**Accounting app**

**Sign Up**

**First Way:**

**In urls.py**

path('signup', views.signup, name='signup')

**In views.py**

from django.contrib.auth.forms import UserCreationForm

def signup(request):

form = UserCreationForm()

if request.method == 'POST':

form = UserCreationForm(request.POST)

if form.is\_valid():

user = form.save(commit=False)

login(request, user)

return redirect('home')

return render(request, 'signup.html',{'form':form})

**In signup.html**

<h1> sign up</h1>

<form method='POST'>

{%csrf\_token %}

{{form.as\_p}}

<button type='submit' > submit</button>

</form>

**In home.html**

{% if user.is\_authenticated%}

<p> welcome {{user.username}} </p>

{%endif%}

**Customizations in Sign Up**

**Second Way:**

اذا اردنا اضافة حقول معينة الى فورم تسجيل مستخدم جديد مثل الايميل نقوم بالخطوات التالية:

**In form.py**

from django.contrib.auth.forms import UserCreationForm

class SignUpForm(UserCreationForm):

email = forms.CharField(widget=forms.EmailInput())

class Meta:

model= User

fields = [' username' ,'password1',' password2', 'email']

**In views.py**

def signup(request):

form = SignUpForm()

if request.method == 'POST':

form = SignUpForm(request.POST)

if form.is\_valid():

user = form.save(commit=False)

login(request, user)

return redirect('home')

else:

form = SignUpForm()

return render(request, 'signup.html',{'form':form})

**Third Way:**

اذا اردنا اضافة حقول جديدة تحتوي على بيانات جديدة مثل العنوان و الصورة الشخصية و تاريخ الميلاد و رقم الهاتف نقوم باتباع الخطوات التالية: (هنا نضيف حقول اضافية الى مودل المستخدم)

**In models.py**

from django.db import models

from django.contrib.auth.models import AbstractUser

class CustomUser(AbstractUser):

GENDER\_CHOICES = [

('M', 'Male'),

('F', 'Female'),

('O', 'Other'),]

## add profile image for user

profile\_image = models.ImageField(upload\_to='profile\_images/', blank=True, null=True)

gender = models.CharField(choices=GENDER\_CHOICES, blank=True, null=True)

birth\_date = models.DateField(null=True, blank=True)

phone\_number = models.CharField(blank=True, null=True)

def \_str\_(self):

return self.username

**In settings.py**

نقوم بتحديث اعدادات المشروع لاستخدام مودل المستخدم المخصص

AUTH\_USER\_MODEL = 'accounts.CustomUser'

**In form.py**

from django import forms

from .models import CustomUser

class SignupForm(forms.ModelForm):

password = forms.CharField(widget=forms.PasswordInput)

class Meta:

model = CustomUser

fields = ['username', 'email', 'first\_name', 'last\_name', 'profile\_image', 'gender', 'birth\_date', 'phone\_number', 'password']

def save(self, commit=True):

user = super().save(commit=False)

user.set\_password(self.cleaned\_data['password']) ## لتشفير كلمة المرور

if commit:

user.save()

return user

**In views.py**

from django.shortcuts import render, redirect

from .forms import SignupForm

from django.contrib.auth import login

def signup(request):

if request.method == 'POST':

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

if form.is\_valid():

user = form.save() # حفظ المستخدم الجديد

login(request, user) # تسجيل الدخول تلقائيًا بعد التسجيل

return redirect('home') # توجيه المستخدم إلى الصفحة الرئيسية أو أي صفحة أخرى

else:

form = SignupForm()

return render(request, 'accounts/signup.html', {'form': form})

**In signup.html**

enctype="multipart/form-data": ضرورية لتحميل الملفات مثل الصور و حتى يتم ارسال الملفات بشكل صحيح الى الخادم بدونها لن يتمكن الخادم من استقبال البيانات و معالجتها بشكل صحيح

<h1> sign up</h1>

<form method="POST" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Sign Up</button>

</form>

**Inurls.py**

path('signup/', views.signup, name='signup')]

**Logout**

**First Way:**

**In urls.py**

from django.contrib.auth import views as auth\_views

path('logout/', auth\_views.LogoutView.as\_view(), name='logout'),

**In settings.py**

LOGOUT\_REDIRECT\_URL = 'home'

**In home.html**

<a href="{% url 'accounts:logout' %}">Logout</a>

**Customization in Logout:**

**Second Way:**

**In views.py**

from django.contrib.auth.views import LogoutView

## LogoutView يمكننا هذه الدالة من عمل تخصيص و عمل تحويل مباشر الى صفحة معينة عند تسجيل الخروج

from django.urls import reverse\_lazy

class CustomLogoutView(LogoutView):

next\_page = reverse\_lazy('home')

**In urls.py**

from django.urls import path

from .views import CustomLogoutView

urlpatterns= [

path('logout/', CustomLogoutView.as\_view(), name='logout'),]

**In home.html**

{% if user.is\_authenticated %}

<a href="{% url 'logout' %}">Logout</a>

{% else %}

<a href="{% url 'login' %}">Login</a>

{% endif %}

**Third Way:**

**In urls.py**

path( 'logout', views.logout\_user, name='logout')

**In views.py**

def logout\_user(request):

logout(request)

return redirect('home')

**In home.html**

{% if user.is\_authenticated %}

<a href="{% url 'logout' %}">Logout</a>

{% else %}

<a href="{% url 'login' %}">Login</a>

{% endif %}

**Login**

**First Way:**

**In urls.py**

from django.contrib.auth.views import LoginView

path('login/', LoginView.as\_view(template\_name='login.html'), name='login'),

**In settings.py**

LOGIN\_REDIRECT\_URL = 'home'

**In login.html**

<h1> login</h1>

<form method='POST'>

{%csrf\_token %}

{{form.as\_p}}

<button type='submit' > login</button>

</form>

**In home.html**

{% if user.is\_authenticated %}

<a href="{% url 'logout' %}">Logout</a>

{% else %}

<a href="{% url 'login' %}">Login</a>

{% endif %}

**Second Way:**

**In forms.py**

from django import forms

class LoginForm(forms.Form):

username = forms.CharField(max\_length=150)

password = forms.CharField(widget=forms.PasswordInput())

**In views.py**

from django.shortcuts import render, redirect

from django.contrib.auth import authenticate, login

from .forms import LoginForm

def login\_view(request):

if request.method == 'POST':

form = LoginForm(request.POST)

if form.is\_valid():

username = form.cleaned\_data.get('username')

password = form.cleaned\_data.get('password')

user = authenticate(request, username=username, password=password)

if user is not None:

login(request, user)

return redirect('home')

else:

form = LoginForm()

return render(request, 'login.html', {'form': form})

**In urls.py**

from django.urls import path

from .views import login\_view

path('login/', login\_view, name='login'),

**In login.html**

<h1> login</h1>

<form method='POST'>

{%csrf\_token %}

{{form.username}}

{{form.password}}

<button type='submit' > login</button>

</form>

**In settings.py**

LOGIN\_REDIRECT\_URL = 'home'

**In home.html**

{% if user.is\_authenticated %}

<a href="{% url 'logout' %}">Logout</a>

{% else %}

<a href="{% url 'login' %}">Login</a>

{% endif %}

**Third Way:**

تسجيل الدخول عن طريق الايميل و كلمة المرور

**Change password And Change password done**

**First Way:**

**In urls.py**

from django.contrib.auth import views as auth\_views

path('settings/change\_password/', auth\_views.PasswordChangeView.as\_view(template\_name='change\_password.html',success\_url='/accounts/settings/change\_password/done/') , name='password\_change'),

path('settings/change\_password/done/', auth\_views.PasswordChangeDoneView.as\_view(template\_name= 'change\_password\_done.html') name='password\_change\_done'),

**In change\_password.html**

<h1> change password</h1>

<form method='POST'>

{%csrf\_token %}

{{form.as\_p}}

<button type='submit' > update </button>

</form>

**In change\_password\_done.html**

<div class="alert alert-success" role="alert">

<strong>Success!</strong> Your password has been changed!

</div>

<a href="{% url 'home' %}">Return to home page</a>

**Second Way:**

**In forms.py**

from django import forms

from django.contrib.auth.forms import PasswordChangeForm

class CustomPasswordChangeForm(PasswordChangeForm):

old\_password = forms.CharField(label="كلمة المرور القديمة", widget=forms.PasswordInput(attrs={'class': 'form-control'}),)

new\_password1 = forms.CharField(label="كلمة المرور الجديدة", widget=forms.PasswordInput(attrs={'class': 'form-control'}), help\_text="كلمة المرور يجب أن تكون أكثر من 8 أحرف.",)

new\_password2 = forms.CharField(label="تأكيد كلمة المرور الجديدة",

widget=forms.PasswordInput(attrs={'class': 'form-control'}),)

def clean\_new\_password1(self):

new\_password1 = self.cleaned\_data.get('new\_password1')

if len(new\_password1) < 8:

raise forms.ValidationError("كلمة المرور يجب أن تكون أكثر من 8 أحرف.")

