

ADF Practical – 7 Steps

- Submitted by: - Sanket Chheladiya(18bce042)

-Submitted To: - Prof. Dvijesh Bhatt

Q:-

Demonstrate the concept of permission handling for various users for the Social Media Application.

decorators.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import redirect, render

def allowed_users(allowed_roles=[]):
    def decorator(view_func):
        def wrapper_func(request, *args, **kwargs):
            group = None
            if request.user.groups.exists():
                group = request.user.groups.all()[0].name

            if group in allowed_roles:
                return view_func(request, *args, **kwargs)
            else:
                return render(request, 'registration/permission.html')
        return wrapper_func
    return decorator
```

Views.py

```
from django.shortcuts import render, HttpResponseRedirect
from .forms import RegisterForm, ProfileForm
from django.contrib import messages
from django.shortcuts import render, redirect
from .forms import addPost
from django.contrib.auth.models import User
from .models import Post, UserProfile
from django.conf import settings
from django.core.mail import send_mail
from django.utils.http import urlsafe_base64_encode, urlsafe_base64_decode
from django.contrib.sites.shortcuts import get_current_site
from .token import account_activation_token
from django.core.mail import send_mail
from django.template.loader import render_to_string
from django.utils.encoding import force_bytes, force_text
from .decorators import allowed_users
from django.contrib.auth.models import Group
```

```

def registerUser(request):
    if request.method=="POST":
        form=RegisterForm(request.POST)
        if form.is_valid():
            email = request.POST['email']
            user = form.save(commit=False)
            user.is_active = False
            user.save()
            current_site = get_current_site(request)
            Msub = 'Activate your account on Social Media Site'
            msg = render_to_string('registration/acc_active_email.html', {
                'user': user,
                'domain': current_site.domain,
                'uid': urlsafe_base64_encode(force_bytes(user.pk)),
                'token': account_activation_token.make_token(user),
            })
            email_from = settings.EMAIL_HOST_USER
            recipient_list = [email,]
            send_mail(Msub, msg, email_from, recipient_list)
            return HttpResponse("Please confirm your email address to complete the registration!")
        else:
            context={'form':form}
            messages.error(request,'There is Error in your information...kindly refill the form')
            render(request,'registration/signup.html',context)
            form=RegisterForm()
            context={'form':form}
            return render(request,'registration/signup.html',context)

def activate(request, uidb64, token):
    try:
        uid = force_text(urlsafe_base64_decode(uidb64))
        user = User.objects.get(pk=uid)
    except(TypeError, ValueError, OverflowError, User.DoesNotExist):
        user = None
    if user is not None and account_activation_token.check_token(user, token):
        user.is_active = True
        group = Group.objects.get(name='customer')
        user.groups.add(group)
        user.save()
        messages.success(request, 'Thank you for your email confirmation. Now you can login your account.')
        messages.success(request, 'Please complete your profile in profile section after login.')
        return redirect('login')
    else:
        return HttpResponse('Activation link is invalid!')

@allowed_users(allowed_roles=['admin'])
def add_post(request):
    pi = User.objects.get(username=request.user)
    if(request.method == 'POST'):
        fm = addPost(request.POST)
        if(fm.is_valid()):
            pname = fm.cleaned_data['postname']
            desc = fm.cleaned_data['description']
            reg = Post(postusername=pi, postname=pname, description=desc)
            reg.save()
            fm = addPost()

```

```

else:
    fm = addPost()
    return render(request, 'registration/addPost.html', {'form':fm})

@allowed_users(allowed_roles=['admin'])
def show_post(request):
    stud = Post.objects.filter(postusername=request.user)
    return render(request, 'registration/showPost.html' , {'stu':stud})

@allowed_users(allowed_roles=['admin'])
def delete_post(request, postname):
    if request.method == 'POST':
        pi = Post.objects.get(postusername=request.user, postname=postname)
        pi.delete()
        stud = Post.objects.filter(postusername=request.user)
        return render(request, 'registration/showPost.html' , {'stu':stud})

