**Setting up registration form:**

Api/settings.py - Open up *api/settings.py* and add the lines for other \_DIR to look like :

BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  
TEMPLATE\_DIR = os.path.join(BASE\_DIR, 'templates')  
STATIC\_DIR = os.path.join(BASE\_DIR, 'static')  
MEDIA\_DIR = os.path.join(BASE\_DIR, 'media')

Also In the Installed apps section you need to add clubkit.api

INSTALLED\_APPS = [  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
 'rest\_framework',  
 'clubkit.api',  
]

Add *TEMPLATE\_DIR* to *‘DIRS’: []* section too

TEMPLATES = [  
 {  
 'BACKEND': 'django.template.backends.django.DjangoTemplates',  
 'DIRS': [TEMPLATE\_DIR, ],  
 'APP\_DIRS': True,  
 'OPTIONS': {  
 'context\_processors': [  
 'django.template.context\_processors.debug',  
 'django.template.context\_processors.request',  
 'django.contrib.auth.context\_processors.auth',  
 'django.contrib.messages.context\_processors.messages',  
 ],  
 },  
 },  
]

Add these lines at the end of our *api/settings.py* for *static*, *templates* and *media* folder :

STATIC\_URL = '/static/'  
STATICFILES\_DIRS = [STATIC\_DIR, ]  
  
MEDIA\_ROOT = MEDIA\_DIR  
MEDIA\_URL = '/media/'  
  
LOGIN\_URL = '/api/user\_login/'

Next we create the *api/models.py* to be used that will be the basis of our *forms.py :*

from django.db import models  
from django.contrib.auth.models import User  
  
  
class UserProfileInfo (models.Model):  
  
 user = models.OneToOneField(User, on\_delete=models.CASCADE)  
 profile\_pic = models.ImageField(upload\_to='profile\_pics', blank=True)  
  
 def \_\_str\_\_(self):  
 return self.user.username

from *models.py*we now switch to *api/forms.py*

from django import forms  
from clubkit.api.models import UserProfileInfo  
from django.contrib.auth.models import User  
  
  
class UserForm(forms.ModelForm):  
 password = forms.CharField(widget=forms.PasswordInput())  
  
 class Meta():  
 model = User  
 fields = ('username', 'password', 'email',)  
  
  
class UserProfileInfoForm(forms.ModelForm):  
  
 class Meta():  
 model = UserProfileInfo  
 fields = ('profile\_pic',)

Register your models in *api/admin.py* too:

from django.contrib import admin  
from clubkit.api.models import UserProfileInfo  
  
# Register your models here.  
admin.site.register(UserProfileInfo)

View is generated for our *.html*pages in *api/views.py* :

from django.contrib.auth.models import User  
from rest\_framework import viewsets  
from clubkit.api.serializers import UserSerializer  
from django.shortcuts import render  
from clubkit.api.forms import UserForm, UserProfileInfoForm  
from django.contrib.auth import authenticate, login, logout  
from django.http import HttpResponseRedirect, HttpResponse  
from django.urls import reverse  
from django.contrib.auth.decorators import login\_required  
  
  
class UserViewSet(viewsets.ModelViewSet):  
 *"""  
 API endpoint that allows users to be viewed or edited.  
 """* queryset = User.objects.all().order\_by('-date\_joined')  
 serializer\_class = UserSerializer  
  
 '''  
 Rather than write multiple views we're grouping together all the common behavior into classes called ViewSets.  
 We can easily break these down into individual views if we need to,   
 but using viewsets keeps the view logic nicely organized as well as being very concise.  
 '''  
  
  
def index(request):  
 return render(request, 'index.html')  
  
  
@login\_required  
def special():  
 return HttpResponse("You are logged in !")  
  
  
@login\_required  
def user\_logout(request):  
 logout(request)  
 return HttpResponseRedirect(reverse('index'))  
  
  
def register(request):  
 registered = False  
 if request.method == 'POST':  
 user\_form = UserForm(data=request.POST)  
 profile\_form = UserProfileInfoForm(data=request.POST)  
 if user\_form.is\_valid() and profile\_form.is\_valid():  
 user = user\_form.save()  
 user.set\_password(user.password)  
 user.save()  
 profile = profile\_form.save(commit=False)  
 profile.user = user  
 if 'profile\_pic' in request.FILES:  
 print('found it')  
 profile.profile\_pic = request.FILES['profile\_pic']  
 profile.save()  
 registered = True  
 else:  
 print(user\_form.errors, profile\_form.errors)  
 else:  
 user\_form = UserForm()  
 profile\_form = UserProfileInfoForm()  
 return render(request,  
 'registration.html',  
 {'user\_form': user\_form,  
 'profile\_form': profile\_form,  
 'registered': registered})  
  
  
def user\_login(request):  
 if request.method == 'POST':  
 username = request.POST.get('username')  
 password = request.POST.get('password')  
 user = authenticate(username=username, password=password)  
 if user:  
 if user.is\_active:  
 login(request, user)  
 return HttpResponseRedirect(reverse('index'))  
 else:  
 return HttpResponse("Your account was inactive.")  
 else:  
 print("Someone tried to login and failed.")  
 print("They used username: {} and password: {}".format(username, password))  
 return HttpResponse("Invalid login details given")  
 else:  
 return render(request, 'login.html', {})

The templates can be arranged now for generating the views, we use four templates for this : *‘base.html’, ‘registration.html’, ‘login.html’, ‘index.html’*

***base.html*** :

<!DOCTYPE html>  
<html>  
 <head>  
 <meta charset="utf-8">  
 <title>Base</title>  
 <!-- Latest compiled and minified CSS -->  
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">  
</head>  
 <body>  
 <nav class="navbar navbar-default navbar-static-top">  
 <div class="container">  
 <ul class="nav navbar-nav">  
{# Django Home Link / Admin Link / Register Link#}  
 <li><a class="navbar-brand" href="{% url 'index' %}">DJANGO</a></li>  
 <li><a class="navbar-link" href="{% url 'admin:index' %}">Admin</a></li>  
 <li><a class="navbar-link" href="{% url 'api:register' %}">Register</a></li>  
 {# Some logic on what to display for last item#}  
 {% if user.is\_authenticated %}  
 <li><a href="{% url 'logout' %}">Logout</a></li>  
 {% else %}  
 <li><a class="navbar-link" href="{% url 'api:user\_login' %}">Login</a></li>  
 {% endif %}  
</ul>  
 </div>  
 </nav>  
 <div class="container">  
 {% block body\_block %}  
 {% endblock %}  
 </div>  
 </body>  
</html>

***index.html*** :

{% extends "base.html" %}  
{% block body\_block %}  
<div class="container">  
 <div class="home">  
 <h1>Welcome to ClubKit!</h1>  
 {% if user.is\_authenticated %}  
 <h2>Hello {{ user.username }}</h2>  
 {% else %}  
 <h2>Register or Login if you'd like to</h2>  
 {% endif %}  
 </div>  
</div>  
{% endblock %}

***login.html*** :

{% extends 'base.html' %}  
{% block body\_block %}  
 <div class="container">  
 <div class="login">  
 <h1>Login here :</h1>  
<form method="post" action="{% url 'api:user\_login' %}">  
 {% csrf\_token %}  
 {# A more "HTML" way of creating the login form#}  
 <label for="username">Username:</label>  
 <input type="text" name="username" placeholder="Username">  
 <label for="password">Password:</label>  
 <input type="password" name="password" placeholder="Password">  
<input type="submit" name="" value="Login">  
</form>  
</div>  
 </div>  
{% endblock %}

***registration.html*** :

{% extends "base.html" %}  
{% load staticfiles %}  
{% block body\_block %}  
 <div class="container">  
 <div class="registration">  
 {% if registered %}  
 <h1>Thank you for registering!</h1>  
 {% else %}  
 <h1>Register Here</h1>  
 <h3>Just fill out the form.</h3>  
<form enctype="multipart/form-data" method="POST">  
 {% csrf\_token %}  
 {{ user\_form.as\_p }}  
 {{ profile\_form.as\_p }}  
 <input type="submit" name="" value="Register">  
 </form>  
   
 {% endif %}  
 </div>  
 </div>  
{% endblock %}

We then register the above urls into our project *urls.py* files for this create a file *api/urls.py*

from django.conf.urls import url  
from clubkit.api import views  
  
# SET THE NAMESPACE!  
app\_name = 'api'  
# Be careful setting the name to just /login use userlogin instead!  
  
urlpatterns = [  
 url(r'^register/$', views.register, name='register'),  
 url(r'^user\_login/$', views.user\_login, name='user\_login'),  
]

In the main *urls.py* file the rest of the pattern is specified.

from django.conf.urls import url, include  
from django.contrib import admin  
from django.urls import path  
from clubkit.api import views  
  
# router = routers.DefaultRouter()  
# router.register(r'users', views.UserViewSet)  
  
# Wire up our API using automatic URL routing.  
# Additionally, we include login URLs for the browsable API.  
urlpatterns = [  
  
 path('admin/', admin.site.urls),  
 url(r'^$', views.index, name='index'),  
 url(r'^special/', views.special, name='special'),  
 url(r'^clubkit/', include('clubkit.api.urls')),  
 url(r'^logout/$', views.user\_logout, name='logout'),  
   
]

Apply the migrations using:

*$ python manage.py migrate  
$ python manage.py makemigrations app  
$ python manage.py migrate*

Ref: https://medium.com/@himanshuxd/how-to-create-registration-login-webapp-with-django-2-0-fd33dc7a6c67