return new\_password1

**In views.py**

from django.contrib.auth import update\_session\_auth\_hash

from django.contrib.auth.decorators import login\_required

from .forms import CustomPasswordChangeForm

from django.contrib import messages

@login\_required

def change\_password(request):

if request.method == 'POST':

form = CustomPasswordChangeForm(request.user, request.POST)

if form.is\_valid():

user = form.save()

لتحديث جلسة المستخدم بعد تغيير كلمة المرور ###

update\_session\_auth\_hash(request, user) # Important!

messages.success(request, 'تم تغيير كلمة المرور بنجاح!')

return redirect('password\_change\_done')

else:

form = CustomPasswordChangeForm(request.user)

return render(request, 'change\_password.html', {'form': form})

@login\_required

def password\_change\_done(request):

return render(request, 'password\_change\_done.html')

**In change\_password.html**

{% extends 'base.html' %}

{% block content %}

<h2> Change Password</h2>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit"> Update</button>

</form>

{% endblock %}

**In password\_change\_done.html**

{% extends 'base.html' %}

{% block content %}

<h2>password changed successfully done!</h2>

<p>Your password has been changed successfully.</p>

{% endblock %}

**In base.html**

{% if messages %}

<div class="messages">

{% for message in messages %}

### message.tags تستخدم لتحديد نوع الرسالة سواء رسالة نجاح او خطا او تحذير

<div class="alert {% if message.tags %}alert-{{ message.tags }}{% endif %}">

{{ message }}

</div>

{% endfor %}

</div>

{% endif %}

{% block content %}

############

{% endblock %}

**In urls.py**

from django.urls import path

from . import views

urlpatterns = [

path('change-password/', views.change\_password, name='change\_password'),

path('change-password/done/', views.password\_change\_done, name='password\_change\_done'),

]

**Update profile**

**First Way:**

**In forms.py**

from django import forms

from django.contrib.auth.models import User

class UpdateProfileForm(forms.ModelForm):

password = forms.CharField(widget=forms.PasswordInput, required=False)

class Meta:

model = User

fields = ['username', 'email', 'first\_name', 'last\_name', 'password']

def save(self, commit=True):

user = super().save(commit=False)

if self.cleaned\_data['password']: # إذا تم إدخال كلمة مرور جديدة

user.set\_password(self.cleaned\_data['password'])

if commit:

user.save()

return user

**In views.py**

from django.shortcuts import render, redirect

from .forms import UpdateProfileForm

from django.contrib.auth.decorators import login\_required

@login\_required

def update\_profile(request):

user = request.user

if request.method == 'POST':

form = UpdateProfileForm(request.POST)

if form.is\_valid():

form.save() # حفظ التحديثات

return redirect('profile') #بعد حفظ التحديثات، يمكنك توجيه المستخدم إلى صفحة الملف الشخصي أو أي صفحة أخرى

else:

form = UpdateProfileForm(instance=user)

return render(request, 'accounts/update\_profile.html', {'form': form})

**In update\_profile.html**

<h2>Update Your Profile</h2>

<form method="POST">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Update Profile</button>

</form>

**In urls.py**

path('update\_profile/', views.update\_profile, name='update\_profile'),

**Customization in Update profile**

**Second Way:**

لتحديث بيانات الملف الشخصي مثل الاسم الاول و الاسم الاخير و الايميل و الصورة الشخصية و الجنس و كلمة المرور و تاريخ الميلاد و رقم الهاتف و العنوان

**In models.py**

from django.db import models

from django.contrib.auth.models import AbstractUser

class CustomUser(AbstractUser):

GENDER\_CHOICES = [('M', 'Male'),('F', 'Female'),('O', 'Other'),]

profile\_image = models.ImageField(upload\_to='profile\_images/', blank=True, null=True)

gender = models.CharField(max\_length=1, choices=GENDER\_CHOICES, blank=True, null=True)

birth\_date = models.DateField(null=True, blank=True)

phone\_number = models.CharField(max\_length=15, blank=True, null=True)

address = models.TextField(blank=True, null=True) # إضافة حقل العنوان

def \_str\_(self):

return self.username

**In settings.py**

AUTH\_USER\_MODEL = 'accounts.CustomUser'

