



الجمهورية العربية السورية
اللاذقية - جامعة تشرين
كلية الهندسة الكهربائية والميكانيكية
قسم هندسة الاتصالات والالكترونيات
السنة الخامسة: مشروع برمجة شبكات

تقديم

الاء محمد سماعيل 1442

سارة محمد شهلا 2391

فرح بسام احمد 2378

إشراف

د. مهند عيسى

إرسال بريد إلكتروني باستخدام SMTP

ملخص:

يتم استخدام بروتوكول نقل البريد البسيط (SMTP) كبروتوكول للتعامل مع نقل البريد الإلكتروني باستخدام Python. يتم استخدامه لتوجيه رسائل البريد الإلكتروني بين خوادم البريد الإلكتروني. إنه بروتوكول طبقة تطبيق يسمح للمستخدمين بإرسال بريد إلى آخر. يسترد المتلقي البريد الإلكتروني باستخدام البروتوكولات POP (Post Office Protocol) و IMAP (بروتوكول الوصول إلى الرسائل عبر الإنترنت). سنقوم ببناء برنامج باستخدام لغة بايثون لإرسال البريد الإلكتروني عبر البروتوكول smtp وسيكون أول بريد إلكتروني مرسل عبارة عن رسالة نصية وثاني بريد عبارة عن صفحة html.

الكلمات المفتاحية: بروتوكول نقل البريد البسيط smtp

Python send email using SMTP

Abstract

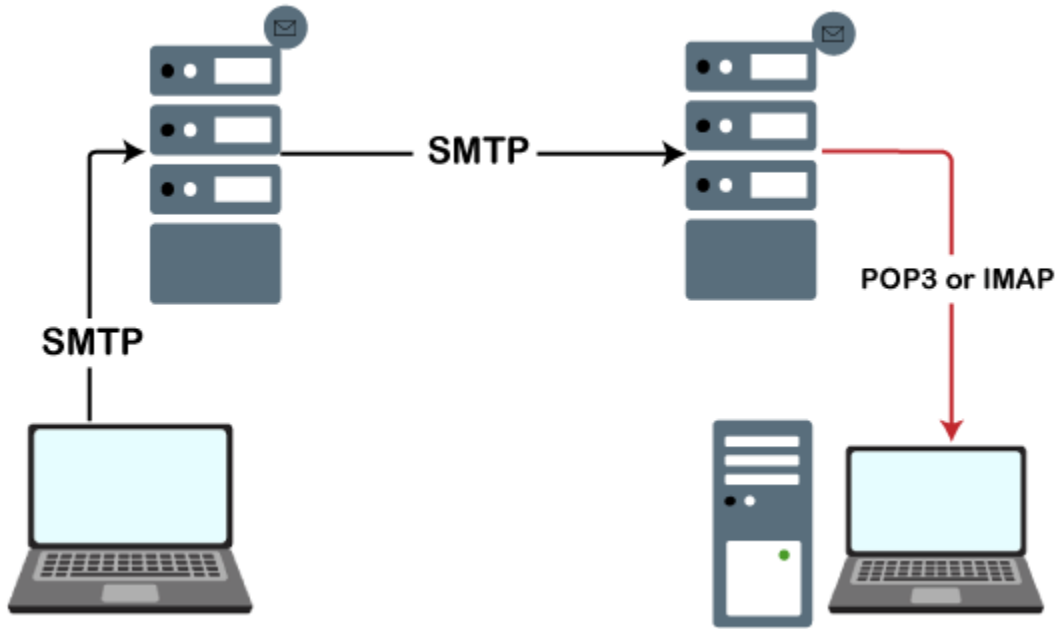
Simple Mail Transfer Protocol (SMTP) is used as a protocol for handling email transmission using Python. It is used to route email messages between email servers. It is an application layer protocol that allows users to send mail to another. The recipient retrieves the email using the protocols POP (Post Office Protocol) and IMAP (Internet Message Access Protocol). We are going to build a program using python language to send email via smtp protocol and the first email sent will be a text message and the second email will be an html page.

Keywords: smtp . simple mail transfer protocol

مقدمة:

يُعرّف البريد الإلكتروني (Electronic Mail) أو ما يُشار له بالاختصار (Email) ، بأنه عبارة عن خدمة يُمكن من خلالها إرسال واستقبال رسائل إلكترونية عبر شبكة اتصالات معينة وباستخدام أنواع مُختلفة من التطبيقات والبرامج، ويتم تبادل رسائل البريد الإلكتروني من وإلى أي شخص يمتلك عنوان بريد إلكتروني حول العالم، وقد تحتوي رسالة البريد الإلكتروني على ملفات مُضمنة بداخلها، أو صور، أو نصوص كتابية، حيث تحتوي معظم أنظمة البريد الإلكتروني على مُحرر أولي لتحرير النصوص، وإجراء مختلف التعديلات عليها كتغيير لون الخط، وحجمه، وغيرها من التعديلات الأخرى. يتم تخزين رسائل البريد الإلكتروني الواردة إلى المُستخدم فيما يُعرف بصندوق البريد الوارد (Inbox) ، والذي يجب أن يتفقدّه المُستخدم بشكل دوري للتحقق من تلقيه أية رسائل جديدة، ويُمكن ضبط العديد من أنظمة البريد الإلكتروني بحيث تُصدر تنبيهات عند ورود أية رسائل جديدة، والتي يتم حفظها على الجهاز، أو يتم حذفها، أو إعادة توجيهها بهدف إرسالها إلى شخص آخر، كما يُمكن طباعتها على شكل نسخة ورقية.

يتم استخدام بروتوكول نقل البريد البسيط (SMTP) كبروتوكول للتعامل مع نقل البريد الإلكتروني باستخدام Python. يتم استخدامه لتوجيه رسائل البريد الإلكتروني بين خوادم البريد الإلكتروني. إنه بروتوكول طبقة تطبيق يسمح للمستخدمين بإرسال بريد إلى آخر. يسترد المتلقي البريد الإلكتروني باستخدام البروتوكولات POP (Post Office Protocol) و IMAP (بروتوكول الوصول إلى الرسائل عبر الإنترنت).



الشكل 1 البنية العامة لإرسال واستقبال البريد الإلكتروني

أهمية البحث:

تكمن أهمية البحث في إنشاء برنامج بلغة بايثون يتم من خلاله إرسال الايميل وبالتالي امكانية تحكم أكبر في خصائص الرسالة بطريقة لا تتيحها تطبيقات ارسال البريد الالكتروني المعروفة.

أدوات البحث:

- 1- بيئة بايثون
- 2- مكتبة smtp
- 3- بيئة التطوير الشاملة pycharm

منهجيات البحث:

1- ما هو بروتوكول نقل البريد البسيط SMTP؟

SMTP ، الذي يرمز إلى Simple Mail Transfer Protocol ، هو بروتوكول بريد إلكتروني يُستخدم لإرسال رسائل البريد الإلكتروني من حساب بريد إلكتروني إلى حساب آخر عبر الإنترنت. بروتوكولات البريد الإلكتروني هي مجموعات من القواعد التي تسمح لعملاء وحسابات بريد إلكتروني مختلفة بتبادل المعلومات بسهولة ، و SMTP هو أحد أكثر البروتوكولات شيوعًا إلى جانب POP و IMAP. وهو أيضًا البروتوكول المخصص الوحيد لإرسال رسائل البريد الإلكتروني. تعتمد معظم برامج البريد الإلكتروني - بما في ذلك Outlook و Apple Mail و Gmail و Yahoo Mail - على SMTP "لدفْع" أو إرسال الرسائل من المرسل إلى المستلم.

2- ما هو خادم SMTP؟

مثل جميع الخوادم، يعد خادم SMTP تطبيقاً يوفر خدمة لتطبيقات أخرى داخل شبكة تسمى العملاء. على وجه التحديد، يتعامل خادم SMTP مع إرسال البريد الإلكتروني واستلامه وترحيله.

يمكنك التفكير في الخوادم على أنها مكاتب بريد حقيقية. عندما ترسل رسالة من المدينة "A" إلى المدينة "B" ، فإنها تصل أولاً إلى مكتب بريد محلي في المدينة "A" ؛ هنا ، تتم معالجتها وإرسالها إلى مكتب البريد في المدينة B ، وهو المسؤول عن تسليمها إلى وجهتها النهائية.

يحدث الشيء نفسه مع خوادم SMTP — على الرغم من أن العملية تستغرق بضع دقائق على الأكثر بدلاً من أن تستغرق أياماً. ربما تكون قد صادفت أيضاً مصطلح منفذ SMTP. هذه هي نقاط نهاية الاتصال التي تتعامل مع نقل بيانات البريد الإلكتروني عبر SMTP أثناء انتقالها عبر شبكة ، من خادم إلى آخر. نحن نغطي تلك بالتفصيل هنا.

3- كيف يعمل SMTP؟

أفضل طريقة لشرح كيفية عمل SMTP هي مراجعة عملية الإرسال والقواعد والأوامر الفردية التي تشغله والأخطاء التي قد تواجهها.

بمجرد إنشاء خادم SMTP ، يمكن لعملاء البريد الإلكتروني الاتصال به والتواصل معه. عندما يضغط المستخدم "إرسال" في رسالة بريد إلكتروني ، يفتح عميل البريد الإلكتروني اتصال SMTP بالخادم حتى يتمكن من الإرسال. (تم بناء اتصال SMTP على شيء يسمى اتصال TCP ، والذي يرمز إلى بروتوكول التحكم في الإرسال.)

من هناك ، يستخدم عميل SMTP الأوامر لإخبار الخادم بما يجب فعله ونقل البيانات، مثل عنوان البريد الإلكتروني للمرسل وعنوان البريد الإلكتروني للمستلم ومحتوى البريد الإلكتروني. يتحقق وكيل نقل البريد أو وكيل نقل الرسائل (MTA) لمعرفة ما إذا كان كلا عنواني البريد الإلكتروني من نفس مجال البريد الإلكتروني ، مثل gmail.com :

إذا كانوا كذلك ، فإنه يرسل البريد الإلكتروني على الفور ، إذا لم يكن الأمر كذلك ، يستخدم الخادم نظام اسم المجال (DNS) لتحديد نطاق المستلم ثم إرساله إلى الخادم الصحيح.

بروتوكول نقل البريد البسيط (SMTP) هو بروتوكول اتصال قياسي عبر الإنترنت لنقل البريد الإلكتروني. تستخدم خوادم البريد ووكلاء نقل الرسائل الآخرون SMTP لإرسال رسائل البريد واستلامها. عادةً ما يستخدم عملاء البريد الإلكتروني على مستوى المستخدم SMTP فقط لإرسال الرسائل إلى خادم بريد للترحيل، وعادةً ما يرسلون البريد الإلكتروني الصادر إلى خادم البريد على المنفذ 587 أو 465. لاسترداد الرسائل ، IMAP (الذي حل محل POP3 الأقدم) هو غالباً ما تطبقه الخوادم القياسية ولكن المملوكة ملكية بروتوكولات خاصة، على سبيل المثال ، Exchange .ActiveSync

بدأت أصول SMTP في عام 1980، بناءً على المفاهيم المطبقة على ARPANET منذ عام 1971. وقد تم تحديثها وتعديلها وتمديدتها عدة مرات. إصدار البروتوكول الشائع الاستخدام اليوم له هيكل قابل للتوسعة بامتدادات مختلفة للمصادقة والتشفير ونقل البيانات الثنائية وعناوين البريد الإلكتروني الدولية. تستخدم خوادم SMTP عادةً بروتوكول التحكم في الإرسال على المنفذ رقم 25 (للنص العادي) و 587 (للاتصالات المشفرة).

4- أوامر SMTP الأساسية

كما ذكرنا سابقاً ، أوامر SMTP هي مجموعة من الرموز التي تعمل على نقل رسائل البريد الإلكتروني بين الخوادم. فيما يلي أوامر SMTP الأساسية التي يجب أن تكون على دراية بها:

HELO أو EHLO (مرحباً): هذا أمر حاسم لبدء عملية إرسال البريد الإلكتروني بالكامل. يقوم عميل البريد الإلكتروني بتعريف نفسه لخادم SMTP. إنها بداية محادثة وعادة ما تتضمن الخادم إرسال أمر HELO مرة أخرى مع اسم المجال/عنوان IP الخاص به.