@allowed_users(allowed_roles=['admin'])
def update_post(request, id):
    if request.method == 'POST':
        pi = Post.objects.get(pk=id, postusername=request.user)
        fm = addPost(request.POST, instance=pi)
        if fm.is_valid():
            fm.save()
    else:
        pi = Post.objects.get(pk=id, postusername=request.user)
        fm = addPost(instance=pi)
    return render(request, 'registration/updatePost.html', {'form':fm})

def view_profile(request):
    if not UserProfile.objects.filter(username=request.user).exists():
        if request.method=="POST":
            fm=ProfileForm(request.POST,request.FILES)
            if fm.is_valid():
                uname = request.user
                email = uname.email
                bdate = fm.cleaned_data['birthdate']
                mobno = fm.cleaned_data['mobilenno']
                gender = fm.cleaned_data['gender']
                city = fm.cleaned_data['city']
                pincode = fm.cleaned_data['pincode']
                profile = fm.cleaned_data['profile']
                document = fm.cleaned_data['document']
                reg = UserProfile(username=uname,
email=email,birthdate=bdate,mobilenno=mobno,
gender=gender,city=city,pincode=pincode,profile=profile,document=document)
                reg.save()
                fm = ProfileForm()
                return redirect('user_profile')
            else:
                context={'form':fm}
                messages.error(request,'There is Error in your
information...kindly refill the form')
                render(request,'registration/userProfile.html',context)
                fm=ProfileForm()
                return render(request, 'registration/userProfile.html',
{'form':fm})

        stud = UserProfile.objects.filter(username=request.user)
        return render(request, 'registration/showProfile.html' , {'stu':stud})

