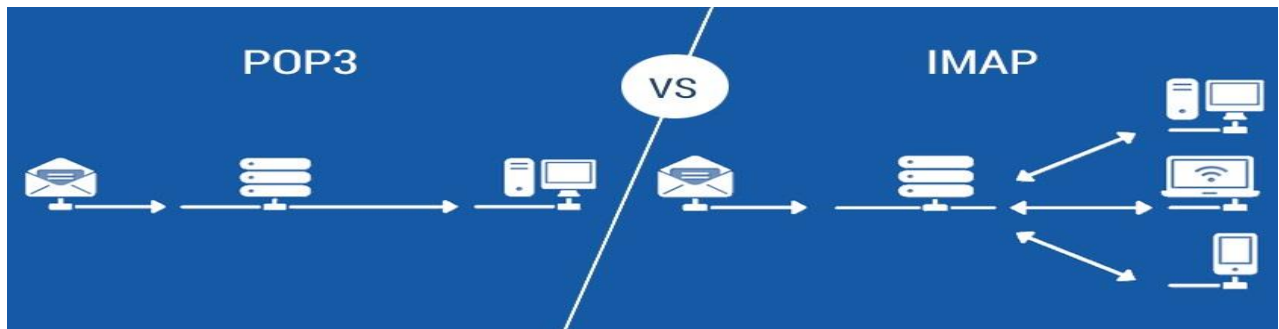




NETWORK PROGRAMMING PROJECT

Department of communication and electrical engineering

م



بإشراف الدكتور: مهند عيسى

إعداد الطلاب:

رهام معن حسن

نور راجح الحسن

نور حسن الحسن



مقدمة

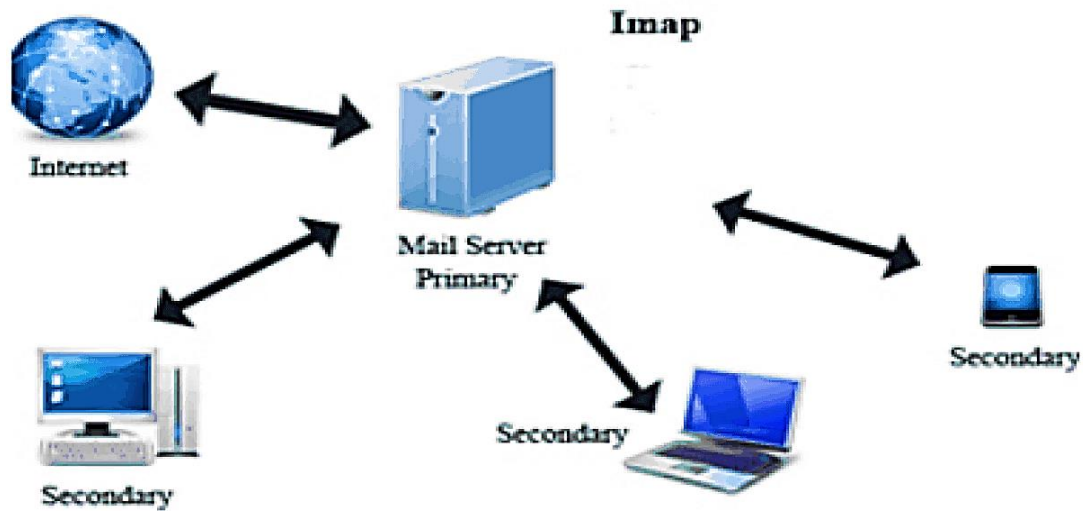
يتيح بروتوكول "IMAP" الوصول إلى رسائل البريد الإلكتروني الخاصة أينما كان

المستخدم، وفي كثير من الأحيان يتم الوصول إليه عبر [الإنترنت](#)، كما يتم حفظ رسائل البريد

الإلكتروني على الخوادم وعندما يتم التحقق من صندوق الوارد يعمل عميل البريد الإلكتروني

الخاص بالاتصال بالخادم لتوصيل الرسائل، عندما تقرأ رسالة بريد إلكتروني باستخدام

بروتوكول "IMAP" فأنت لا تقوم فعلياً بتنزيلها أو تخزينها على جهاز الكمبيوتر الخاص.



الشكل ١ : الوصول لمخدم البريد الالكتروني

ما هو بروتوكول IMAP:

بروتوكول الوصول إلى الرسائل عبر الإنترنت "IMAP" هو بروتوكول قياسي لإنشاء بريد إلكتروني على خادم صغير من مستخدم محلي، وإنه بروتوكول إنترنت بطبقة تطبيق يستخدم بروتوكولات طبقة النقل الأساسية لإنشاء خدمات اتصال من مضيف إلى مضيف للتطبيقات، ويتيح ذلك استخدام خادم بريد بعيد وموقع المنفذ المعروف لـ "IMAP" هو "143".

"IMAP" هي اختصار لـ "Internet Message Access Protocol".

أساسيات بروتوكول IMAP:

تسمح بنية "IMAP" للمستعملين بإرسال واستلام رسائل البريد الإلكتروني من خلال خادم صغير دون دعم من جهاز معين، وهذا النوع من الوصول إلى البريد الإلكتروني هو الهدف للمسافرين الذين يتلقون أو يردون على رسائل البريد الإلكتروني من سطح المكتب أو نظام المكتب في المنزل، كما تُعرف هذه الطريقة أيضاً باسم:

بروتوكول الوصول إلى البريد التفاعلي.

بروتوكول الوصول إلى بريد الإنترنت.

بروتوكول الوصول المؤقت إلى البريد.

5th

يمكن استخدام "IMAP" على نطاق واسع ولكنه أقل أهمية، حيث يتم إرسال الكثير من البريد الإلكتروني عبر الواجهات المستندة إلى الويب بما في ذلك "Gmail" و "Hotmail" و "Yahoo Mail"، وإنها طريقة للوصول إلى رسائل البريد الإلكتروني على الخادم دون الحاجة إلى تنزيلها على القرص الصلب المحلي وهذا هو الفرق الرئيسي بين "IMAP" و بروتوكول بريد إلكتروني مشهور آخر يُعرف باسم "POP3".

يحتاج "POP3" إلى العملاء لتنزيل الرسائل على محركات الأقراص الثابتة قبل قراءتها، وتتمثل فائدة استخدام خادم بريد "IMAP"، في أنه يمكن للعملاء التحقق من بريدهم من أجهزة كمبيوتر مختلفة وعرض الرسائل المماثلة باستمرار، وذلك لأن الرسائل تستمر على الخادم قبل أن يختار العميل تنزيلها من محرك الأقراص المحلي الخاص به.

نظراً لأن العالم أصبح أكثر قدرة على التنقل من أي وقت مضى أصبح الوصول عبر "IMAP" أكثر شيوعاً وأدى انتشار الهواتف الذكية، وأجهزة الكمبيوتر المحمولة والأجهزة اللوحية وغيرها من الأجهزة إلى زيادة الطلب على "IMAP" أكثر من أي وقت مضى.

بينما سيظل بروتوكول "POP" شائعاً لدى الأشخاص الذين يصلون إلى بريدهم الإلكتروني عبر جهاز واحد أو جهازين فقط وأولئك الذين لديهم اتصالات بطيئة بالإنترنت، فمن المؤكد أن "IMAP" سيظل البروتوكول المفضل لمعظم الأشخاص المشغولين اليوم.

"POP" هي اختصار لـ "Post Office Protocol".

مبدأ عمل بروتوكول IMAP:

أسهل طريقة لفهم كيفية عمل بروتوكول "IMAP" هي التفكير فيه كوسيط بين عميل البريد الإلكتروني وخادم البريد الإلكتروني، كما تُستخدم خوادم البريد الإلكتروني دائماً عند إرسال رسائل البريد الإلكتروني واستلامها، ومع "IMAP" يظلون على الخادم ما لم تحذفهم منه صراحة وعندما تقوم بتسجيل الدخول إلى عميل بريد إلكتروني مثل "Microsoft Outlook" فإنه يتصل بخادم البريد الإلكتروني باستخدام "IMAP".

ثم يتم عرض رؤوس جميع رسائل البريد الإلكتروني الخاصة، وإذا اخترت قراءة رسالة فسيتم تنزيلها بسرعة حتى تتمكن من رؤيتها، لا يتم تنزيل رسائل البريد الإلكتروني إلا إذا كنت بحاجة إلى فتحها.



