m:\My-Documents\programming\java\source\Python Django Practise\mysite\account  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*admin.py\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
from django.contrib import admin  
from .models import UserProfile  
  
class UserProfileAdmin(admin.ModelAdmin):  
 list\_display = ('user', 'birth', 'phone')  
 list\_filter = ("phone",)  
  
admin.site.register(UserProfile, UserProfileAdmin)  
  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*apps.py\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
from django.apps import AppConfig  
  
  
class AccountConfig(AppConfig):  
 name = 'account'  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*forms.py\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
from django import forms  
from django.contrib.auth.models import User  
from .models import UserProfile, UserInfo  
  
class LoginForm(forms.Form):  
 username = forms.CharField()  
 password = forms.CharField(widget=forms.PasswordInput)  
  
class RegistrationForm(forms.ModelForm):  
 password = forms.CharField(label="Password", widget=forms.PasswordInput)  
 password2 = forms.CharField(label="Confirm Pssword", widget=forms.PasswordInput)  
  
 class Meta:  
 model = User   
 fields = ("username", "email")  
  
 def clean\_password2(self):   
 cd = self.cleaned\_data  
 if cd['password'] != cd['password2']:  
 raise forms.ValidationError("passwords do not match.")  
 return cd['password2']  
  
  
class UserProfileForm(forms.ModelForm):  
 class Meta:  
 model = UserProfile  
 fields = ("phone", "birth")  
  
class UserInfoForm(forms.ModelForm):  
 class Meta:  
 model = UserInfo  
 fields = ("school", "company", "profession", "address", "aboutme", "photo")  
  
class UserForm(forms.ModelForm):  
 class Meta:  
 model = User  
 fields = ("email",)   
  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*models.py\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
from django.db import models  
from django.contrib.auth.models import User  
  
class UserProfile(models.Model): #①  
 user = models.OneToOneField(User, unique=True) #②  
 birth = models.DateField(blank=True, null=True)  
 phone = models.CharField(max\_length=20, null=True)  
   
 def \_\_str\_\_(self):  
 return 'user {}'.format(self.user.username)  
  
class UserInfo(models.Model):  
 user = models.OneToOneField(User, unique=True)  
 school = models.CharField(max\_length=97, blank=True)  
 company = models.CharField(max\_length=97, blank=True)  
 profession = models.CharField(max\_length=27, blank=True)  
 address = models.CharField(max\_length=177, blank=True)  
 aboutme = models.TextField(blank=True)   
 photo = models.ImageField(blank=True)  
  
 def \_\_str\_\_(self):  
 return "user:{}".format(self.user.username)  
  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*tests.py\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
from django.test import TestCase  
  
# Create your tests here.  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*urls.py\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
from django.conf.urls import url  
from . import views  
from django.contrib.auth import views as auth\_views  
  
urlpatterns = [  
 #url(r'^login/$', views.user\_login, name='user\_login'),  
 url(r"^login/$", auth\_views.login, name="user\_login"),  
 url(r"^new-login/$", auth\_views.login, {"template\_name": "account/login.html"}),  
 url(r'^logout/$', auth\_views.logout, {"template\_name": "account/logout.html"}, name='user\_logout'),  
 url(r'^register/$', views.register, name="user\_register"),  
 url(r'^password-change/$', auth\_views.password\_change, {"post\_change\_redirect":"account/password-change-done"}, name='password\_change'),  
 url(r'^password-change-done/$', auth\_views.password\_change\_done, name='password\_change\_done'),  
 url(r'^password-reset/$', auth\_views.password\_reset, {"template\_name":"account/password\_reset\_form.html", "email\_template\_name":"account/password\_reset\_email.html", "post\_reset\_redirect":"/account/password-reset-done"}, name="password\_reset"),  
 url(r'^password-reset-done/$', auth\_views.password\_reset\_done, {"template\_name":"account/password\_reset\_done.html"}, name="password\_reset\_done"),  
 url(r'^password-reset-confirm/(?P<uidb64>[-\w]+)/(?P<token>[-\w]+)/$', auth\_views.password\_reset\_confirm, {"template\_name":"account/password\_reset\_confirm.html", "post\_reset\_redirect":"/account/password-reset-complete"}, name="password\_reset\_confirm"),  
 url(r'^password-reset-complete/$', auth\_views.password\_reset\_complete, {"template\_name":"account/password\_reset\_complete.html"}, name="password\_reset\_complete"),  
 url(r'^my-information/$', views.myself, name="my\_information"),  
 url(r'^edit-my-information/$', views.myself\_edit, name="edit\_my\_information"),  
 url(r'^my-image/$', views.my\_image, name="my\_image"),  
]  
  
