**Worksheet 18**

**SignUp Confirmation Email**

A confirmation email is a transactional email sent to customers after a certain condition is triggered, such as: placing an order, booking tickets, subscribing to a newsletter, opening an account, etc.

Confirmation emails are the messages you send to welcome a user after they’ve signed up for a service or updated their profile information.

It is recommended to send new customers a confirmation email to let them know that you received their order and that it is being processed.

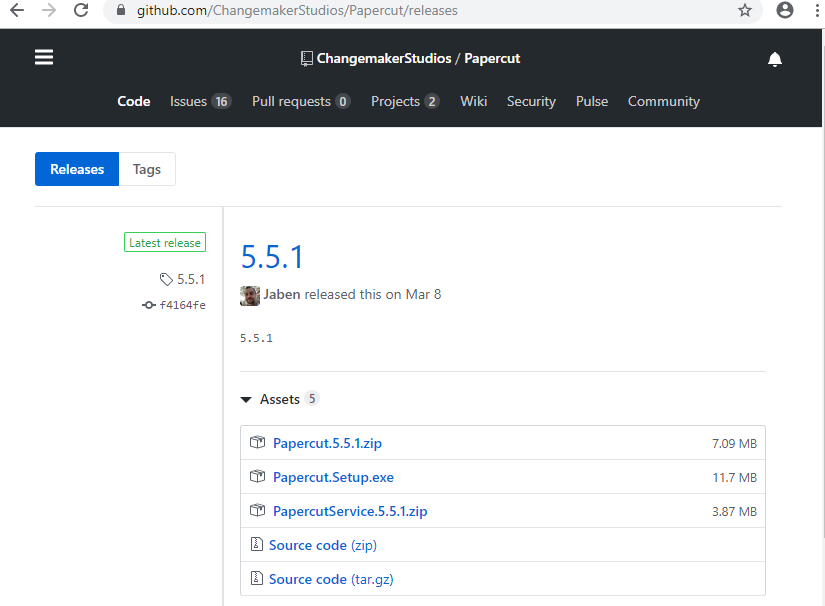
In this exercise you will learn how to send an email to the customer after they signup.

Sending email in development can be very challenging when building web applications as we must deal with restrictions imposed on us by the firewall in the college. The firewall prevents us from using external mail servers in our code to send emails.

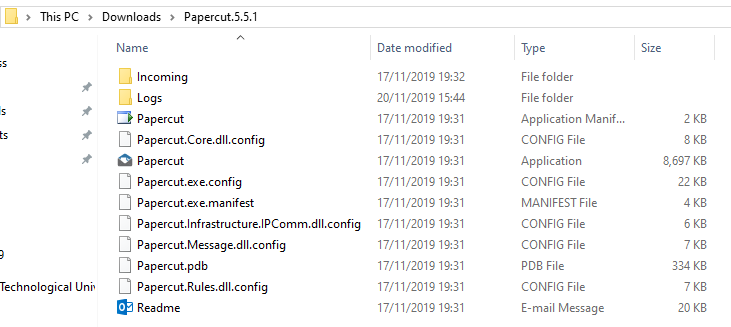
In order to address this issue we will use Papercut which provides us with a local SMTP server that captures all email sent through it, instead of sending it out to the actual recipients.

This server can be found at:

<https://github.com/ChangemakerStudios/Papercut/releases>



Download the zip file and extract it. Locate the application file shown below and run it.



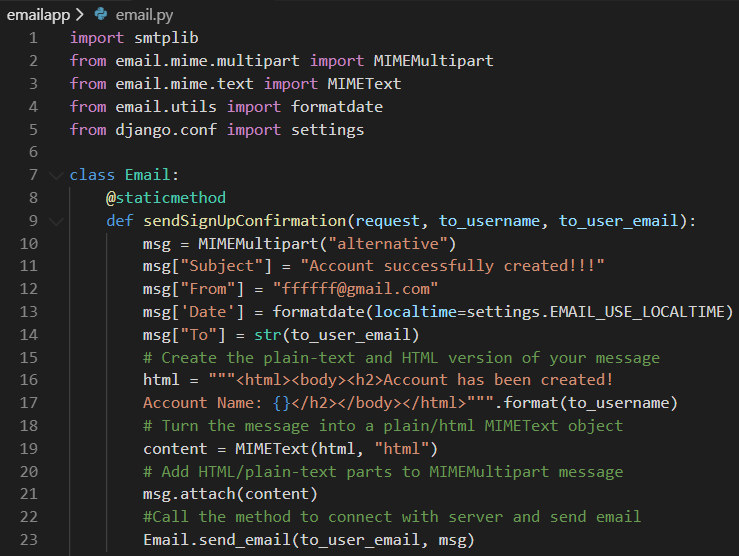
The Papercut application minimizes to the system tray shown below instead of having an always present taskbar icon.



