Lessons

lessons/admin.py

For the superuser manage this pdf's lessons (add, delete, change name...), we have to write this script ->

```
from django.contrib import admin
from .models import Python, Django

admin.site.register(Python)
admin.site.register(Django)
```

lessons/models.py

Models contains the essencial fields and beahaviors of the data we are storing

We have to add the variable document, type file, in our Django Project, like this ->

```
from django.db import models


class Python(models.Model):
    python_file = models.FileField()


class Django(models.Model):
    django_file = models.FileField()
```

lessons/views.py

The views is a python script that takes a web request and returns a web response

Each response can be given by a class or a function

Render script takes 3 argumens: the request, a template, the variables for the template; and this is how we are going to give your response

Overwrite this scrips ->

Imports ->

```
from django.shortcuts import render
from django.contrib.auth import authenticate, login, logout
from django.views.generic import View
from .users import UserForm
from django.http import HttpResponseRedirect
from django.core.urlresolvers import reverse
from .models import Python, Django
```

Registration view ->

Saves the user registration and authenticates the user, if the form is not valid the response is the same page with the same forms to be filled again

```
class UserFormView(View):
    form_class = UserForm

    def get(self, request):
        form = self.form_class(None)
        return render(request, 'registration.html', {'form': form})

    def post(self, request):
        form = self.form_class(request.POST)
        if form.is_valid():
            user = form.save(commit=False)
            username = form.cleaned_data['username']
            email = form.cleaned_data['email']
            password = form.cleaned_data['password']
            user.set_password(password)
            if user.email:
                user.save()
                user = authenticate(username=username, email=email,
password=password)
                if user is not None:
                    login(request, user)
                    return HttpResponseRedirect(reverse('home'))
        return render(request, 'registration.html', {'form': form})
```

Logout view ->

Logs out the user, if the user was logged in we get the variable: 1, otherwise, we get the variable 2; this variable is going to be used in the index template

```
def logout_user(request):
    if request.user.is_authenticated():
        logout(request)
        return render(request, 'index.html', {'logged': 1})
    return render(request, 'index.html', {'logged': 2})
```

Home view ->

If the user is authenticated, the response is the page with all the lessons, otherwise, the response is the index page

```
def home(request):
    if request.user.is_authenticated():
        python_list = Python.objects.all()
        django_list = Django.objects.all()
        return render(request, 'home.html', {'username':
request.user.get_username(),
                                              'python_list':
python_list, 'django_list': django_list})
    python = Python.objects.all()[0]
    django = Django.objects.all()[0]
    return render(request, 'index.html', {'registration': 1,
'python_lesson': python,
                                           'django_lesson': django})
```

Index view ->

First page of the webservice, the user have just access to two lessons

```
def index(request):
    python = Python.objects.all()[0]
    django = Django.objects.all()[0]
    return render(request, 'index.html', {'python_lesson': python,
'django_lesson': django})
```

Login view ->

Just similar to the registration view

```
class LogUser(View):
    form_class = UserForm

    def get(self, request):
        form = self.form_class(None)
        return render(request, 'login.html', {'form': form})

    def post(self, request):
        form = self.form_class(request.POST)
        username = request.POST['username']
        email = request.POST['email']
        password = request.POST['password']
        user = authenticate(username=username, email=email,
password=password)
        if user is not None:
            if email == user.email:
                login(request, user)
                return HttpResponseRedirect(reverse(home))
            return render(request, 'login.html', {'email': 1, 'form':
form})
        return render(request, 'login.html', {'form': form})
```

Creating more directories and files

In lessons directory create -> templates, users.py, urls.py

lessons/urls.py

If you need, for your webservice, two urls with names: test/1 and test/2, you don’t need to create those two urls in djangofromscratch/urls.py. What you need to do is create the url test in djangofromscratch/urls.py and include lessons.urls, just like we did, and then in lessons/urls.py add ‘1’ and ‘2’, in our case, we just want to add ‘home’, like this->

```
from django.conf.urls import url
from .views import home

urlpatterns = [
    url(r'^home/$', home, name='home')
]
```

lessons/users.py

This is how we manage ours users ->

```
from django.contrib.auth.models import User
from django import forms


class UserForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput)

    class Meta(object):
        model = User
        fields = ['username', 'email', 'password']
```