**Lab 5 Part 6**

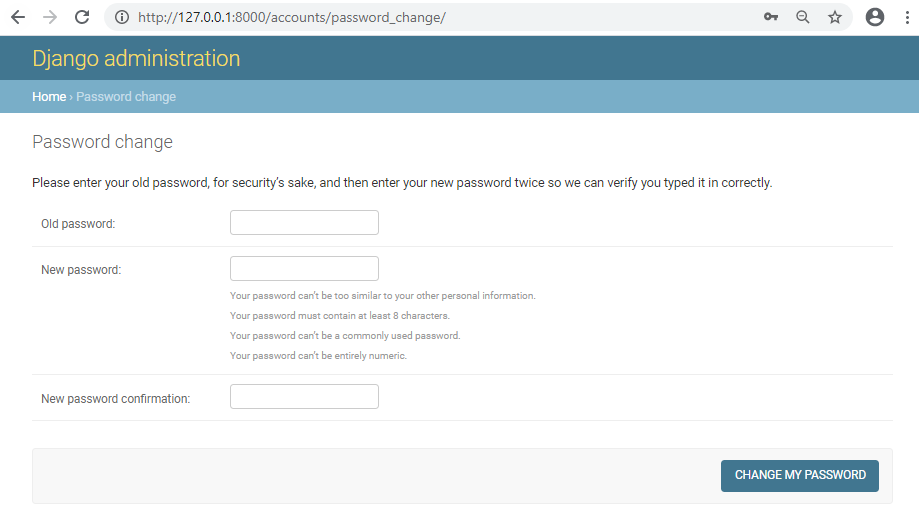
**Password Change & Reset**

In this exercise we will add new functionality to the newspaper project to so that users will be able to change their current password or, if they’ve forgotten it, to reset it via email.

Initially we will implement Django’s built-in views and URLs for both password change and password reset before then customizing them with our own Bootstrap-styled templates.

**Password Change**

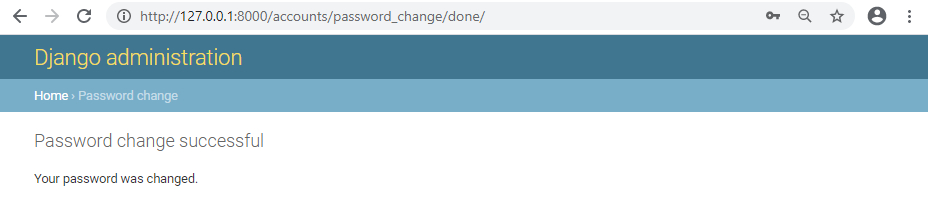
Django provides a default implementation that already works at this stage. To try it out launch the server and click on the “Log In” button to make sure you are logged in. Then navigate to the “Password change” page at <http://127.0.0.1:8000/accounts/password_change/>.



Enter in both your old password and then a new one. Then click the “Change My Password” button.

You will be redirected to the “Password change successful” page located at:

http://127.0.0.1:8000/accounts/password\_change/done/.



**Customizing password change**

Next we will change these two password change pages so that they match the look and feel of our Newspaper site. Because Django already has created the views and URLs for us, we only need to add new templates.

On the command line stop the local server Control+BREAK and create two new template files in the registration directory.

**Command Line**

(env) djangoprojects\lab5\templates\regiatration> new-item password\_change\_form.html

(env) djangoprojects\lab5\templates\registration> new-item password\_change\_done.html

Open the password\_change\_form.html in VS Code and update it with the following code:

{% extends 'base.html' %}

{% block title %}Password Change{% endblock title %}

{% block content %}

<h1>Password change</h1>

<p>Please enter your old password, for security's sake, and then enter your

new password twice so we can verify you typed it in correctly.</p>

<form method="POST">

{% csrf\_token %}

{{ form.as\_p }}

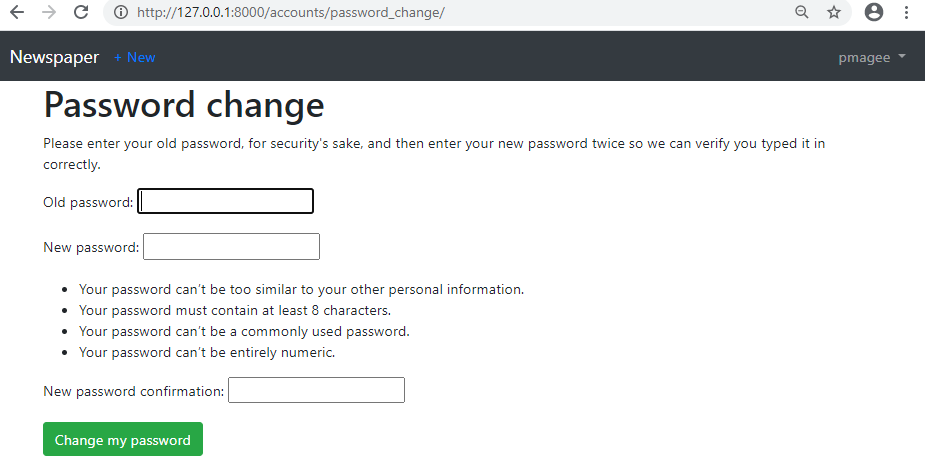
<input class="btn btn-success" type="submit" value="Change my password">

</form>

{% endblock content %}

At the top we extend base.html and set our page title. Because we used “block” titles in our base.html file we can override them here. The form uses POST since we are sending data and a csrf\_token for security reasons. By using form.as\_p we are simply displaying in paragraphs the content of the default password reset form. And finally we include a submit button that uses Bootstrap’s btn btn-success styling to make it green.

Reload the page at http://127.0.0.1:8000/accounts/password\_change/ to see our changes.



Next open the password\_change\_done.html and update it with the following code:

{% extends 'base.html' %}

{% block title %}Password Change Successful{% endblock title %}

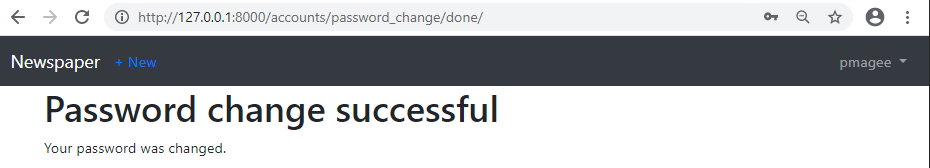
{% block content %}

<h1>Password change successful</h1>

<p>Your password was changed.</p>

{% endblock content %}

Now if you go ahead and change your password you should see the following page:



**Password Reset**

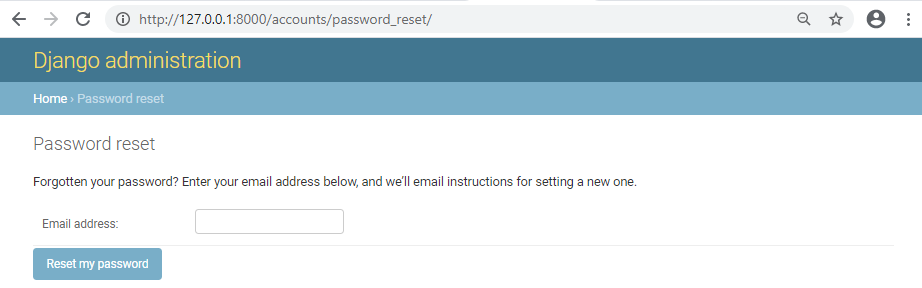
Password reset handles the common case of users forgetting their passwords. The steps are very similar to configuring password change, as we just did. Django already provides a default implementation that we will use and then customize the templates so it matches the rest of our site.

The only configuration required is telling Django how to send emails. After all, a user can only reset a password if they have access to the email linked to the account. For testing purposes we can rely on Django’s console backend setting which outputs the email text to our command line console.

At the bottom of the settings.py file make the following one-line change.