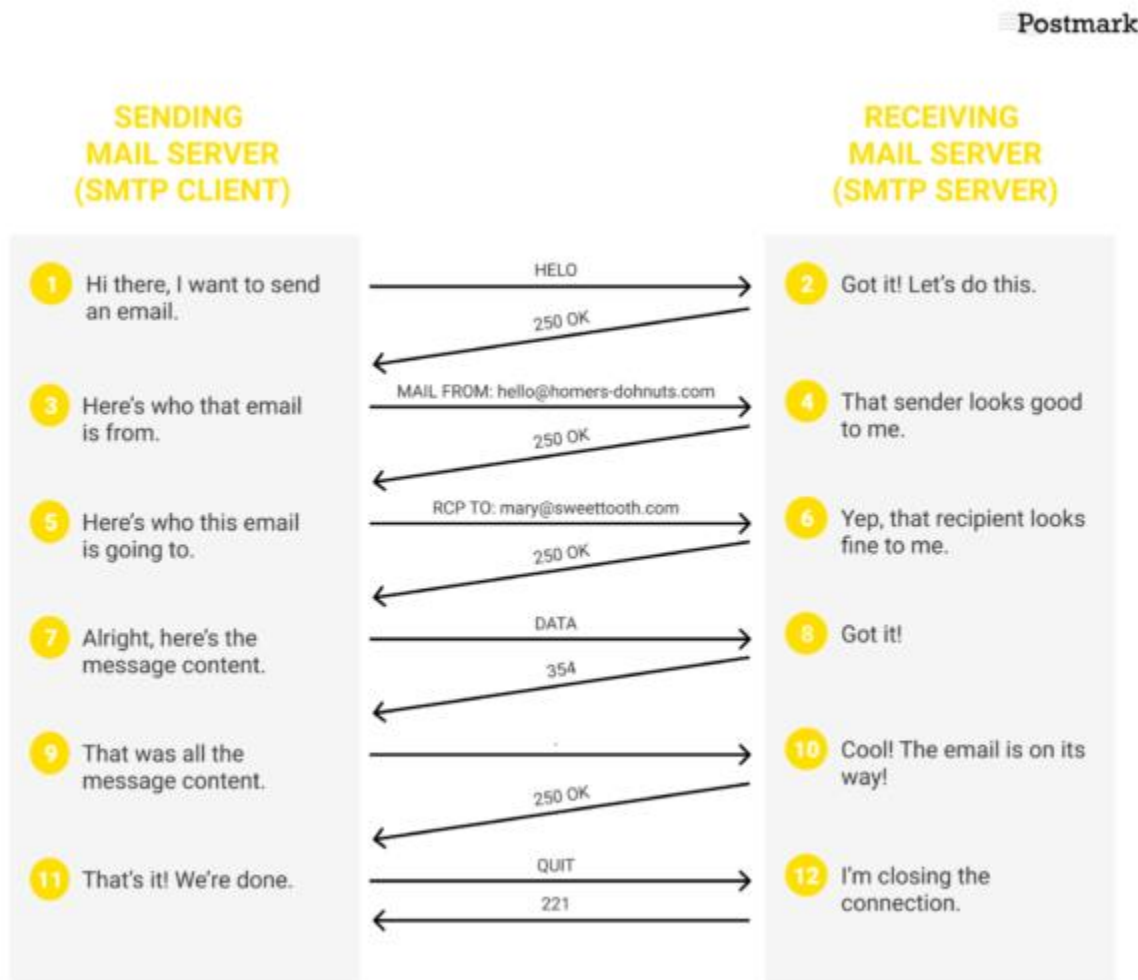
MAIL FROM: باتباع أمر التعريف، سيشترك المرسل الرمز الذي يحدد من هو البريد. يحدد هذا عنوان البريد الإلكتروني ويخبر خادم SMTP أن معاملة جديدة على وشك البدء. من هنا ، يقوم الخادم بإعادة تعيين كل شيء ويكون جاهزاً لقبول عنوان البريد الإلكتروني. بمجرد قبولها ، سيتم الرد عليها بـ 250 رمز رد موافق.