Download & extract the file emailproject.zip from Moodle and open it in VS Code

You will add code to this project to implement the email functionality:

Create a new file inside the **emailapp** called **email.py**:



In this file we use the @staticmethod decorator to create this static method.

This method sendSignUpConfirmation will create the email message that we want to send to the new user.

Python provides us with the smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP listener daemon. Python’s built-in email package allows us to structure emails, which can then be transferred with smtplib

MIME is a kind of add on or a supplementary protocol which allows non-ASCII data to be sent through SMTP. It allows the users to exchange different kinds of data files on the Internet: audio, video, images, application programs as well.

When creating email messages, you need to set the Content-Type to multipart/alternative when sending HTML and text or multipart/mixed when sending text and attachments.

On line 10 we use multipart/alternative as we are sending HTML and text

Lines 11-14: We initialise the email message components Subject, From, Date and To

Lines 16-17: We create the HTML version of our greeting message which will include the name of the new user

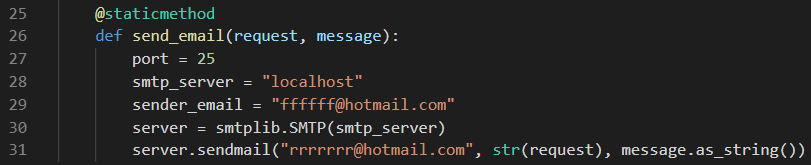
Line 19: We convert the HTML into a MIMEText object

Line 21: We then add the MIMEText object to the MIMEMultipart message

Line 22: We call another stati method which will connect with the server and send the message

In the method above we first defined the plain-text and HTML message as string literals, and then stored them as plain/html MIMEText objects. These can then be added in this order to the MIMEMultipart("alternative") message and sent through your connection with the email server. Remember to add the HTML message after the plain-text alternative, as email clients will try to render the last subpart first.

Add the following code just after the sendSignUpConfirmation method:



Line 27: We set the port that we will use. Normally e-mail is port 25 and thankfully we do have access to open port 25 in the labs locally.

Line 28: We set the server to be localhost

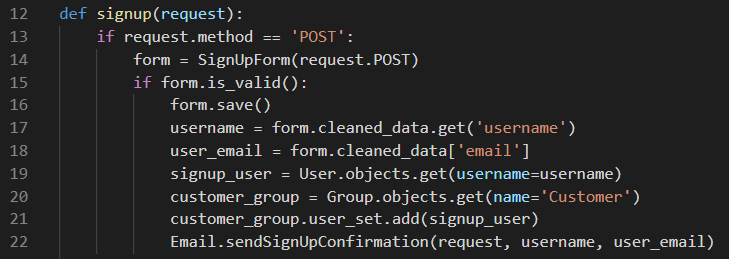
Line 29: We set the email of the sender (can be anything)

Line 30: We use SMTP is to connect to the mail server

Line 31: We send the message

Open **emailapp/views.py** and add the following code:



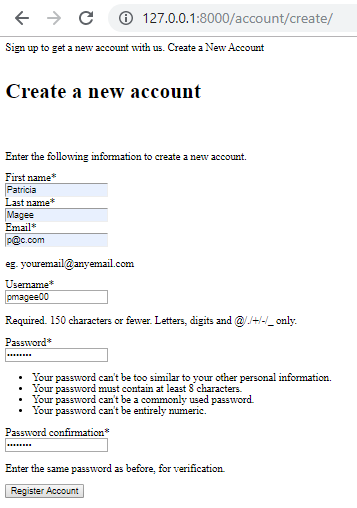


Save your project.

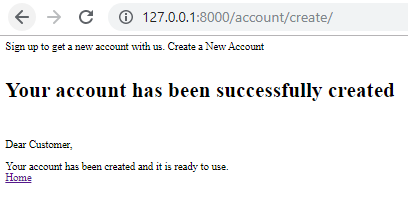
When you run the server, you will see a very basic home page with a link for SignUp



When you click the Sign Up link you will see the following form:



When we fill out the details in the form and press the Register Account button, we see that our account has been successfully created.



If we check the PaperCut application in the system tray:

