**Lab 5 Part 2**

**User Authentication**

Now that we have a working custom user model we can add the functionality every website needs: the ability to sign up, log in, and log out users. Django provides everything we need for log in and log out but we will need to create our own form to sign up new users. We will also build a basic homepage with links to all three features

so we don’t have to type in the URLs by hand every time.

**Templates**

Create a new templates directory and within it a registration directory as that is where Django will look for the log in template.

**Command Line**

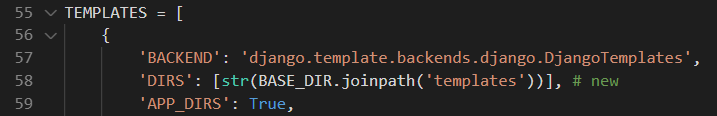
(env) djangoprojects\lab5>mkdir templates

(env) djangoprojects\lab5>cd templates

(env) djangoprojects\lab5\templates>mkdir registration

Now we need to tell Django about this new directory by updating the configuration

for 'DIRS' in settings.py.



If you think about what happens when you log in or log out of a site, you are immediately redirected to a subsequent page. We need to tell Django where to send

users in each case. The LOGIN\_REDIRECT\_URL and LOGOUT\_REDIRECT\_URL settings do that.

Add these two lines at the bottom of the settings.py file.

**Code**



Now we can create four new templates:

**Command Line**

(env) djangoprojects\lab5\templates>new-item base.html

(env) djangoprojects\lab5\templates>new-item home.html

Move into the registration folder

(env) djangoprojects\lab5\templates>cd registration

Create the following two files:

(env) djangoprojects\lab5\templates>registration>new-item login.html

(env) djangoprojects\lab5\templates>registration>new-item signup.html

The code is provided for each of these files – just copy and paste it in.

**base.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>{% block title %}Newspaper App{% endblock title %}</title>

</head>

<body>

<main>

{% block content %}

{% endblock content %}

</main>

</body>

</html>

**home.html**

{% extends 'base.html' %}

{% block title %}Home{% endblock title %}

{% block content %}

{% if user.is\_authenticated %}

Hi {{ user.username }}!

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

{% else %}

<p>You are not logged in</p>

<a href="{% url 'login' %}">Log In</a> |

<a href="{% url 'signup' %}">Sign Up</a>

{% endif %}

{% endblock content %}

**login.html**

{% extends 'base.html' %}

{% block title %}Log In{% endblock title %}

{% block content %}

<h2>Log In</h2>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Log In</button>

</form>

{% endblock content %}

**signup.html**

{% extends 'base.html' %}

{% block title %}Sign Up{% endblock title %}

{% block content %}

<h2>Sign Up</h2>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

{% endblock content %}

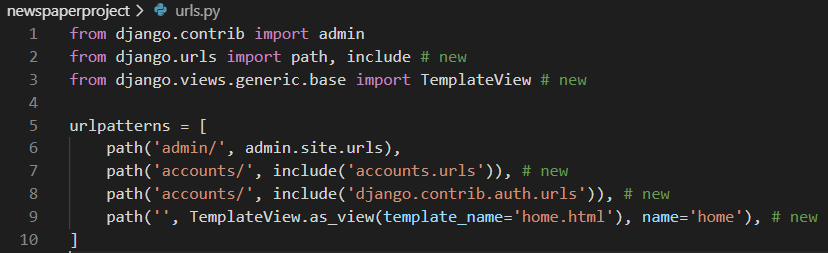
The next step is to create the URLs and views.

**URLs**

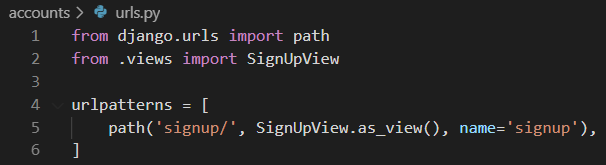
In our newspaperproject/urls.py file we want to have our home.html template appear as the homepage. But we don’t want to build a dedicated pages app just yet, so we can use the shortcut of importing TemplateView and setting the template\_name right in our url pattern.