RCPT TO (المستلم إلى): يتبع الأمر التالي رمز الرد 250 OK الذي يحدد من يتم إرسال البريد الإلكتروني إليه. مرة أخرى ، يستجيب خادم SMTP بنفس الرمز ، وعند هذه النقطة يمكن إرسال أمر RCPT TO آخر باستخدام عنوان بريد إلكتروني مختلف للمستلم. يمكن أن يذهب هذا ذهاباً وإياباً عدة مرات حسب الحاجة اعتماداً على عدد الأشخاص الذين سيتلقون البريد الإلكتروني.

DATA (البيانات): يؤدي هذا إلى نقل البيانات بين العميل والخادم. سيتم نقل جميع محتويات الرسالة إلى خادم SMTP ، والذي سيستجيب برمز رد 345. يتم نقل محتويات الرسائل إلى الخادم ، حيث يتم إرسال نقطة واحدة في سطر بمفرده للإشارة إلى نهاية الرسالة. إذا تم قبوله وجاهزاً للتسليم ، يرسل الخادم 250 رمز موافق آخر. في هذه المرحلة ، تكون الرسالة في طريقها إلى المستلمين.

QUIT: عندما يتم إرسال البريد الإلكتروني ، يرسل العميل الأمر QUIT إلى الخادم، مما يؤدي إلى قطع الاتصال. إذا تم إغلاقه بنجاح ، سيرد الخادم برقم 221.

RSET (إعادة تعيين): يتم إرسال هذا الأمر إلى الخادم عندما يلزم إحباط معاملة البريد. لا يغلق الاتصال ، لكنه يعيد تعيين كل شيء ويزيل جميع البيانات السابقة حول البريد الإلكتروني والأطراف المعنية. يتم استخدام هذا هذا بشكل شائع عندما يكون هناك خطأ، مثل إدخال معلومات المستلم الخاطئة ، وتحتاج العملية إلى إعادة التشغيل. تخيل هذه الأوامر باعتبارها اللغة التي تسمح لخوادم البريد الإلكتروني بإجراء محادثة مع بعضها البعض. ستبدو محادثتهم إلى حد ما كما هو موضح بالشكل 2:



الشكل 2 محادثة عميل وخادم smtp

5- عنوان البريد الإلكتروني

يُعرّف عنوان البريد الإلكتروني (Email Address) بأنه عبارة عن اسم مُميز وفريد خاص بكل مُستخدم من مُستخدمي البريد الإلكتروني، ويتكون أيّ عنوان لبريد إلكتروني من ثلاثة أجزاء رئيسية، وهي كالآتي:

- اسم المُستخدم Username: وهو ذلك الجزء الذي يقع في بداية عنوان البريد الإلكتروني، وهو ذلك الجزء الذي يختاره المُستخدم بنفسه، والذي يُعبّر عن الاسم الحقيقي أو المُستعار للمُستخدم.
- رمز (@): وهو أحد الأجزاء الرئيسية التي يجب أن تكون موجودة في العنوان، وهو الذي يفصل بين اسم المُستخدم و اسم النطاق Domain name.
- اسم النطاق Domain name: وهو اسم النطاق أو المجال الذي ينتمي إليه عنوان البريد الإلكتروني.

القسم العملي:

عندما يستمع الخادم لاتصال TCP من العميل ، فإنه يبدأ الاتصال على المنفذ 587.

يوفر Python وحدة smtplib ، والتي تحدد كائن جلسة عميل SMTP المستخدم لإرسال رسائل البريد الإلكتروني إلى جهاز الإنترنت. لهذا الغرض ، يتعين علينا استيراد وحدة smtplib باستخدام تعليمة الاستيراد.

```
import smtplib
```

يتم استخدام كائن SMTP لنقل البريد الإلكتروني. يتم استخدام التعليمة التالية لإنشاء كائن smtplib.

```
import smtplib
```

```
smtpObj = smtplib.SMTP(host, port, local_hostname)
```

يقبل البرامترات التالية.

