**Source Code**

**urls.py**

from django.conf.urls import url

from django.contrib import admin

from django.urls import path

from django.conf import settings

from django.conf.urls.static import static

from fstapp.views import index

from cloud import views as cloud

from admn import views as admn

from user import views as user

from project7 import views as cloudmonitor

urlpatterns = [

url(r'^admn/', admin.site.urls),

url(r'^index/', index, name="index"),

path(r'userlogin', user.userlogin, name='userlogin'),

path(r'userregister', user.userregister, name='userregister'),

path(r'storeregistration', user.storeregistration, name='storeregistration'),

path(r'userlogincheck', user.userlogincheck, name='userlogincheck'),

path(r'usercreateapp',user.usercreateapp,name='usercreateapp'),

path(r'appcreaterequest',user.appcreaterequest,name='appcreaterequest'),

path(r'useruploadfile/<appname>/$',user.useruploadfile,name='useruploadfile'),

path(r'^snippet\_detail/$',user.snippet\_detail,name='snippet\_detail'),

path(r'clddeletefile',user.clddeletefile,name='clddeletefile'),

path(r'cldresturl',user.cldresturl,name='cldresturl'),

path(r'accounts', user.AccountAPIView.as\_view(), name='account-list'),

path(r'contacts', user.ContactAPIView.as\_view(), name='contact-list'),

path(r'activities', user.ActivityAPIView.as\_view(), name='activity-list'),

path(r'activitystatuses', user.ActivityStatusAPIView.as\_view(), name='activity-status-list'),

path(r'contactsources', user.ContactSourceAPIView.as\_view(), name='contact-source-list'),

path(r'contactstatuses', user.ContactStatusAPIView.as\_view(), name='contact-status-list'),

path(r'logout',user.logout,name='logout'),

url(r'^adminlogin/',admn.adminlogin, name="adminlogin"),

path(r'adminlogincheck',admn.adminlogincheck,name='adminlogincheck'),

path(r'adminactivateusers',admn.adminactivateusers,name='adminactivateusers'),

path(r'activatewaitedusers/<id>/$',admn.activatewaitedusers,name='activatewaitedusers'),

path(r'cloudlogin',cloud.cloudlogin,name='cloudlogin'),

path(r'cloudlogincheck',cloud.cloudlogincheck,name='cloudlogincheck'),

path(r'activateuserapp',cloud. activateuserapp, name='activateuserapp'),

path(r'cloudlogincheck',cloud. cloudlogincheck, name='cloudlogincheck'),

path(r'clouduserappactivations/<appname>/$',cloud. clouduserappactivations, name='clouduserappactivations'),

path(r'uploadfile',cloudmonitor.uploadfile,name='uploadfile'),

path(r'deletefile/<id>',cloudmonitor.deletefile,name='deletefile'),

path(r'downloadfile/<id>',cloudmonitor.downloadfile,name='downloadfile'),

path(r'resturl/<id>',cloudmonitor.resturl,name='resturl'),

]

if settings.DEBUG:

urlpatterns += static(settings.MEDIA\_URL,document\_root=settings.MEDIA\_ROOT)

**views.py:**

from django.shortcuts import render,HttpResponseRedirect,HttpResponse,redirect

from rest\_framework import generics

import json

from .models import Account, Activity, ActivityStatus, Contact, ContactSource, ContactStatus

from .serializers import AccountSerializer, ActivitySerializer, ActivityStatusSerializer, ContactSerializer, ContactSourceSerializer, ContactStatusSerializer,UserFileModelSerializer

from rest\_framework.decorators import api\_view

from rest\_framework import generics

from django.http import JsonResponse

from .models import Account, Activity, ActivityStatus, Contact, ContactSource, ContactStatus

from .serializers import AccountSerializer, ActivitySerializer, ActivityStatusSerializer, ContactSerializer, ContactSourceSerializer, ContactStatusSerializer

from rest\_framework.renderers import TemplateHTMLRenderer

from rest\_framework.response import Response

from rest\_framework.views import APIView

from rest\_framework import serializers

from .forms import CloudUserFrom,UserFileForm

from .models import CloudUsersModel

from django.contrib import messages

from rest\_framework import status

from .models import CloudUsersModel,UserAppCreatModel,UserFileModel

#@api\_view(['GET', 'POST'])

class AccountAPIView(generics.ListCreateAPIView):

queryset = Account.objects.all()

serializer\_class = AccountSerializer

renderer\_classes = [TemplateHTMLRenderer]