Next we want to “include” both the accounts app and the built-in auth app. The reason is that the built-in auth app already provides views and urls for log in and log out. But for sign up we will need to create our own view and url. To ensure that our URL routes are consistent we place them both at accounts/ so the eventual URLS will be /accounts/login, /accounts/logout, and /accounts/signup.

Open newspaperproject/urls.py and add in the following code:

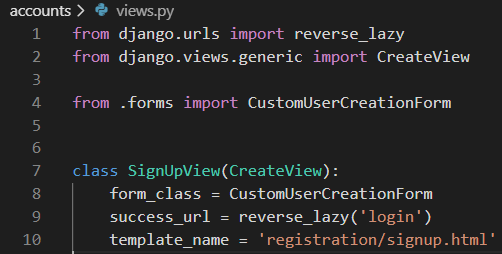


Now create a urls.py file in the accounts app.



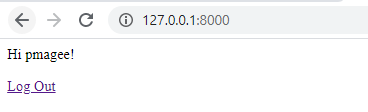
The last step is our views.py file which will contain the logic for our sign up form. We’re using Django’s generic CreateView here and telling it to use our CustomUserCreationForm, to redirect to login once a user signs up successfully, and that our template is named signup.html.

Open accounts/views.py, delete the code at the top and enter the following code:

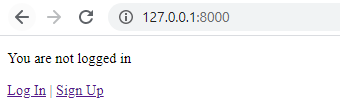


Start up the server with python manage.py runserver and go to the homepage at

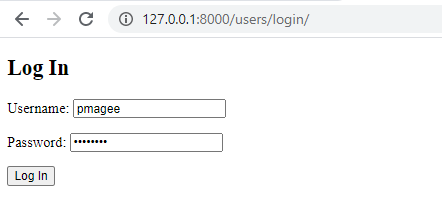
http://127.0.0.1:8000/.



We logged in to the admin in the previous lab so you should see a personalized greeting here. Click on the “Log Out” link.

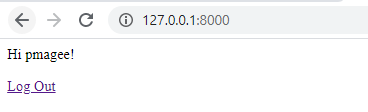


Now we are on the logged out homepage. Go ahead and click on login link and use your superuser credentials.



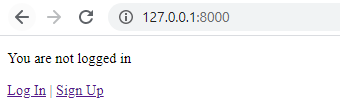
Upon successfully logging in you will be redirected back to the homepage and see the

same personalized greeting.

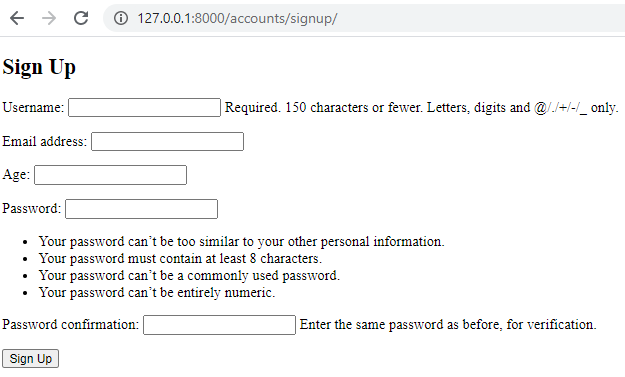


Now click the “Log Out” link to return to the homepage and this time click on the “Sign

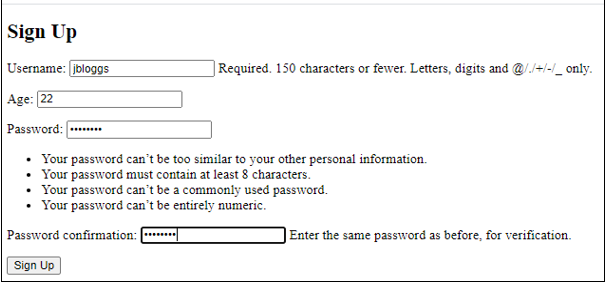
Up” link.



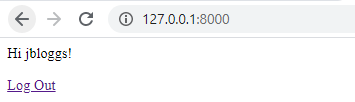
When you click the Sign Up button you are directed to the signup page. See that the age field is included.