المضيف: هو اسم مضيف الجهاز الذي يقوم بتشغيل خادم SMTP الخاص بك.

المنفذ: هو رقم المنفذ الذي يستمع عليه الجهاز المضيف لاتصالات SMTP. هو 25 بشكل افتراضي.

local_hostname: إذا كان خادم SMTP يعمل على جهازك المحلي ، فيمكننا ذكر اسم مضيف الجهاز المحلي. يتم استخدام طريقة الإرسال () لكائن SMTP لإرسال البريد إلى الجهاز المطلوب. الصيغة الواردة أدناه.

إرسال بريد إلكتروني باستخدام Python

الخطوة 1: استيراد مكتبات البريد الإلكتروني

تحتوي Python على وحدات مدمجة يمكننا استخدامها لإرسال رسائل البريد الإلكتروني. هناك نوعان من الوحدات التي سنحتاجها للقيام بذلك وهما `smtplib` و `email.message`.

SMTP (بروتوكول نقل البريد البسيط) هو بروتوكول اتصال يستخدم لإرسال رسائل البريد الإلكتروني. تسمح لنا وحدة `python smtplib` بتعريف عميل SMTP يمكننا من خلاله إرسال رسائل البريد الإلكتروني.

نستخدم وحدة `python email.message` لتجميع بنية البريد الإلكتروني التي تتضمن موضوع البريد الإلكتروني وعنوانه ومحتواه. من وحدة `email.message` ، سنحتاج فقط إلى الكلاس `EmailMessage`.

```
import smtplib
from email.message import EmailMessage
```

الخطوة 2: تكوين حقول رأس البريد الإلكتروني

أولاً ، باستخدام الكلاس `EmailMessage` ، سننشئ كائنًا يحتوي على جميع تفاصيل رسالة البريد الإلكتروني.

```
# create an email message object
message = EmailMessage()
```

بعد ذلك ، يمكننا إضافة تفاصيل رأس رسالة البريد الإلكتروني لتحديد الموضوع وعنوان الإرسال وعنوان الاستلام كما هو موضح أدناه.

```
email_subject = "Hello From Communcation Engineer"
sender_email_address = "networkprogramming0@gmail.com"
receiver_email_address = ["alaa9705549@gmail.com", "shahlasara884@gmail.com"]
email_password = "bnvygusypogkbhne"

# configure email headers
message['Subject'] = email_subject
```

```
message['From'] = sender_email_address
message['To'] = receiver_email_address
```

الخطوة 3: تكوين نص رسالة البريد الإلكتروني

في هذا المثال ، ستكون محتويات الرسالة عبارة عن رسالة نصية عادية. يتم تعيين نص البريد الإلكتروني باستخدام التابع set_content في كائن البريد الإلكتروني على النحو التالي.

```
# set email body text
message.set_content("Hello, Our names is sara and farah and alaa")
```

الخطوة 4: تكوين خادم البريد الإلكتروني

خادم البريد الإلكتروني هو تطبيق يُستخدم لإرسال رسائل البريد الإلكتروني واستلامها. تستفيد هذه الخوادم من بروتوكول SMTP لإرسال رسائل البريد الإلكتروني إلى خوادم أخرى.

سيعتمد الخادم المراد استخدامه على مزود البريد الإلكتروني الخاص بنا. بالنسبة إلى gmail ، يكون خادم البريد smtp.gmail.com أو smtp.live.com بالنسبة إلى hotmail.

في مثالنا ، البريد الإلكتروني تم إرساله من حساب gmail ، لذلك نحتاج إلى استخدام smtp.gmail.com كخادمنا.

ملاحظة: يحظر gmail افتراضياً الوصول إلى عملاء جهات خارجية غير معروفين يصلون إلى حساب gmail الخاص بنا مثل برنامج python الخاص بنا. وبالتالي لكي يعمل برنامج python هذا، ننقل إلى إعدادات أمان حساب gmail ونفعل وصول التطبيقات الأقل أماناً.