template\_name = 'profile\_list.html'

def get(self, request):

queryset = Account.objects.all()

return Response({'profiles': queryset})

def post(self, request, pk):

print('AM going to Execute atleast once in my life')

profile =get\_object\_or\_404(Account,pk=pk)

serializer = AccountSerializer(profile, data=request.data)

if not serializer.is\_valid():

return Response({'serializer': serializer, 'profile': Account})

serializer.save()

return redirect('account-list')

class ActivityAPIView(generics.ListCreateAPIView):

queryset = Activity.objects.all()

serializer\_class = ActivitySerializer

class ActivityStatusAPIView(generics.ListCreateAPIView):

queryset = ActivityStatus.objects.all()

serializer\_class = ActivitySerializer

class ContactAPIView(generics.ListCreateAPIView):

queryset = Contact.objects.all()

serializer\_class = ContactSerializer

class ContactStatusAPIView(generics.ListCreateAPIView):

queryset = ContactStatus.objects.all()

serializer\_class = ContactSerializer

class ContactSourceAPIView(generics.ListCreateAPIView):

queryset = ContactSource.objects.all()

serializer\_class = ContactSourceSerializer

def index(request):

return render(request,'index.html')

def userlogin(request):

return render(request,'userlogin.html')

def userregister(request):

return render(request,'user/userregister.html')

@api\_view(['GET', 'POST'])

def storeregistration(request):

if request.method == 'POST':

form = CloudUserFrom(request.POST)

if form.is\_valid():

try:

rslt = form.save()

print("Form Result Status ", rslt)

messages.success(request, 'You have been successfully registered')

except:

messages.success(request, 'Email Already Registerd')

return render(request, 'user/userregister.html',{})

else:

print("Invalid form")

else:

form = CloudUserFrom()

return render(request, 'user/userregister.html', {'form': form})

def userlogincheck(request):

if request.method == "POST":

email = request.POST.get('cf-email')

pswd = request.POST.get('cf-password')

print("Email = ", email)

try:

check = CloudUsersModel.objects.get(email=email,password=pswd)

request.session['id'] = check.id

request.session['loggeduser'] = check.name

request.session['email'] = check.email

request.session['role']='user'

status = check.status

if status == "activated":

print("User id At", check.id, status)

return render(request,'user/userpage.html')

else:

messages.success(request, 'Your Account Not at activated')

return render(request, 'userlogin.html')

return render(request, 'userlogin.html')

except:

pass

messages.success(request, 'Invalid Email id and password')

return render(request, 'userlogin.html')

@api\_view(['GET', 'PUT', 'DELETE','POST'])

def snippet\_detail(request):

role = request.session['role']

usremail = request.session['email']

try:

snippet = UserFileModel.objects.filter(email=usremail)

#print('Type is ',snippet.id)

except CloudUsersModel.DoesNotExist:

return Response(status=status.HTTP\_404\_NOT\_FOUND)

if request.method == 'GET':

if role == 'user':

print('Get Method Works Fine')

usremail = request.session['email']

queryset = UserFileModel.objects.filter(email=usremail)

serializer\_class = UserFileModelSerializer

print('Return Type is ',serializer\_class)

return render(request,'user/uploadedfiles.html',{'objects':queryset})

elif role=='admin':

queryset = UserFileModel.objects.all()

serializer\_class = UserFileModelSerializer

return render(request, 'admin/adminuploadedfiles.html', {'objects': queryset})

elif role =='cloud':

queryset = UserFileModel.objects.all()

serializer\_class = UserFileModelSerializer

return render(request, 'cloud/clouduploadedfiles.html', {'objects': queryset})

elif request.method == 'PUT':

print('PUT Method Works Fine')

serializer = UserFileModelSerializer(snippet, data=request.data)

if serializer.is\_valid():

serializer.save()

return Response(serializer.data)

return Response(serializer.errors, status=status.HTTP\_400\_BAD\_REQUEST)

elif request.method == 'DELETE':

print('DELETE Method Works Fine')

#snippet.delete()

return Response(status=status.HTTP\_204\_NO\_CONTENT)

elif request.method=="POST":

print('POST Method Works Fine')

form = UserFileForm(request.POST, request.FILES)

print("form",form)

if form.is\_valid():

form.save()

print("data saved:")

else:

print('Invalid Form')

return Response(status=status.HTTP\_201\_CREATED)

def logout(request):

request.session.modified = True

return render(request,'index.html',{})

def usercreateapp(request):

usremail = request.session['email']

dict = UserAppCreatModel.objects.filter(email=usremail)

return render(request,'user/userappcreations.html',{'objects':dict})

def appcreaterequest(request):

if request.method=='POST':

usrname = request.POST.get('usrname')

usremail = request.POST.get('usremail')

appname = request.POST.get('appname')

accesskey = request.POST.get('accesskey')

secretkey = request.POST.get('secretkey')

try:

UserAppCreatModel.objects.create(name=usrname,email=usremail,appname=appname)

messages.success(request, 'Your App creation Request is Under Process')

except:

messages.success(request, 'App Name Already exist')

pass

print(usrname,usremail,appname,accesskey,secretkey)

dict = UserAppCreatModel.objects.filter(email=usremail)

return render(request,'user/userappcreations.html',{'objects':dict})

def useruploadfile(request,appname):

check = UserAppCreatModel.objects.get(appname=appname)

acckey = check.accesskey

secretkey = check.secretkey

dict = {'appname':appname,'acckey':acckey,'secretkey':secretkey}

return render(request,'user/uploaddatatocloud.html',dict)

def cldresturl(request):

return render(request,'cloud/edit.html')

def clddeletefile(request):

return render(request,'cloud/dlt.html')

**models.py:**

from django.db import models

from django.contrib.auth.models import User

import os

INDCHOICES = (

('FINANCE', 'FINANCE'),

('HEALTHCARE', 'HEALTHCARE'),

('INSURANCE', 'INSURANCE'),

('LEGAL', 'LEGAL'),

('MANUFACTURING', 'MANUFACTURING'),

('PUBLISHING', 'PUBLISHING'),

('REAL ESTATE', 'REAL ESTATE'),

('SOFTWARE', 'SOFTWARE'),

)

class Account(models.Model):

name = models.CharField("Name of Account", "name", max\_length=64)

email = models.EmailField(blank = True, null = True)

phone = models.CharField(max\_length=20, blank = True, null = True)

industry = models.CharField("Industry Type", max\_length=255, choices=INDCHOICES, blank=True, null=True)

website = models.URLField("Website", blank=True, null=True)

description = models.TextField(blank=True, null=True)

createdBy = models.ForeignKey(User, related\_name='account\_created\_by', on\_delete=models.CASCADE)

createdAt = models.DateTimeField("Created At", auto\_now\_add=True)

isActive = models.BooleanField(default=False)

def \_\_str\_\_(self):

return self.name

class ContactSource(models.Model):

status = models.CharField("Contact Source", max\_length=20)

def \_\_str\_\_(self):

return self.status

class ContactStatus(models.Model):

status = models.CharField("Contact Status", max\_length=20)

def \_\_str\_\_(self):

return self.status

class Contact(models.Model):

first\_name = models.CharField("First name", max\_length=255, blank = True, null = True)

last\_name = models.CharField("Last name", max\_length=255, blank = True, null = True)

account = models.ForeignKey(Account, related\_name='lead\_account\_contacts', on\_delete=models.CASCADE, blank=True, null=True)

email = models.EmailField()

phone = models.CharField(max\_length=20, blank = True, null = True)

address = models.TextField(blank=True, null=True)

description = models.TextField(blank=True, null=True)

createdBy = models.ForeignKey(User, related\_name='contact\_created\_by', on\_delete=models.CASCADE)

createdAt = models.DateTimeField("Created At", auto\_now\_add=True)

isActive = models.BooleanField(default=False)

def \_\_str\_\_(self):

return self.first\_name

class ActivityStatus(models.Model):

status = models.CharField("Activity Status", max\_length=20)

def \_\_str\_\_(self):

return self.status

class Activity(models.Model):

description = models.TextField(blank=True, null=True)

createdAt = models.DateTimeField("Created At", auto\_now\_add=True)

contact = models.ForeignKey(Contact, on\_delete=models.CASCADE, blank=True, null=True)

def \_\_str\_\_(self):

return self.description

class UserAppCreatModel(models.Model):

id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=200)

email = models.CharField(max\_length=200)

appname = models.CharField(max\_length=200,unique=True)

accesskey = models.CharField(max\_length=200,default='waiting')

secretkey = models.CharField(max\_length=200,default='waiting')

def \_\_str\_\_(self):

return self.appname

class Meta:

db\_table = "userapps"

class CloudUsersModel(models.Model):

id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=200)

email = models.CharField(max\_length=100,unique=True)