Create a new user. Mine is called jbloggs and I’ve set the age to 22

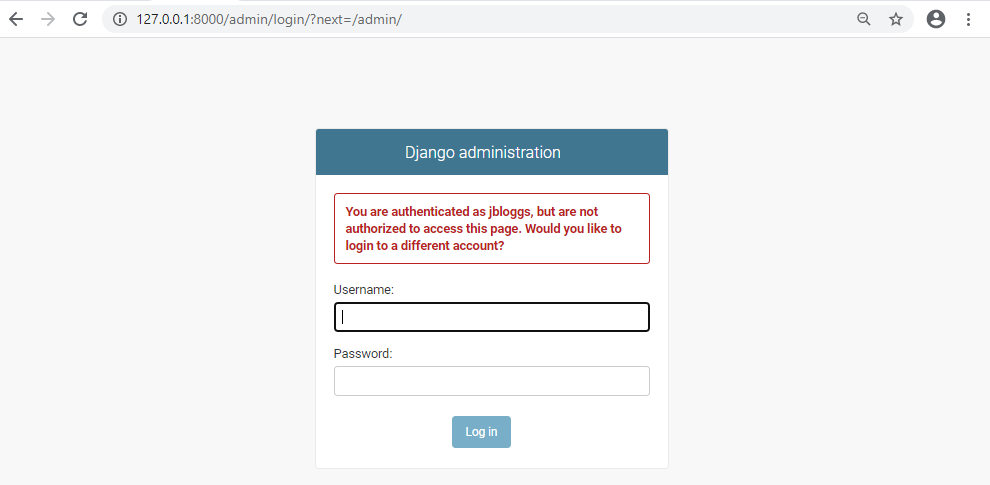


After successfully submitting the form you will be redirected to the log in page. Log in with your new user and you will again be redirected to the homepage with a personalized greeting for the new user.



Next we will try to log in to the admin to view our two user accounts. Navigate to:

http://127.0.0.1:8000/admin and we cannot log in. Why?



We are logged in our new jbloggs account not our superuser account. Only a superuser account has the permissions to log in to the admin! So use your superuser

account to log in instead.

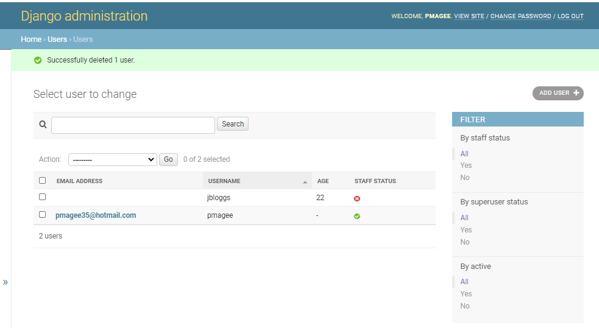
After you’ve done that you should see the normal admin homepage. Click on Users

and you can see our two users: the one we just created and your previous superuser

name (mine is pmagee).

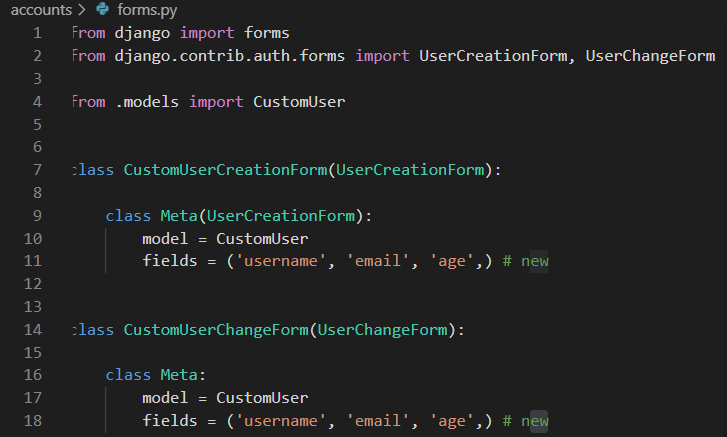
Everything is working but you may notice that there is no “Email address” for our

jbloggs. Why is that?