إعداد آخر نحتاج إلى تهيئته لخادم SMTP الخاص بنا هو منفذ البريد. هذا هو منفذ الشبكة الذي سيستخدمه تطبيق python الخاص بالعمل لإرسال رسائل بريد إلكتروني إلى خادم آخر. سنستخدم منفذ SMTP 587 ، وهو المنفذ الافتراضي المستخدم لمعالجة إرسال رسائل البريد الإلكتروني عبر SMTP. كما أنه يستخدم تشفير TLS لضمان تسليم البريد الإلكتروني بشكل آمن.

```
email_smtp = "smtp.gmail.com"

# set smtp server and port
server = smtplib.SMTP(email_smtp, '587')
```

بعد تعريف الخادم ، نحتاج إلى بعض الخطوات الإضافية لإجراء الاتصال بخادم SMTP.

يتضمن ذلك تعريف أنفسنا بالخادم باستخدام التابع ehlo ثم بدء اتصال TLS الآمن بالخادم باستخدام starttls.

```
# identify this client to the SMTP server
server.ehlo()
# secure the SMTP connection
server.starttls()
```

الخطوة الخامسة: إرسال البريد الإلكتروني

الخطوة الأخيرة هي إرسال البريد الإلكتروني.

أولاً ، نحتاج إلى تسجيل الدخول إلى بريدنا الإلكتروني SMTP باستخدام تابع تسجيل الدخول. نمرر عنوان البريد الإلكتروني وكلمة المرور المستخدمين عند تسجيل الدخول إلى حساب البريد الإلكتروني الخاص بنا.

بعد ذلك ، يمكننا ببساطة بدء إرسال البريد الإلكتروني باستخدام send_message وتمرير الكائن الذي يحتوي على محتوى الرسالة الذي قمنا بتكوينه مسبقاً

أخيراً نستخدم quit لإنهاء الاتصال بخادم SMTP.

```
# login to email account
server.login(sender_email_address, email_password)
# send email
server.send_message(message)
# close connection to server
server.quit()
```

نقوم بتشغيل تطبيق python الخاص بك ويجب أن يتلقى عنوان البريد الإلكتروني للمستلم الرسالة من حساب البريد الإلكتروني للمرسل!

إرسال محتوى HTML بالبريد الإلكتروني

هذا لتوضيح كيفية تكوين نص البريد الإلكتروني الخاص بك لمحتوى HTML. يظل إجراء إعداد خادم البريد الإلكتروني كما هو.

سنقرأ أولاً محتويات ملف يحتوي على HTML ثم نرسل المحتويات بالبريد الإلكتروني.

سيحتوي ملف يسمى message.html على بعض HTML البسيط ، وهنا المحتويات.

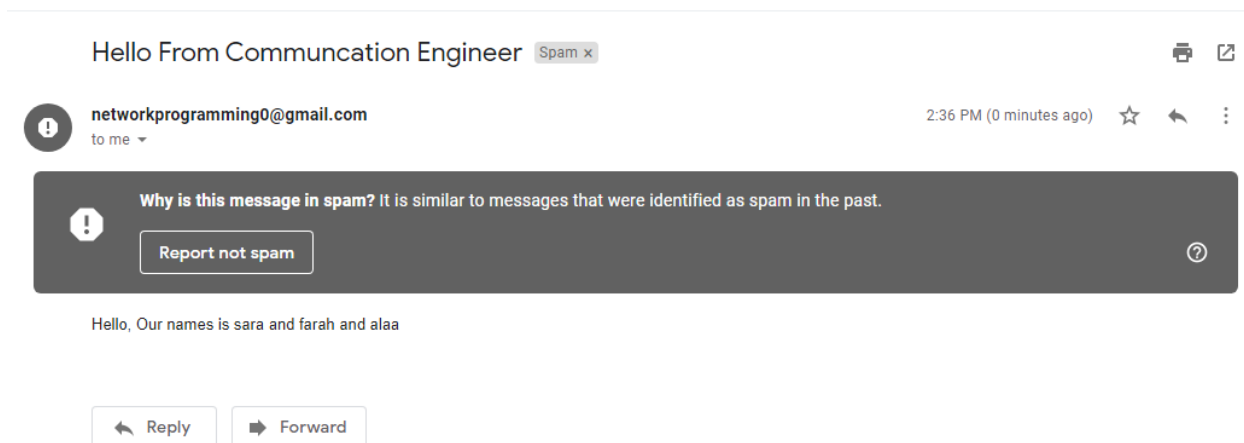
```
<!DOCTYPE html>
<html>
  <head>
  </head>
<body>
  <h1>Hello, Our names is sara and farah and alaa</h1>
</body>
</html>
```