مزايا استخدام بروتوكول IMAP:

- يسمح لك بالوصول إلى رسائل البريد الإلكتروني الخاصة بك من أي مكان، وعبر العديد من الأجهزة المختلفة التي تريدها.
- يقوم بتنزيل رسالة فقط عند النقر فوقها، ونتيجة لذلك لا يتعين عليك انتظار تنزيل جميع رسائلك الجديدة من الخادم قبل أن تتمكن من قراءتها.
- لا يتم تنزيل المرفقات تلقائياً باستخدام "IMAP" ونتيجة لذلك يمكن التحقق من رسائلك بسرعة أكبر والتحكم بشكل أكبر في المرفقات التي يتم فتحها.
- يمكن استخدام "IMAP" في وضع عدم الاتصال تماماً مثل بروتوكول "POP" ويمكن الاستمتاع بشكل أساسي بمزايا كلا البروتوكولين في واحد.
- يمكن استخدام "IMAP" في وضع عدم الاتصال، تماماً مثل بروتوكول POP
- البحث عن كلمات رئيسية في نص رسائل البريد الإلكتروني.
- إنشاء وإدارة صناديق بريد أو مجلدات متعددة.
- عرض العناوين لإجراء عمليات مسح مرئية سهلة لرسائل البريد الإلكتروني.

الفرق بين بروتوكول IMAP وبروتوكول POP:

- يعمل بروتوكول "POP" من خلال الاتصال بخادم البريد الإلكتروني الخاص بك وتنزيل جميع رسائلك الجديدة منه، وبمجرد تنزيلها تختفي من الخادم وإذا قررت التحقق من بريدك الإلكتروني من جهاز مختلف فلن تكون الرسائل التي تم تحميلها سابقاً متاحة لك.
- يعمل بروتوكول "POP" جيداً لأولئك الذين يقومون عموماً بفحص رسائل البريد الإلكتروني الخاصة بهم من جهاز واحد فقط، وأولئك الذين يسافرون أو يحتاجون إلى الوصول إلى بريدك الإلكتروني من أجهزة مختلفة أفضل حالاً مع خدمة البريد الإلكتروني المستندة إلى "IMAP".
- وبروتوكول الوصول إلى الرسائل عبر الإنترنت "IMAP" هو بروتوكول بريد يستعمل للانتقال إلى البريد الإلكتروني على خادم ويب مستقل من عميل داخلي، ويُعد "IMAP" و "POP3" بأنهما أكثر بروتوكولي بريد الإنترنت استعمالاً لاسترداد رسائل البريد الإلكتروني، وكلا البروتوكولين معتمدان من قبل جميع مستخدمي البريد الإلكتروني وخوادم الويب الحديثة.



بينما يُعتبر بروتوكول "POP3" أنه يتم الانتقال إلى البريد الإلكتروني من تطبيق واحد فقط، ويقدم بروتوكول "IMAP" الانتقال المتزامن لعدة مستخدمين، وهذا هو السبب في أن بروتوكول "IMAP" يناسبك أكثر إذا كنت ستنتقل إلى بريدك الإلكتروني من مواقع مختلفة أو إذا كانت رسائلك يتحكم بها عدة مستخدمين وكما يعمل بروتوكول "IMAP" على منفذين:

المنفذ "143": هو منفذ "IMAP" الافتراضي غير المشفر.

المنفذ "993": هو المنفذ الذي تحتاج إلى استخدامه إذا كنت تريد الاتصال باستخدام "IMAP" بأمان.

تطبيقات بروتوكول IMAP:

- بخلاف بروتوكول "POP" يتيح بروتوكول "IMAP" الوصول إلى رسائل البريد الإلكتروني وتنظيمها وقراءتها وفرزها دون الحاجة إلى تنزيلها أولاً، ونتيجةً لذلك فإن بروتوكول "IMAP" سريع وفعال للغاية.
- يحتفظ الخادم أيضاً بسجل لجميع الرسائل التي ترسلها مما يسمح لك بالوصول إلى رسائلك المرسلة من أي مكان.
- لا يقوم بروتوكول "IMAP" بنقل الرسائل من الخادم إلى جهاز الكمبيوتر الخاص، وبدلاً من ذلك يقوم بمزامنة البريد الإلكتروني الموجود على جهاز الكمبيوتر الخاص مع البريد الإلكتروني الموجود على الخادم.



الجزء العملي:

IMAP هو بروتوكول لاسترداد البريد الإلكتروني لا يقوم بتنزيل رسائل البريد الإلكتروني. إنه يقرأها فقط ويعرضها. هذا مفيد جدًا في حالة النطاق الترددي المنخفض. تُستخدم مكتبة جانب عميل Python المسماة imaplib للوصول إلى رسائل البريد الإلكتروني عبر بروتوكول imap.

يرمز IMAP إلى بروتوكول الوصول إلى بريد الإنترنت. تم اقتراحه لأول مرة في عام 1986.

يسمح بروتوكول IMAP لبرنامج العميل بمعالجة رسالة البريد الإلكتروني على الخادم دون تنزيلها على الكمبيوتر المحلي. يتم الاحتفاظ بالبريد الإلكتروني والاحتفاظ به بواسطة الخادم البعيد.

تمكننا من اتخاذ أي إجراء مثل التنزيل أو حذف البريد دون قراءة البريد ، كما أنها تمكننا من إنشاء مجلدات الرسائل البعيدة التي تسمى صناديق البريد ومعالجتها وحذفها.

يتيح بروتوكول IMAP للمستخدمين البحث في رسائل البريد الإلكتروني.

يسمح بالوصول المتزامن إلى علب بريد متعددة على خوادم بريد متعددة.

أوامر IMAP:

يصف الجدول التالي بعض أوامر IMAP:

S.N.	توصيف الأمر
1	IMAP_LOGIN هذا الأمر يفتح الاتصال.
2	CAPABILITY يطلب هذا الأمر سرد الإمكانيات التي يدعمها الخادم.
3	NOOP يتم استخدام هذا الأمر كإقتراع دوري للرسائل الجديدة أو تحديثات حالة الرسائل خلال فترة عدم النشاط.
4	SELECT يساعد هذا الأمر في تحديد صندوق بريد للوصول إلى الرسائل.
5	EXAMINE إنه مماثل لأمر SELECT باستثناء عدم السماح بإجراء أي تغيير على صندوق البريد.
6	CREATE يتم استخدامه لإنشاء صندوق بريد باسم محدد.
7	DELETE يتم استخدامه لحذف صندوق بريد باسم معين بشكل دائم.

8	RENAME يتم استخدامه لتغيير اسم صندوق البريد.
9	LOGOUT يقوم هذا الأمر بإعلام الخادم بأن العميل قد تم الانتهاء من الجلسة. يجب على الخادم إرسال استجابة BYE بدون علامات قبل الاستجابة "موافق" ثم إغلاق اتصال الشبكة.

في البرنامج التالي ،

نقوم بتسجيل الدخول إلى خادم gmail باستخدام بيانات اعتماد المستخدم.

ثم نختار عرض الرسائل في البريد الوارد. يتم استخدام حلقة for لعرض الرسائل التي تم جلبها واحدة تلو الأخرى .

وأخيرًا يتم إغلاق الاتصال.

Code 1

```
import imaplib
import pprint

imap_host = 'imap.gmail.com'
imap_user = 'username@gmail.com'
imap_pass = 'password'

# connect to host using SSL
imap = imaplib.IMAP4_SSL(imap_host)

## login to server
imap.login(imap_user, imap_pass)

imap.select('Inbox')

tmp, data = imap.search(None, 'ALL')
for num in data[0].split():
    tmp, data = imap.fetch(num, '(RFC822)')
    print('Message: {0}\n'.format(num))
    pprint.pprint(data[0][1])
    break
imap.close()
```



```
imap.logout()
```

code 2

```
import imaplib
import email
from email.header import decode_header
import webbrowser
import os

# account credentials
username = "youremailaddress@provider.com"
password = "yourpassword"

def clean(text):
    # clean text for creating a folder
    return "".join(c if c.isalnum() else "_" for c in text)

# number of top emails to fetch
N = 3

# create an IMAP4 class with SSL, use your email provider's IMAP server
imap = imaplib.IMAP4_SSL("imap.gmail.com")
# authenticate
imap.login(username, password)

# select a mailbox (in this case, the inbox mailbox)
# use imap.list() to get the list of mailboxes
status, messages = imap.select("INBOX")

# total number of emails
messages = int(messages[0])

for i in range(messages, messages-N, -1):
    # fetch the email message by ID
    res, msg = imap.fetch(str(i), "(RFC822)")
    for response in msg:
```

5th

```

if isinstance(response, tuple):
    # parse a bytes email into a message object
    msg = email.message_from_bytes(response[1])
    # decode the email subject
    subject, encoding = decode_header(msg["Subject"])[0]
    if isinstance(subject, bytes):
        # if it's a bytes, decode to str
        subject = subject.decode(encoding)
    # decode email sender
    From, encoding = decode_header(msg.get("From"))[0]
    if isinstance(From, bytes):
        From = From.decode(encoding)
    print("Subject:", subject)
    print("From:", From)
    # if the email message is multipart
    if msg.is_multipart():
        # iterate over email parts
        for part in msg.walk():
            # extract content type of email
            content_type = part.get_content_type()
            content_disposition = str(part.get("Content-Disposition"))
            try:
                # get the email body
                body = part.get_payload(decode=True).decode()
            except:
                pass
            if content_type == "text/plain" and "attachment" not in
content_disposition:
                # print text/plain emails and skip attachments
                print(body)
            elif "attachment" in content_disposition:
                # download attachment
                filename = part.get_filename()
                if filename:
                    folder_name = clean(subject)
                    if not os.path.isdir(folder_name):
                        # make a folder for this email (named after the subject)
                        os.mkdir(folder_name)

```



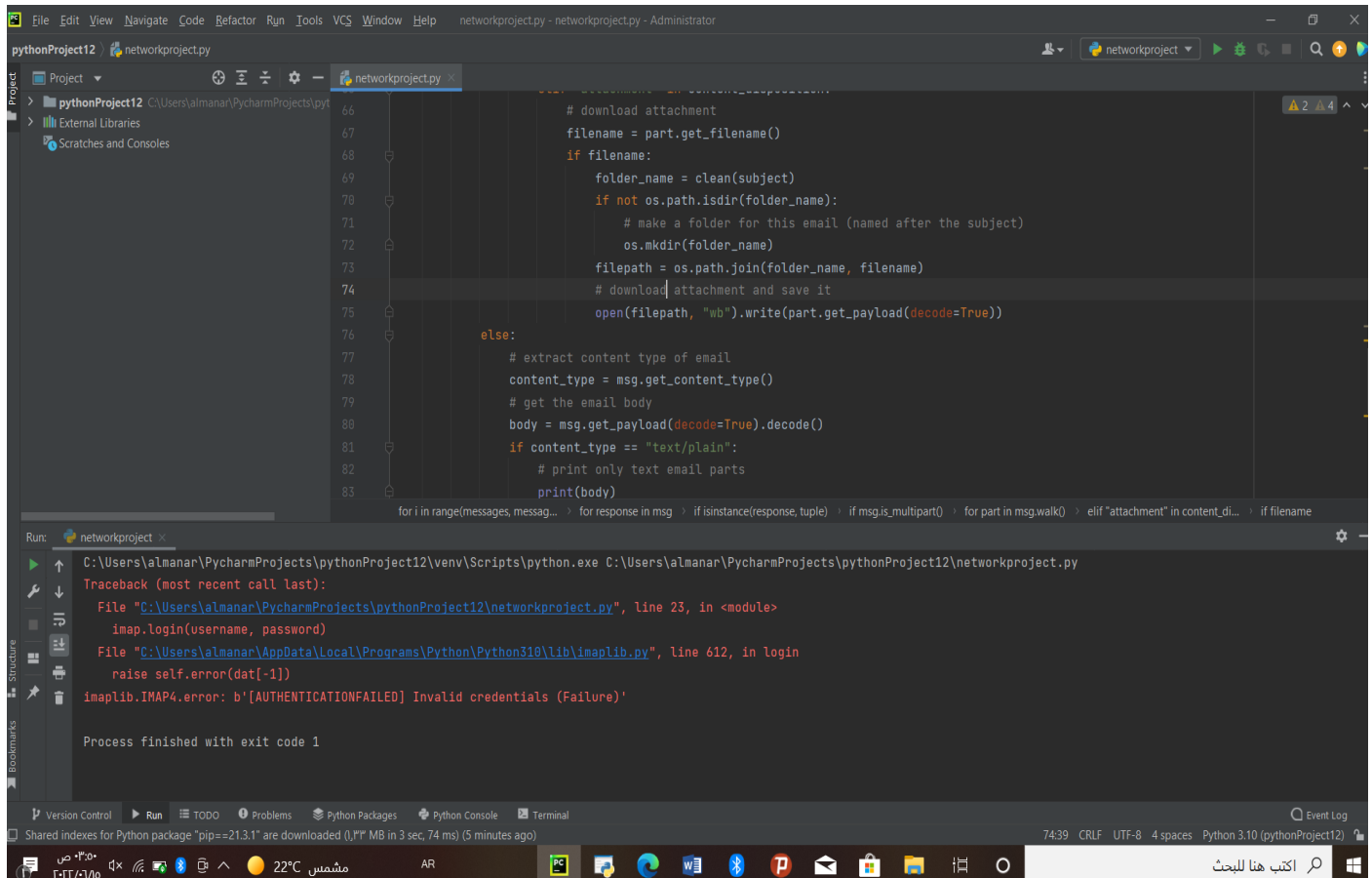
```

        filepath = os.path.join(folder_name, filename)
        # download attachment and save it
        open(filepath, "wb").write(part.get_payload(decode=True))

    else:
        # extract content type of email
        content_type = msg.get_content_type()
        # get the email body
        body = msg.get_payload(decode=True).decode()
        if content_type == "text/plain":
            # print only text email parts
            print(body)
        if content_type == "text/html":
            # if it's HTML, create a new HTML file and open it in browser
            folder_name = clean(subject)
            if not os.path.isdir(folder_name):
                # make a folder for this email (named after the subject)
                os.mkdir(folder_name)
            filename = "index.html"
            filepath = os.path.join(folder_name, filename)
            # write the file
            open(filepath, "w").write(body)
            # open in the default browser
            webbrowser.open(filepath)
        print("="*100)

# close the connection and logout
imap.close()
imap.logout()

```



```

66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
# download attachment
filename = part.get_filename()
if filename:
    folder_name = clean(subject)
    if not os.path.isdir(folder_name):
        # make a folder for this email (named after the subject)
        os.mkdir(folder_name)
    filepath = os.path.join(folder_name, filename)
    # download attachment and save it
    open(filepath, "wb").write(part.get_payload(decode=True))
else:
    # extract content type of email
    content_type = msg.get_content_type()
    # get the email body
    body = msg.get_payload(decode=True).decode()
    if content_type == "text/plain":
        # print only text email parts
        print(body)

for i in range(messages, messag...
for response in msg
if isinstance(response, tuple)
if msg.is_multipart()
for part in msg.walk()
elif "attachment" in content_di...
if filename

Run: networkproject
C:\Users\almanar\PycharmProjects\pythonProject12\venv\Scripts\python.exe C:\Users\almanar\PycharmProjects\pythonProject12\networkproject.py
Traceback (most recent call last):
  File "C:\Users\almanar\PycharmProjects\pythonProject12\networkproject.py", line 23, in <module>
    imap.login(username, password)
  File "C:\Users\almanar\AppData\Local\Programs\Python\Python310\lib\imaplib.py", line 612, in login
    raise self.error(dat[-1])
imaplib.IMAP4.error: b'[AUTHENTICATIONFAILED] Invalid credentials (Failure)'

Process finished with exit code 1

```

قائمة الأشكال

الشكل ١ : الوصول لمخدم البريد الالكتروني



المصطلحات المستخدمة

IMAP	Internet Message Access Protocol
POP	Post Office Protocol
Gmail	Google Mail

المراجع المستخدمة

1. <https://e3arabi.com/>
2. https://www.tutorialspoint.com/python_network_programming/