Well, look back at the sign up page at http://127.0.0.1:8000/accounts/signup/

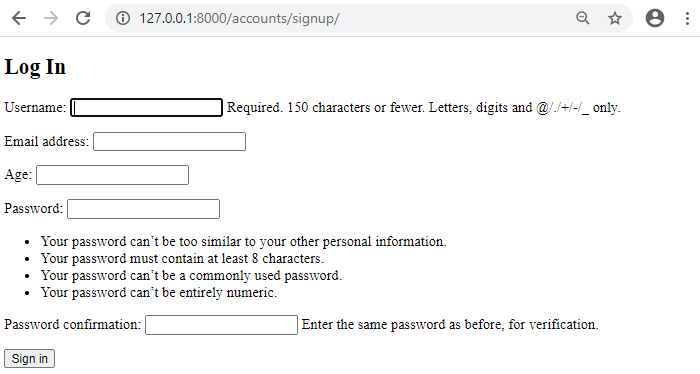
You’ll see that it asks for username, age, and password but not an email! However, we can easily change it. Let’s return to our users/forms.py file.



Change lines 11 and 18 to explicitly configure the fields we want to use

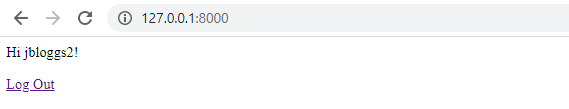
Currently under fields we’re using Meta.fields which just displays the default settings of username/password. But we can also explicitly set which fields we want displayed so let’s update it to ask for a username/email/password by setting it to ('username', 'email',). We don’t need to include the password fields because they are required! However, all the other fields can be configured however we choose.

Now if you try out the sign up page again at http://127.0.0.1:8000/accounts/signup/ you can see the additional “Email address” field is there.



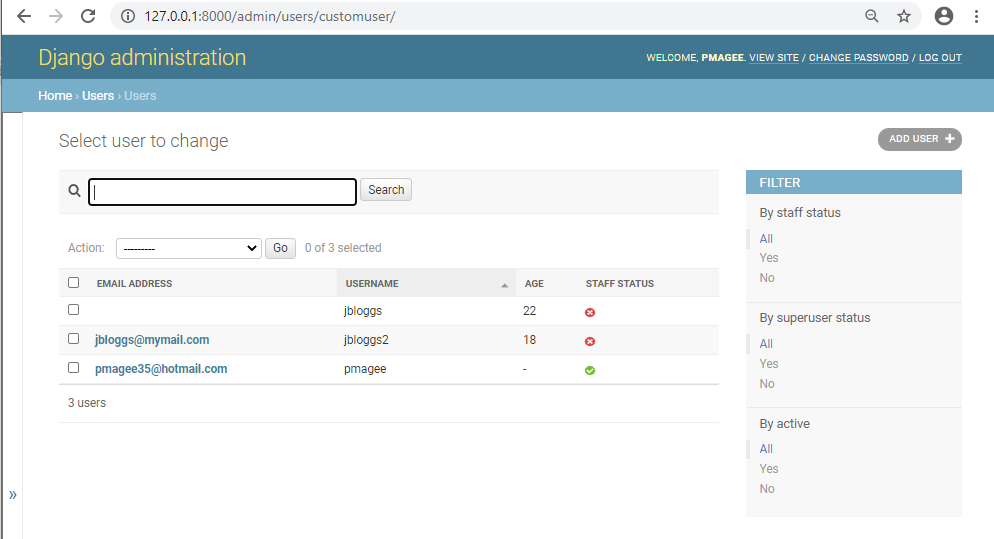
Sign up with a new user account. I’ve named mine jbloggs2 with an age of 18 and an

email address of jbloggs2@mymail.com. Continue to log in and you will see a personalized greeting on the homepage.



Then switch back to the admin page–log in using our superuser account to do so–and

all three users are on display.



Run the following git commands to update the local and remote repositories:

(env) djangoprojects\lab5>git add -A

(env) djangoprojects\lab5>git commit -m “lab 5 part 2 commit”

(env) djangoprojects\lab5>git push -u origin master

Django’s user authentication flow requires a little bit of set up but you should be starting to see that it also provides us incredible flexibility to configure sign up and log in exactly how we want.

So far our Newspaper app has a custom user model and working sign up, log in, and

log out pages. But you may have noticed our site doesn’t look very good. In the next

lab we will add Bootstrap for styling and create a dedicated pages app.