سنقرأ برنامج python الخاص بنا أولاً محتويات الملف. ثم نضيف المحتويات إلى كائن الرسالة باستخدام تابع set_content كما فعلنا عند ضبط النص العادي. الاختلاف الوحيد هو أننا نحدد أن المحتوى من نوع HTML باستخدام الخيار "html" = subtype.

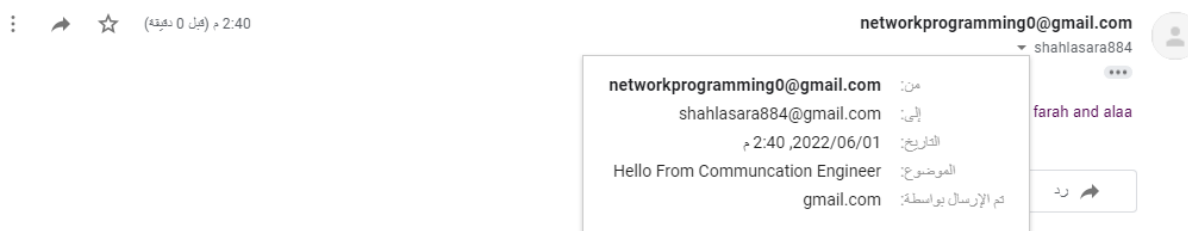
```
with open('message.html', 'r') as file:
    file_content = file.read()
# add message content as html type
message.set_content(file_content, subtype='html')
```

تظل جميع أجزاء البرنامج الأخرى كما هي. نقوم بتشغيل برنامج python وسيتلقى عنوان البريد الإلكتروني للمستلم الرسالة ، هذه المرة بتنسيق HTML!

النتائج والمناقشة:



الشكل 3 الرسالة الواردة



الشكل 4 الرسالة المرسلية

networkprogramming0@gmail.com
to me

2:43 PM (0 minutes ago) ☆ ↶ ⋮



Why is this message in spam? It is similar to messages that were identified as spam in the past.

Report not spam



Hello, Our names is sara and farah and alaa

↶ Reply

➦ Forward

mail. OK No thanks ✕

الشكل 5 صفحة ال html المستلمة

⋮ ↶ ☆ 2:44 م (قبل 0 دقيقة)
المزيد

networkprogramming0@gmail.com
shahlasara884



Hello, Our names is sara and farah and alaa

↶ إعادة توجيه

➦ رد

الشكل 6 صفحة ال html المرسله

الاستنتاجات:

قدمت بايثون العديد من التسهيلات في جميع المجالات وذلك من خلال مكتباتها وفي هذا المشروع استخدمنا مكتبة `smtplib` الخاصة بالتعامل مع البريد الالكتروني ورأينا كيف يمكن ارسال بريد الكتروني بكل بساطة ومرونة وأيضاً إمكانية ارسال ملفات الـ `html` هي ميزة جيدة وخاصة عند الترويج لبعض المواقع فبإمكاننا ارسال صفحة من الموقع وتظهر عند المستلم كصفحة بتنسيق جميل.

المراجع:

- 1- <https://docs.python.org/3/library/smtplib.html>
- 2- Goerzen, J. (2004). Simple Message Transport Protocol. In Foundations of Python Network Programming (pp. 197-210). Apress, Berkeley, CA.
- 3- Jain, R., Jain, N., & Kumar, P. (2022). Home Security Using Machine Learning (No. 7697). EasyChair.
- 4- Delaney, R. (2012). Python scripts as a replacement for bash utility scripts. Linux Journal, 2012(223), 1.