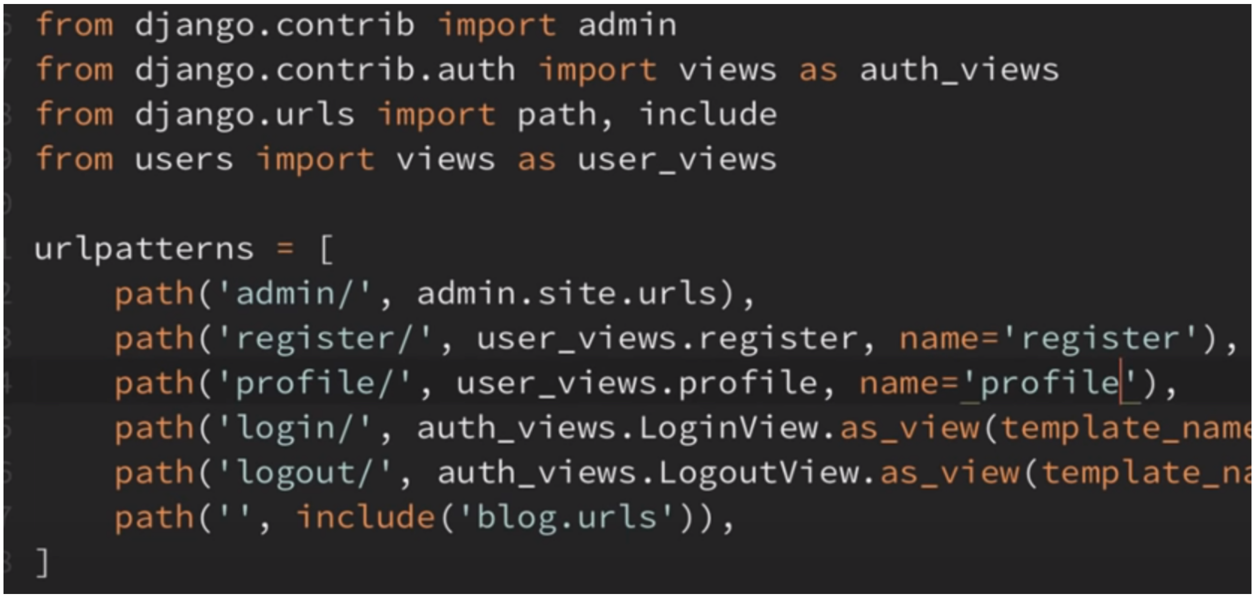
---

5) create the profile.html template to render the user name there



```
{% extends "blog/base.html" %}
{% load crispy_forms_tags %}
{% block content %}
      <h1>{{ user.username }}</h1>
{% endblock content %}
```

6) Add the profile link in the URL



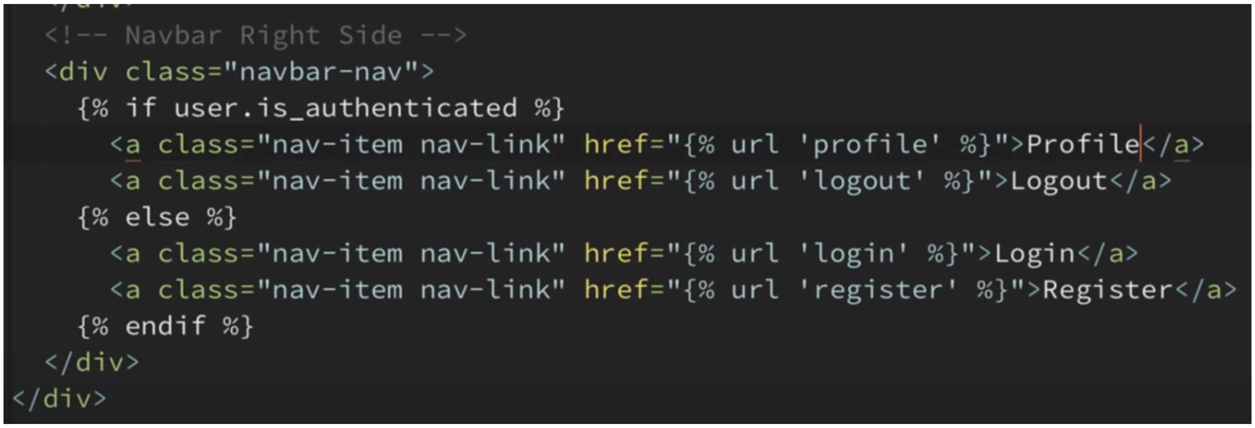
```
from django.contrib import admin
from django.contrib.auth import views as auth_views
from django.urls import path, include
from users import views as user_views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('register/', user_views.register, name='register'),
    path('profile/', user_views.profile, name='profile'),
    path('login/', auth_views.LoginView.as_view(template_name
    path('logout/', auth_views.LogoutView.as_view(template_na
    path('', include('blog.urls')),
]
```

7) Update Navbar Depending on user authentication status

-if user is authenticated (aka loggin in) , show him the log out nav link.

-If not, then show him the Login / Register nav links



```
  </div>
  <!-- Navbar Right Side -->
  <div class="navbar-nav">
    {% if user.is_authenticated %}
      <a class="nav-item nav-link" href="{% url 'profile' %}">Profile</a>
      <a class="nav-item nav-link" href="{% url 'logout' %}">Logout</a>
    {% else %}
      <a class="nav-item nav-link" href="{% url 'login' %}">Login</a>
      <a class="nav-item nav-link" href="{% url 'register' %}">Register</a>
    {% endif %}
  </div>
</div>
```