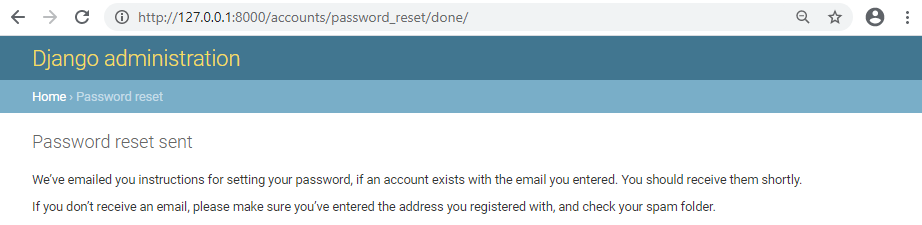


Access the page at <http://127.0.0.1:8000/accounts/password_reset/>

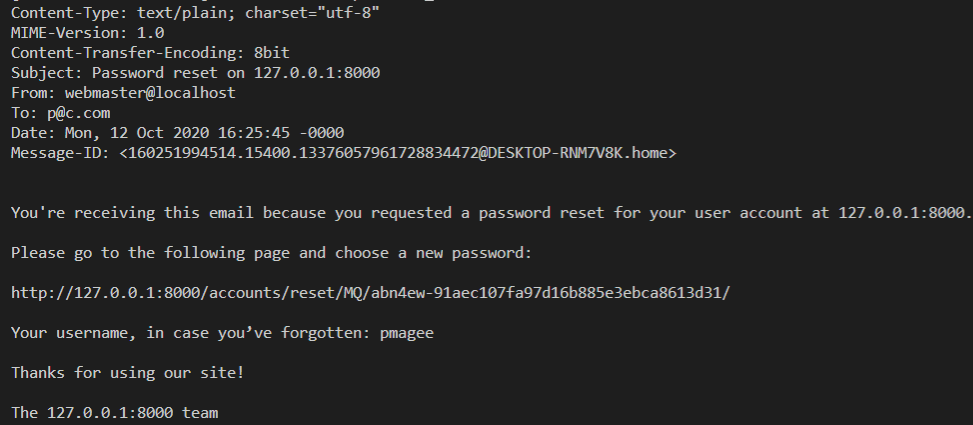


Make sure the email address you enter matches one of your Django user accounts. Upon submission you will then be redirected to the password reset done page at:

http://127.0.0.1:8000/accounts/password\_reset/done/.



The message above says to check our email. Since we’ve told Django to send emails to the command line console, the email text will now be there. This is what I see in my console.

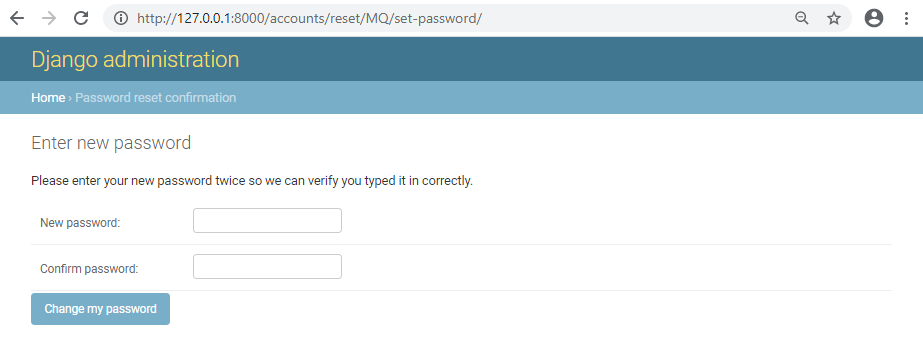


Your email text should be identical except for three lines:

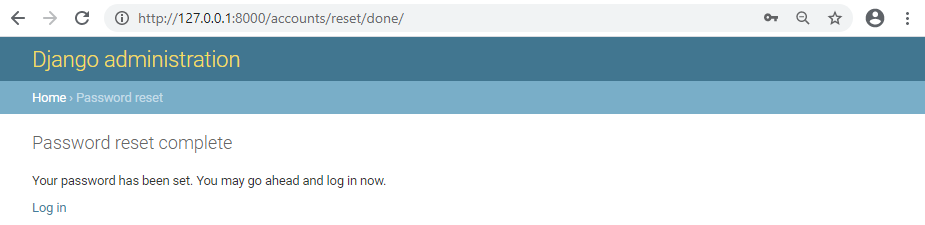
* the “To” on the sixth line contains the email address of the user
* the URL link contains a secure token that Django randomly generates for us and can be used only once
* Django helpfully reminds us of our username

Copy the link you received which is similar to that shown below into your browser and you will be redirected to the “change password page”.





Now enter in a new password and click on the “Change my password” button. The final step is you will be redirected to the “Password reset complete” page.



To confirm everything worked, click on the “Log in” link and use your new password.

It should work.

**Custom Templates**

As with “Password change” we only need to create new templates to customize the look and feel of password reset. Stop the local server with Control+BREAK and then create four new template files.

(env) djangoprojects\lab5\templates\registration> new-item password\_reset\_form.html

(env) djangoprojects\lab5\templates\registration> new-item password\_reset\_done.html

(env) djangoprojects\lab5\templates\registration> new-item password\_reset\_confirm.html

(env) djangoprojects\lab5\templates\registration> new-item password\_reset\_complete.html

Open password\_reset\_form.html and update it with the following code:

{% extends 'base.html' %}

{% block title %}Forgot Your Password?{% endblock title %}

{% block content %}

<h1>Forgot your password?</h1>

<p>Enter your email address below, and we'll email instructions for setting

a new one.</p>

<form method="POST">

{% csrf\_token %}

{{ form.as\_p }}

<input class="btn btn-success" type="submit" value="Send me instructions!">

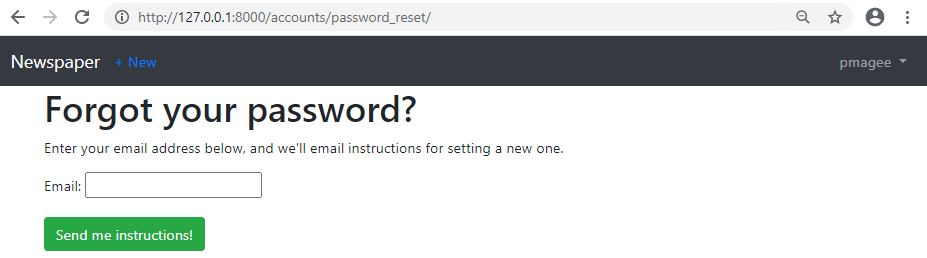
</form>

{% endblock content %}

Start up the server again with python manage.py runserver. Navigate to:

http://127.0.0.1:8000/accounts/password\_reset/.

Refresh the page you can see our new page.



Now we can update the other three pages. Each takes the same form of extending

base.html, a new title, new content text, and for password\_reset\_confirm.html an

updated form as well.

Copy the following code into password\_reset\_done.html

{% extends 'base.html' %}

{% block title %}Email Sent{% endblock title %}

{% block content %}

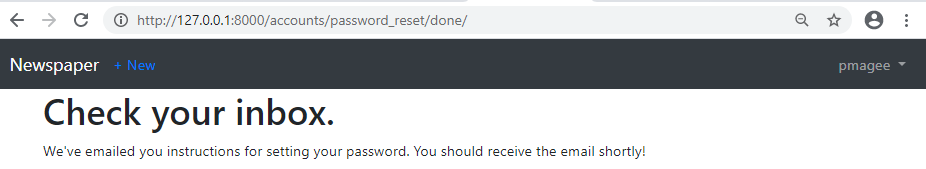
<h1>Check your inbox.</h1>

<p>We've emailed you instructions for setting your password.

You should receive the email shortly!</p>

{% endblock content %}

When you click the “Send me instructions” button you should see the following screen:



Next, add the following code to the password reset confirm page

password\_reset\_confirm.html

{% extends 'base.html' %}

{% block title %}Enter new password{% endblock title %}

{% block content %}

<h1>Set a new password!</h1>

<form method="POST">

{% csrf\_token %}

{{ form.as\_p }}

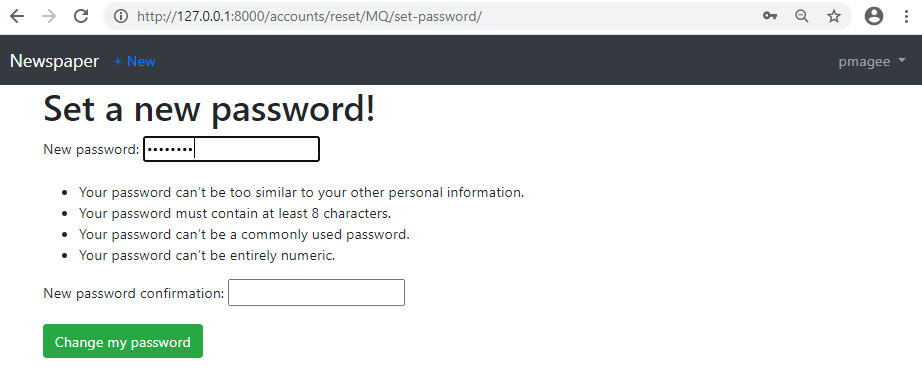
<input class="btn btn-success" type="submit" value="Change my password">

</form>

{% endblock content %}

In the command line grab the URL link from the email outputted to the console– mine was http://127.0.0.1:8000/users/reset/MQ/4yy-2dde95cd69631c8d938e/–and you will see

the following.



Finally copy the following code into the password\_reset\_complete.html page:

{% extends 'base.html' %}

{% block title %}Password reset complete{% endblock title %}

{% block content %}

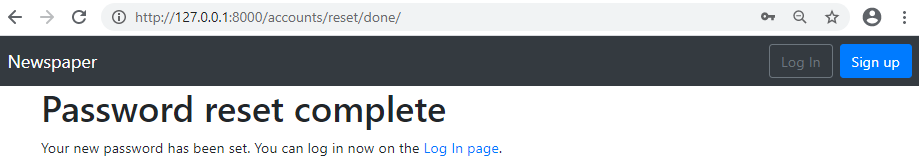
<h1>Password reset complete</h1>

<p>Your new password has been set. You can log in now on the

<a href="{% url 'login' %}">Log In page</a>.</p>

{% endblock content %}

You can view it at http://127.0.0.1:8000/accounts/reset/done/.



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 6 commit”

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