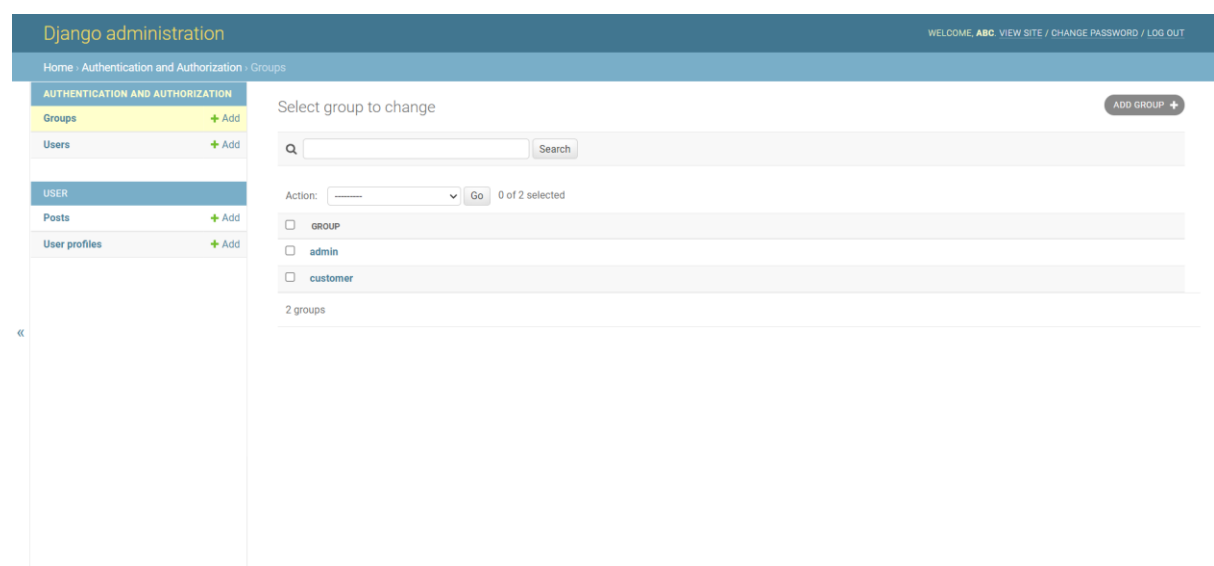
```

```
# @allowed_users(allowed_roles=['admin'])
def home(request):
    return render(request, 'home.html')
```

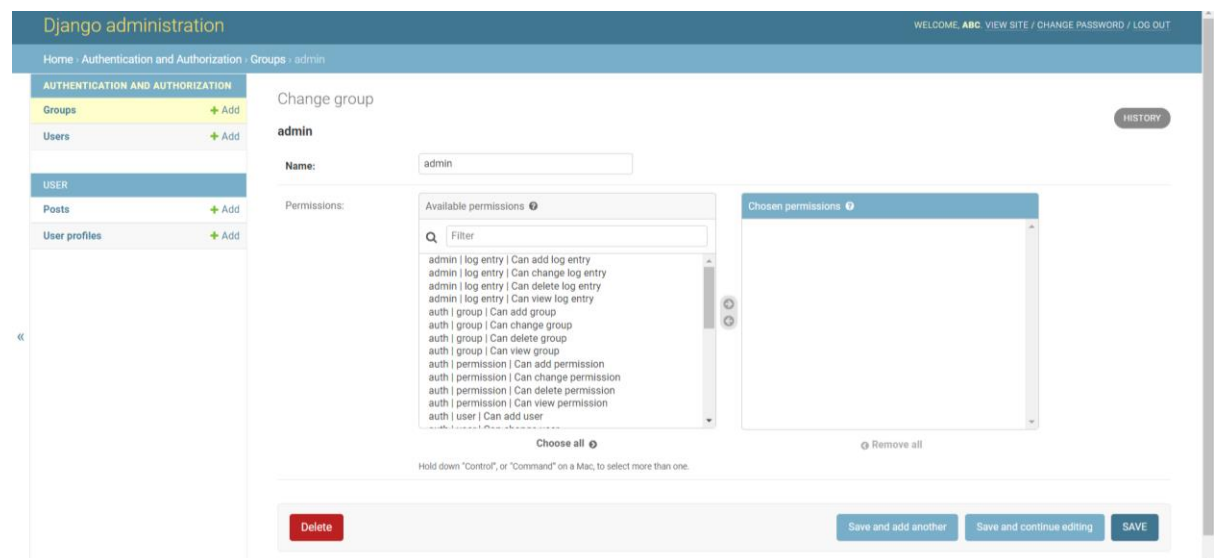
permission.html

```
{% extends 'base.html' %}
{% block title %}Show Post{% endblock title %}
{% block body %}
    <div class="container">
        <h2>Sorry, You are not authorized to view this page </h2>
    </div>
{% endblock body %}
```

Screenshots:



-Admin mode to use Add Post and Post List



AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

USER

Posts + Add

User profiles + Add

EMAIL ADDRESS:

Permissions

☒ Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☒ Staff status
Designates whether the user can log into this admin site.

☐ Superuser status
Designates that this user has all permissions without explicitly assigning them.

Groups:

Available groups ⓘ

customer

Choose all ⓘ

Chosen groups ⓘ +

admin

Remove all ⓘ

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

Available user permissions ⓘ

Chosen user permissions ⓘ

Social Media Site

Add Post

Post list

Profile

Welcome Sanket123 ▾

Add Post Page

Postname:

Description:

Submit

Social Media Site

Add Post

Post list

Profile

Welcome Sanket123 ▾

Post List

Username	Post ID	Post Name	Post Description	Actions
Sanket123	1	Test_Upload_Post_By_Sanket	Testing	<div><div>Edit</div><div>Delete</div></div>

-Customer mode to Restrict to use Add Post and Post List

Django administration

WELCOME, ABC VIEW SITE / CHANGE PASSWORD / LOG OUT

Home / Authentication and Authorization / Groups / customer

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

USER

Posts + Add

User profiles + Add

Change group

customer

customer

Permissions:

Available permissions

Filter

admin | log entry | Can add log entry
admin | log entry | Can change log entry
admin | log entry | Can delete log entry
admin | log entry | Can view log entry
auth | group | Can add group
auth | group | Can change group
auth | group | Can delete group
auth | group | Can view group
auth | permission | Can add permission
auth | permission | Can change permission
auth | permission | Can delete permission
auth | permission | Can view permission
auth | user | Can add user

Choose all

Chosen permissions

user | post | Can add post
user | post | Can change post
user | post | Can delete post
user | post | Can view post

Remove all

Hold down "Control", or "Command" on a Mac, to select more than one.

Delete

Save and add another

Save and continue editing

SAVE

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

USER

Posts + Add

User profiles + Add

Permissions

☒ Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☒ Staff status
Designates whether the user can log into this admin site.

☐ Superuser status
Designates that this user has all permissions without explicitly assigning them.

Groups:

Available groups

Filter

admin

Choose all

Chosen groups

customer

Remove all

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

Available user permissions

Filter

admin | log entry | Can add log entry
admin | log entry | Can change log entry

Chosen user permissions

Sorry, You are not authorized to view this page

Sorry, You are not authorized to view this page

-----Thank You-----