password = models.CharField(max\_length=100)

mobile = models.CharField(max\_length=100)

state = models.CharField(max\_length=100)

status = models.CharField(max\_length=100,default='waiting')

def \_\_str\_\_(self):

return self.email

class Meta:

db\_table = "userregistrations"

class UserFileModel(models.Model):

id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=200)

email = models.CharField(max\_length=200)

appname = models.CharField(max\_length=200)

accesskey = models.CharField(max\_length=200)

secretkey = models.CharField(max\_length=200)

filename = models.CharField(max\_length=200)

userfile = models.FileField(upload\_to='media/')

def \_\_str\_\_(self):

return os.path.basename(self.userfile.name)

class Meta:

db\_table = "userfiles"

def delete(self, \*args, \*\*kwargs):

self.userfile.delete()

super().delete(\*args, \*\*kwargs)

**forms.py:**

from django import forms

from .models import CloudUsersModel,UserFileModel

class CloudUserFrom(forms.ModelForm):

name = forms.CharField(widget=forms.TextInput(attrs={'size':50,'class': 'special'}), required=True,max\_length=100 )

email = forms.CharField(widget=forms.TextInput(attrs={'size':50}), required=True, max\_length=100)

password = forms.CharField(widget=forms.PasswordInput(attrs={'size':50}), required=True,max\_length=100)

mobile = forms.CharField(widget=forms.TextInput(attrs={'size':50}), required=True,max\_length=100)

status = forms.CharField(widget=forms.HiddenInput(), initial='waiting', max\_length=100)

class Meta():

model = CloudUsersModel

fields=['name','email','password','mobile','status']

class UserFileForm(forms.ModelForm):

name = forms.CharField(max\_length=100 )

class Meta():

model = UserFileModel

fields = '\_\_all\_\_'

**adminbase.html:**

{% load static %}

<!doctype html>

<!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-ie7" lang=""> <![endif]-->

<!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8" lang=""> <![endif]-->

<!--[if IE 8]> <html class="no-js lt-ie9" lang=""> <![endif]-->

<!--[if gt IE 8]><!-->

<html class="no-js" lang="">

<!--<![endif]-->

<head>

<meta charset="utf-8">

<meta name="description" content="">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Minifolio - Bootstrap Responsive Resume, Personal Portfolio Template</title>

<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">

<link rel="stylesheet" href="{% static 'css/jquery.fancybox.css' %}">

<link rel="stylesheet" href="{% static 'css/main.css' %}">

<link rel="stylesheet" href="{% static 'css/responsive.css' %}">

<link rel="stylesheet" href="{% static 'css/animate.min.css' %}">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">

</head>

<body>

<!-- header section -->

<section class="banner" role="banner">

<header id="header">

<!-- navigation section -->

<div class="header-content clearfix"> <!--<a class="logo" href="#"><img src="{% static 'images/logo.png' %}" alt="Mafolio"></a>-->

<nav class="navigation" role="navigation">

<ul class="primary-nav">

<li><a href="#">Home</a></li>

<li><a href="{% url 'adminactivateusers'%}">Users</a></li>

<li><a href="{%url 'snippet\_detail'%}">View Files</a></li>

<li><a href="{%url 'logout'%}">Log Out</a> </li>

</ul>

</nav>

<!--<a href="#" class="nav-toggle">Menu<span></span></a>--> </div>

<!-- navigation section -->

</header>

<!-- banner text -->

<div class="container">

<div class="col-md-10 col-md-offset-1">

<div class="banner-text text-center">

<h2>Checking Security Properties of Cloud Service REST APIs</h2><br><br><br>

<!--<p>Checking Security Properties of Cloud Service REST APIs.</p>-->

{% block contents %}

{% endblock %}

<!-- banner text -->

</div>

</div>

</div>

</section>

<!-- header section -->

<!-- description text section -->

<!--<section id="descripton" class="section descripton">

<div class="container">

<div class="col-md-10 col-md-offset-1 text-center">

<p>Donec sed odio dui. Nulla vitae elit libero, a pharetra augue. Nullam id dolor id nibh ultricies vehicula ut id elit. Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Aenean lacinia bibendum nulla sed consectetur. Cras mattis consectetur purus sit amet avida at eget metus.</p>

</div>

</div>

</section>

<section id="works" class="works section no-padding">

<div class="container-fluid">

<div class="row no-gutter">

<div class="col-lg-3 col-md-6 col-sm-6 work"> <a href="images/work-1.jpg" class="work-box"> <img src="images/work-1.jpg" alt="">

<div class="overlay">

<div class="overlay-caption">

<h5>Project Name</h5>

<p>Logo Design</p>

</div>

</div>

</a> </div>

<div class="col-lg-3 col-md-6 col-sm-6 work"> <a href="images/work-2.jpg" class="work-box"> <img src="images/work-2.jpg" alt="">

<div class="overlay">

<div class="overlay-caption">

<h5>Project Name</h5>

<p>Website Design</p>

</div>

</div>

</a> </div>

<div class="col-lg-3 col-md-6 col-sm-6 work"> <a href="images/work-3.jpg" class="work-box"> <img src="images/work-3.jpg" alt="">

<div class="overlay">

<div class="overlay-caption">

<h5>Project Name</h5>

<p>Branding</p>

</div>

</div>

</a> </div>

<div class="col-lg-3 col-md-6 col-sm-6 work"> <a href="images/work-4.jpg" class="work-box"> <img src="images/work-4.jpg" alt="">

<div class="overlay">

<div class="overlay-caption">

<h5>Project Name</h5>

<p>Graphic Design</p>

</div>

</div>

</a> </div>

<div class="col-lg-3 col-md-6 col-sm-6 work"> <a href="images/work-5.jpg" class="work-box"> <img src="images/work-5.jpg" alt="">

<div class="overlay">

<div class="overlay-caption">

<h5>Project Name</h5>

<p>Website Design</p>

</div>

</div>

</a> </div>

<div class="col-lg-3 col-md-6 col-sm-6 work"> <a href="images/work-6.jpg" class="work-box"> <img src="images/work-6.jpg" alt="">

<div class="overlay">

<div class="overlay-caption">

<h5>Project Name</h5>

<p>Logo Design</p>

</div>

</div>

</a> </div>

<div class="col-lg-3 col-md-6 col-sm-6 work"> <a href="images/work-7.jpg" class="work-box"> <img src="images/work-7.jpg" alt="">

<div class="overlay">

<div class="overlay-caption">

<h5>Project Name</h5>

<p>Branding</p>

</div>

</div>

</a> </div>

<div class="col-lg-3 col-md-6 col-sm-6 work"> <a href="images/work-8.jpg" class="work-box"> <img src="images/work-8.jpg" alt="">

<div class="overlay">

<div class="overlay-caption">

<h5>Project Name</h5>

<p>Website Design</p>

</div>

</div>

</a> </div>

</div>

</div>

</section>

<section id="hireme" class="section hireme">

<div class="container">

<div class="col-md-8 col-md-offset-2 text-center">

<h3>Need something specific?</h3>

<p>We are currently crafting new products but would love to hear from you.</p>

<a href="contact.html" class="btn btn-large">Hire me</a> </div>

</div>

</section>-->

<!-- hire me section -->

<!-- footer -->

<!--<footer class="section footer">-->

<!-- <div class="footer-bottom">-->

<!-- <div class="container">-->

<!-- <div class="col-md-12">-->

<!-- <p>-->

<!-- &lt;!&ndash;<ul class="footer-share">-->

<!-- <li><a href="#"><i class="fa fa-facebook"></i></a></li>-->

<!-- <li><a href="#"><i class="fa fa-twitter"></i></a></li>-->

<!-- <li><a href="#"><i class="fa fa-linkedin"></i></a></li>-->

<!-- <li><a href="#"><i class="fa fa-google-plus"></i></a></li>-->

<!-- <li><a href="#"><i class="fa fa-pinterest-p"></i></a></li>-->

<!-- <li><a href="#"><i class="fa fa-vimeo"></i></a></li>-->

<!-- </ul>&ndash;&gt;-->

<!-- </p>-->

<!-- <p>© 2020 All rights reserved. All Rights Reserved<br>-->

<!-- Designed by Sagar Marri</a></p>-->

<!-- </div>-->

<!-- </div>-->

<!-- </div>-->

<!--</footer>&ndash;&gt;-->

<!-- footer -->

<!-- JS FILES -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

<script src="{% static 'js/bootstrap.min.js' %}"></script>

<script src="{% static 'js/jquery.fancybox.pack.js' %}"></script>

<script src="{% static 'js/jquery.waypoints.min.js' %}"></script>

<script src="{% static 'js/retina.min.js' %}"></script>

<script src="{% static 'js/modernizr.js' %}"></script>

<script src="{% static 'js/main.js' %}"></script>

</body>

</html>