#"subject\_template\_name":"account/password\_reset\_subject.txt",   
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*views.py\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
from django.shortcuts import render  
from django.http import HttpResponse, HttpResponseRedirect  
from django.contrib.auth import authenticate, login  
from django.contrib.auth.decorators import login\_required  
from .forms import LoginForm, RegistrationForm, UserProfileForm, UserInfoForm, UserForm  
from .models import UserProfile, UserInfo  
from django.contrib.auth.models import User  
from django.core.urlresolvers import reverse  
  
def user\_login(request):  
 if request.method == "POST":  
 login\_form = LoginForm(request.POST)  
 if login\_form.is\_valid():  
 cd = login\_form.cleaned\_data  
 user = authenticate(username=cd['username'], password=cd['password'])  
 if user:  
 login(request,user)  
 return HttpResponse("Wellcome You. You have been authenticated successfully.")  
 else:  
 return HttpResponse("Sorry, Your username or password is not right.")  
 else:  
 return HttpResponse("Invalid login")  
  
 if request.method == "GET":  
 login\_form = LoginForm()  
 return render(request, "account/login.html", {"form":login\_form})  
  
#def register(request):  
# if request.method == "POST":  
# user\_form = RegistrationForm(request.POST)  
# if user\_form.is\_valid():  
# new\_user = user\_form.save(commit=False)   
# new\_user.set\_password(user\_form.cleaned\_data['password'])   
# new\_user.save()  
# return HttpResponse("successfully")  
# else:  
# return HttpResponse("sorry, your can not register.")  
# else:  
# user\_form = RegistrationForm()  
# return render(request, "account/register.html", {"form": user\_form})  
  
def register(request):  
 if request.method == "POST":  
 user\_form = RegistrationForm(request.POST)  
 userprofile\_form = UserProfileForm(request.POST)  
 if user\_form.is\_valid()\*userprofile\_form.is\_valid():  
 new\_user = user\_form.save(commit=False)  
 new\_user.set\_password(user\_form.cleaned\_data['password'])  
 new\_user.save()  
 new\_profile = userprofile\_form.save(commit=False)  
 new\_profile.user = new\_user  
 new\_profile.save()  
 UserInfo.objects.create(user=new\_user)  
 #return HttpResponse("successfully")  
 return HttpResponseRedirect(reverse("account:user\_login"))  
 else:  
 return HttpResponse("sorry, your can not register.")  
 else:  
 user\_form = RegistrationForm()  
 userprofile\_form = UserProfileForm()  
 return render(request, "account/register.html", {"form": user\_form, "profile":userprofile\_form})  
  
@login\_required(login\_url='/account/login/')   
def myself(request):  
 user = User.objects.get(username=request.user.username)  
 userprofile = UserProfile.objects.get(user=user)  
 userinfo = UserInfo.objects.get(user=user)  
 return render(request, "account/myself.html", {"user":user, "userinfo":userinfo, "userprofile":userprofile})  
  
@login\_required(login\_url='/account/login/')  
def myself\_edit(request):  
 user = User.objects.get(username=request.user.username)   
 userprofile = UserProfile.objects.get(user=request.user)  
 userinfo = UserInfo.objects.get(user=request.user)  
  
 if request.method == "POST":  
 user\_form = UserForm(request.POST)  
 userprofile\_form = UserProfileForm(request.POST)  
 userinfo\_form = UserInfoForm(request.POST)  
 if user\_form.is\_valid() \* userprofile\_form.is\_valid() \* userinfo\_form.is\_valid():  
 user\_cd = user\_form.cleaned\_data  
 userprofile\_cd = userprofile\_form.cleaned\_data  
 userinfo\_cd = userinfo\_form.cleaned\_data  
 user.email = user\_cd['email']  
 userprofile.birth = userprofile\_cd['birth']  
 userprofile.phone = userprofile\_cd['phone']  
 userinfo.school = userinfo\_cd['school']  
 userinfo.company = userinfo\_cd['company']  
 userinfo.profession = userinfo\_cd['profession']  
 userinfo.address = userinfo\_cd['address']  
 userinfo.aboutme = userinfo\_cd['aboutme']  
 user.save()  
 userprofile.save()  
 userinfo.save()  
 return HttpResponseRedirect('/account/my-information/')   
 else:  
 user\_form = UserForm(instance=request.user)  
 userprofile\_form = UserProfileForm(initial={"birth":userprofile.birth, "phone":userprofile.phone})  
 userinfo\_form = UserInfoForm(initial={"school":userinfo.school, "company":userinfo.company, "profession":userinfo.profession, "address":userinfo.address, "aboutme":userinfo.aboutme})  
 return render(request, "account/myself\_edit.html", {"user\_form":user\_form, "userprofile\_form":userprofile\_form, "userinfo\_form":userinfo\_form})  
  
@login\_required(login\_url='/account/login/')  
def my\_image(request):  
 if request.method == 'POST':  
 img = request.POST['img']  
 userinfo = UserInfo.objects.get(user=request.user.id)  
 userinfo.photo = img  
 userinfo.save()  
 return HttpResponse("1")  
 else:  
 return render(request, 'account/imagecrop.html',)