

Lab 6 Part 2

User Authentication

Now that we have a working custom user model, we can add the functionality all websites need: 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.

Step 1 - Create a new app for the Home Page

1. Create a new **app** called **pages**.
2. Register this app in **settings.py**.
3. Create a new **templates** folder.
4. Update the **settings.py** file to tell Django about this new **templates** folder.
5. Inside the **templates** folder create two files **base.html** and **home.html**.

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

base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %} {% endblock title %}</title>
</head>
<body>
    {% block content %}
    {% endblock content %}
</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 %}
```

6. Add the following code to **pages/views.py**:

```
pages > views.py > ...
1  from django.views.generic import TemplateView
2
3  # Create your views here.
4  class HomePageView(TemplateView):
5      template_name = 'home.html'
```

7. Create a **urls.py** file in the **pages app** and add in the following code:

```
pages > urls.py > ...
1  from django.urls import path
2  from .views import HomePageView
3
4  urlpatterns = [
5      path('', HomePageView.as_view(), name='home'),
6  ]
```

8. Add the path for the **pages** app into **newspaperproject/urls.py** using the following code:

```
16  from django.contrib import admin
17  from django.urls import path, include
18
19  urlpatterns = [
20      path('admin/', admin.site.urls),
21      path('', include('pages.urls')),
22  ]
```

Step 3 - Templates for Login and Signup

Inside the **templates** folder, create a new folder called a **registration** as this is where Django will look for the log in template.

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

```
125     LOGIN_REDIRECT_URL = 'home'
126     LOGOUT_REDIRECT_URL = 'home'
```

Inside the **templates/registration** folder create two files **login.html** & **signup.html** and copy and paste in the following code:

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 %}
```

Step 4 – Configure URLs

Next, we want to “include” both the **users** 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.

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

```
16 from django.contrib import admin
17 from django.urls import path, include
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('accounts/', include('django.contrib.auth.urls')),
22     path('users/', include('users.urls')),
23     path('', include('pages.urls')),
24 ]
```

Now create a **urls.py** file in the **users** app and add in the following code:

```
users > 📄 urls.py > ...
1  from django.urls import path
2  from .views import SignUpView
3
4  urlpatterns=[
5      path('signup/', SignUpView.as_view(), name='signup'),
6  ]
```

Step 5 – Users app Views

The last step is our **views.py** file which will contain the logic for our sign-up form. We are using Django's generic **CreateView** here and telling it the following:

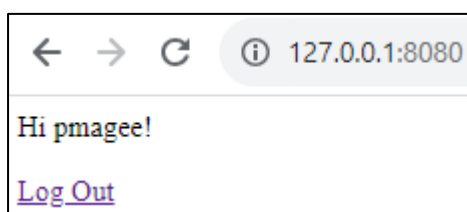
- Use the **CustomUserCreationForm**.
- Use **reverse_lazy** to redirect to the login page once a user signs up successfully.
- Template is named **signup.html**.

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

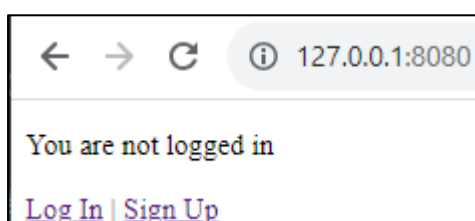
```
users > views.py > ...
1  from django.urls import reverse_lazy
2  from django.views.generic import CreateView
3  from .forms import CustomUserCreationForm
4
5  # Create your views here.
6
7  class SignUpView(CreateView):
8      form_class = CustomUserCreationForm
9      success_url = reverse_lazy('login')
10     template_name = 'registration/signup.html'
```

Step 6 – Run server

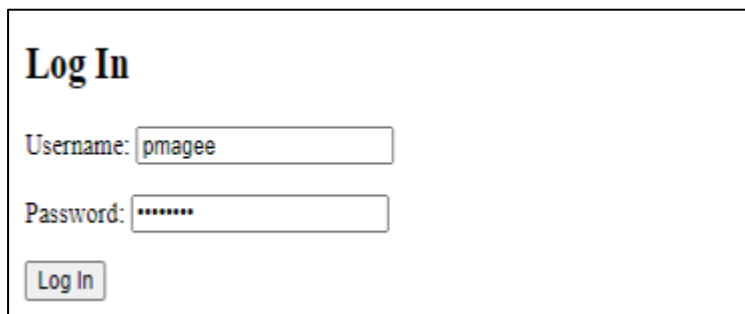
Start up the server with **python manage.py runserver 8080** and go to the homepage at <http://127.0.0.1:8080/> and log in with your superuser details.



You should see a personalized greeting here. Click on the “Log Out” link.

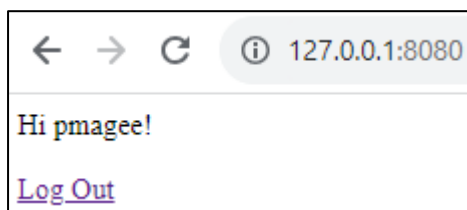


Now we are on the logged-out homepage. Go ahead and click on login link and use your superuser credentials.



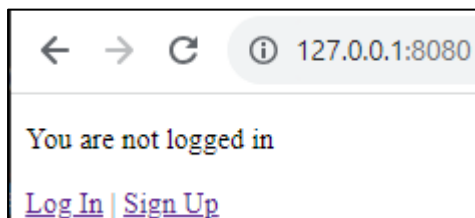
A screenshot of a web form titled "Log In". It contains two input fields: "Username:" with the text "pimagee" entered, and "Password:" with a masked password "*****". Below the fields is a "Log In" button.

Upon successfully logging in you will be redirected back to the homepage and see the same personalized greeting.



A screenshot of a web browser showing the logged-in homepage. The address bar displays "127.0.0.1:8080". The page content includes a personalized greeting "Hi pimagee!" and a "Log Out" link.

Now click the "Log Out" link to return to the homepage and this time click on the "Sign Up" link.



A screenshot of a web browser showing the logged-out homepage. The address bar displays "127.0.0.1:8080". The page content includes the text "You are not logged in" and links for "Log In" and "Sign Up".

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 and press the **Sign Up** button.

Sign Up

Username: Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Age:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

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.

← → ↻ ⓘ 127.0.0.1:8080

Hi jbloggs!

[Log Out](#)

Next, we will try to log in to the admin to view our two user users. Navigate to: <http://127.0.0.1:8080/admin> and we cannot log in. Why?

← → ↻ ⓘ 127.0.0.1:8080/admin/login/?next=/admin/ 🔍 ☆ 👤 ⋮

Django administration

You are authenticated as jbloggs, but are not authorized to access this page. Would you like to login to a different account?

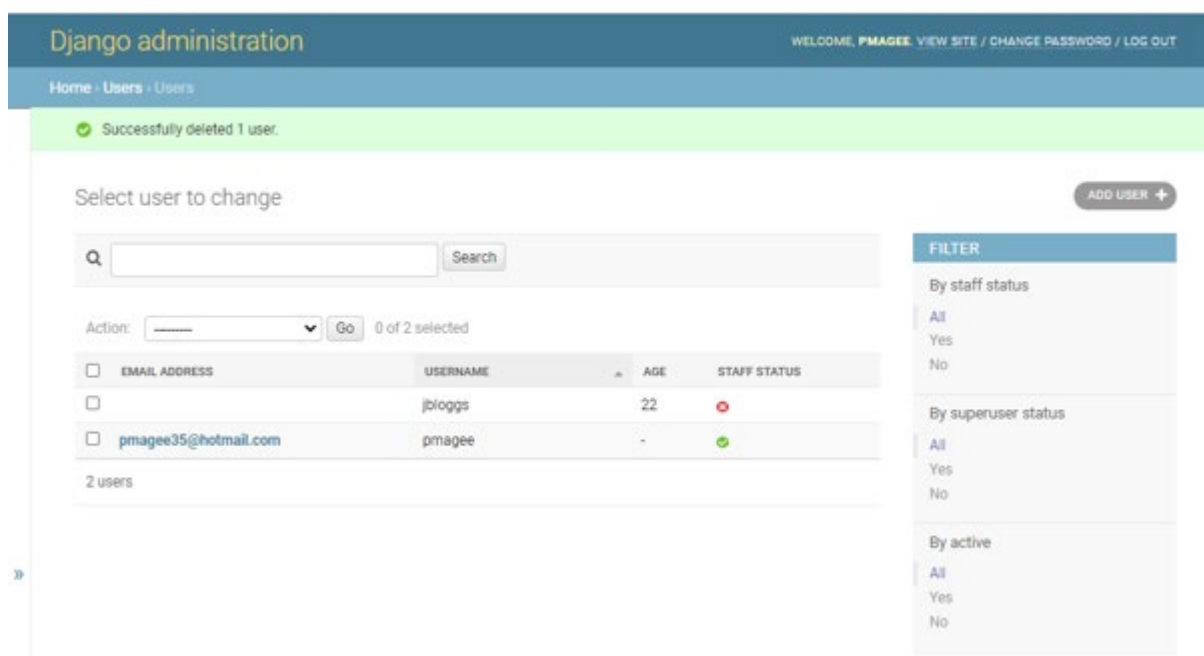
Username:

Password:

We are logged in with our new jbloggs account which is not a 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 have 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.

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://127.0.0.1:8080/users/signup/>. You will 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.

Step 7 - Include email address in Sign Up page

Currently under fields we are 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/age/password by setting it to ('username', 'email', 'age'). We don't need to include the password fields because they are required! However, all the other fields can be configured however we choose.


```
users > forms.py > ...
1  from django import forms
2  from django.contrib.auth.forms import UserCreationForm, UserChangeForm
3  from .models import CustomUser
4
5  class CustomUserCreationForm(UserCreationForm):
6      class Meta:
7          model = CustomUser
8          fields = ('username', 'email', 'age',)
9
10 class CustomUserChangeForm(UserChangeForm):
11     class Meta:
12         model = CustomUser
13         fields = ('username', 'email', 'age',)
```

Change lines 9 and 14 to explicitly configure the fields we want to use

Now if you try out the sign-up page again at <http://127.0.0.1:8080/users/signup/> you can see the additional “Email address” field is there.

Log In

Username: Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email address:

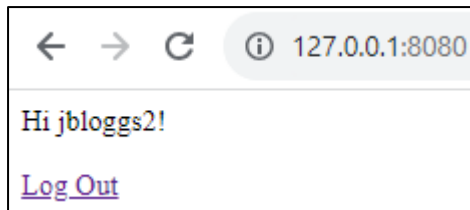
Age:

Password:

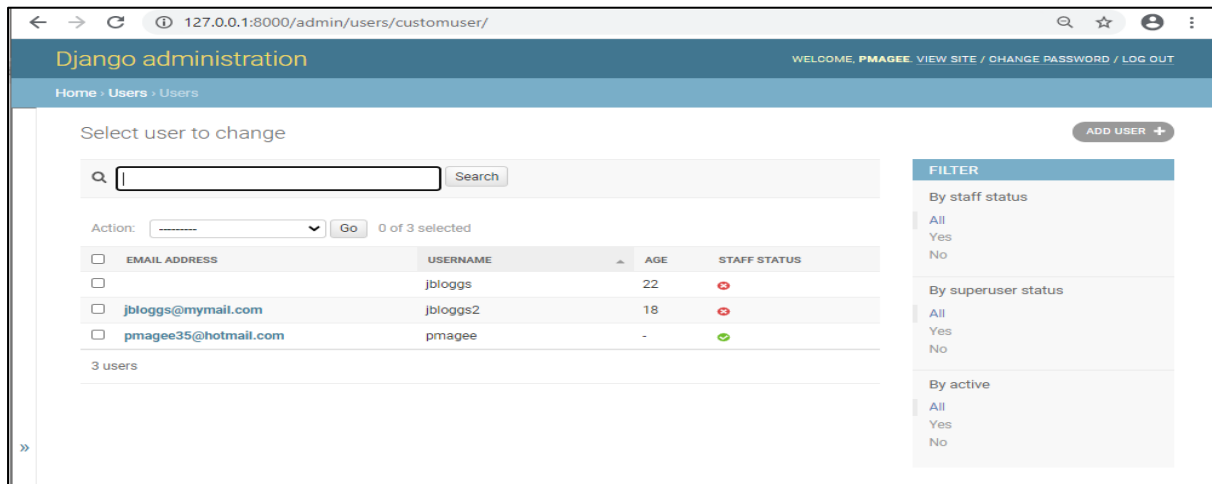
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

Sign up with a new user account. 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 and note the email address for jbloggs2.



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

git add -A

git commit -m "lab 6 part 2 commit"

git push -u origin main