**In forms.py**

from django import forms

from .models import CustomUser

class UpdateProfileForm(forms.ModelForm):

password = forms.CharField(widget=forms.PasswordInput, required=False)

class Meta:

model = CustomUser

fields = ['username', 'email', 'first\_name', 'last\_name', 'profile\_image', 'gender', 'birth\_date', 'phone\_number', 'address', 'password']

def save(self, commit=True):

user = super().save(commit=False)

if self.cleaned\_data['password']: # إذا تم إدخال كلمة مرور جديدة

user.set\_password(self.cleaned\_data['password'])

if commit:

user.save()

return user

**In views.py**

from django.shortcuts import render, redirect

from .forms import UpdateProfileForm

from django.contrib.auth.decorators import login\_required

@login\_required

def update\_profile(request):

user = request.user

if request.method == 'POST':

form = UpdateProfileForm(request.POST, request.FILES, instance=user)

if form.is\_valid():

form.save() # حفظ التحديثات

return redirect('profile') # بعد حفظ التحديثات، يمكن توجيه المستخدم إلى صفحة الملف الشخصي

else:

form = UpdateProfileForm(instance=user)

return render(request, 'accounts/update\_profile.html', {'form': form})

**In update\_profile.html**

<h2>Update Your Profile</h2>

<form method="POST" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Update Profile</button>

</form>

**In urls.py**

path('update\_profile/', views.update\_profile, name='update\_profile'),

**Third Way:**

**In models.py**

from django.contrib.auth.models import User

from django.db import models

from django.db.models.signals import post\_save

from django.dispatch import receiver

class UserProfile(models.Model):

GENDER\_CHOICES = [ ('M', 'ذكر'),('F', 'أنثى'),('O', 'أخرى'),]

user = models.OneToOneField(User, related\_name='profile', on\_delete=models.CASCADE)

profile\_picture = models.ImageField(upload\_to='profile\_pictures/', null=True, blank=True)

address = models.CharField(max\_length=255, null=True, blank=True)

gender = models.CharField(max\_length=1, choices=GENDER\_CHOICES, null=True, blank=True)

date\_of\_birth = models.DateField(null=True, blank=True)

phone\_number = models.CharField(max\_length=15, null=True, blank=True, verbose\_name="رقم الهاتف")

national\_id = models.CharField(max\_length=20, null=True, blank=True, verbose\_name="رقم الهوية")

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

return self.user.username

@receiver(post\_save, sender=User)

def create\_userprofile(sender, instance, created, \*\*kwargs):

if created:

UserProfile.objects.get\_or\_create(user=instance)

@receiver(post\_save, sender=User)

def save\_userprofile(sender, instance, \*\*kwargs):

try:

instance.profile.save()

except UserProfile.DoesNotExist:

pass

**In forms.py**

from django import forms

from .models import UserProfile

from django.contrib.auth.models import User

class UserForm(forms.ModelForm):

class Meta:

model = User

fields = ['username', 'first\_name', 'last\_name']

widgets = {

'username': forms.TextInput(attrs={'class': 'form-control'}),

'first\_name': forms.TextInput(attrs={'class': 'form-control'}),

'last\_name': forms.TextInput(attrs={'class': 'form-control'}),}

class UserProfileForm(forms.ModelForm):

class Meta:

model = UserProfile

fields = ['profile\_picture', 'address', 'gender', 'date\_of\_birth', 'phone\_number', 'national\_id']

widgets = {

'profile\_picture': forms.ClearableFileInput(attrs={'class': 'form-control', 'placeholder': 'Add Profile Image'}),

'address': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'العنوان'}),

'gender': forms.Select(attrs={'class': 'form-control', 'placeholder': 'الجنس'}),

'date\_of\_birth': forms.DateInput(attrs={'class': 'form-control', 'type': 'date', 'placeholder': 'تاريخ الميلاد'}),

'phone\_number': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'رقم الهاتف'}),

'national\_id': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'رقم الهوية'}),

}

**In views.py**

from .forms import UserForm, UserProfileForm

from .models import UserProfile

def profile\_update(request):

user\_profile, created = UserProfile.objects.get\_or\_create(user=request.user)

if request.method == 'POST':

user\_form = UserForm(request.POST, instance=request.user)

profile\_form = UserProfileForm(request.POST, request.FILES, instance=user\_profile)

if user\_form.is\_valid() and profile\_form.is\_valid():

user\_form.save()

profile\_form.save()

return redirect('my\_account')

else:

user\_form = UserForm(instance=request.user)

profile\_form = UserProfileForm(instance=user\_profile)

return render(request, 'myaccount.html', {

'user\_form': user\_form, 'profile\_form': profile\_form,})

**In myaccount.html**

{% if user.profile.profile\_picture %}

<div class="mt-5 mb-2">

<img src="{{ user.profile.profile\_picture.url }}" alt="" >

</div>

{% endif %}

<form action="" method="POST" enctype="multipart/form-data" novalidate>

{% csrf\_token %}

{{ user\_form.as\_p }}

{{ profile\_form.as\_p }}

<button type="submit" class="btn btn-success">Save Changes</button>

</form>

**In urls.py**

path('account/',views.profile\_update, name='my\_account'),