جامعة الإسكندرية
ALEXANDRIA UNIVERSITY
كلية الهندسة
Faculty of Engineering

Lab 3 Computer Networks

- ***Name:*** Saad El Dine Ahmed
- ***ID:*** 7370

Under Supervision Of:

- ***DR:*** Karim banwan

Email Client Application

Objective:

The objective of this project is to develop a Python-based email client application that can send and receive emails using the smtplib and imaplib libraries respectively. The application should be able to establish a TCP connection with a mail server, dialogue with the mail server using the SMTP and IMAP protocols, send an e-mail message to a recipient via the mail server, fetch the latest email from the mailbox, and finally close the TCP connection with the mail server.

File Structure:

The project is divided into the following files:

- send_email.py: Contains the send_email function for sending emails using the SMTP protocol.
- receive_email.py: Contains the receive_email function for receiving emails using the IMAP protocol.
- GUI.py: Contains the graphical user interface code using tkinter for the email client application.
- README.md: Contains instructions for setting up and running the email client application.
- email_client_project.md: This file, describing the project and its requirements.

Implemented Features:

1. Sending Emails

- The application allows users to send emails using the SMTP protocol.
- Users can specify the sender's email, password, recipient's email, subject, and body of the email.

- The send_email function in send_email.py handles the email sending process.

2. Receiving Emails

- Users can receive emails using the IMAP protocol.
- The application fetches the latest email from the mailbox and displays its body.
- The receive_email function in receive_email.py is responsible for fetching and displaying emails.

3. Graphical User Interface (GUI)

- The GUI for the email client application is developed using the tkinter library.
- Users can input their email, password, recipient's email, subject, and body in the GUI.
- Buttons are provided in the GUI to send and receive emails.

4. Error Handling

- The application includes error handling mechanisms to manage errors during the sending or receiving process.
- Appropriate error messages are displayed to the user in case of errors.

5. Push Notifications

- A push notification system using Plyer is implemented to alert the user when a new email arrives in the mailbox.
- Notifications are displayed to the user on the desktop.

6. Mail Server

- For testing purposes and to maintain privacy, the application uses mail.tm or an alternative mail server.

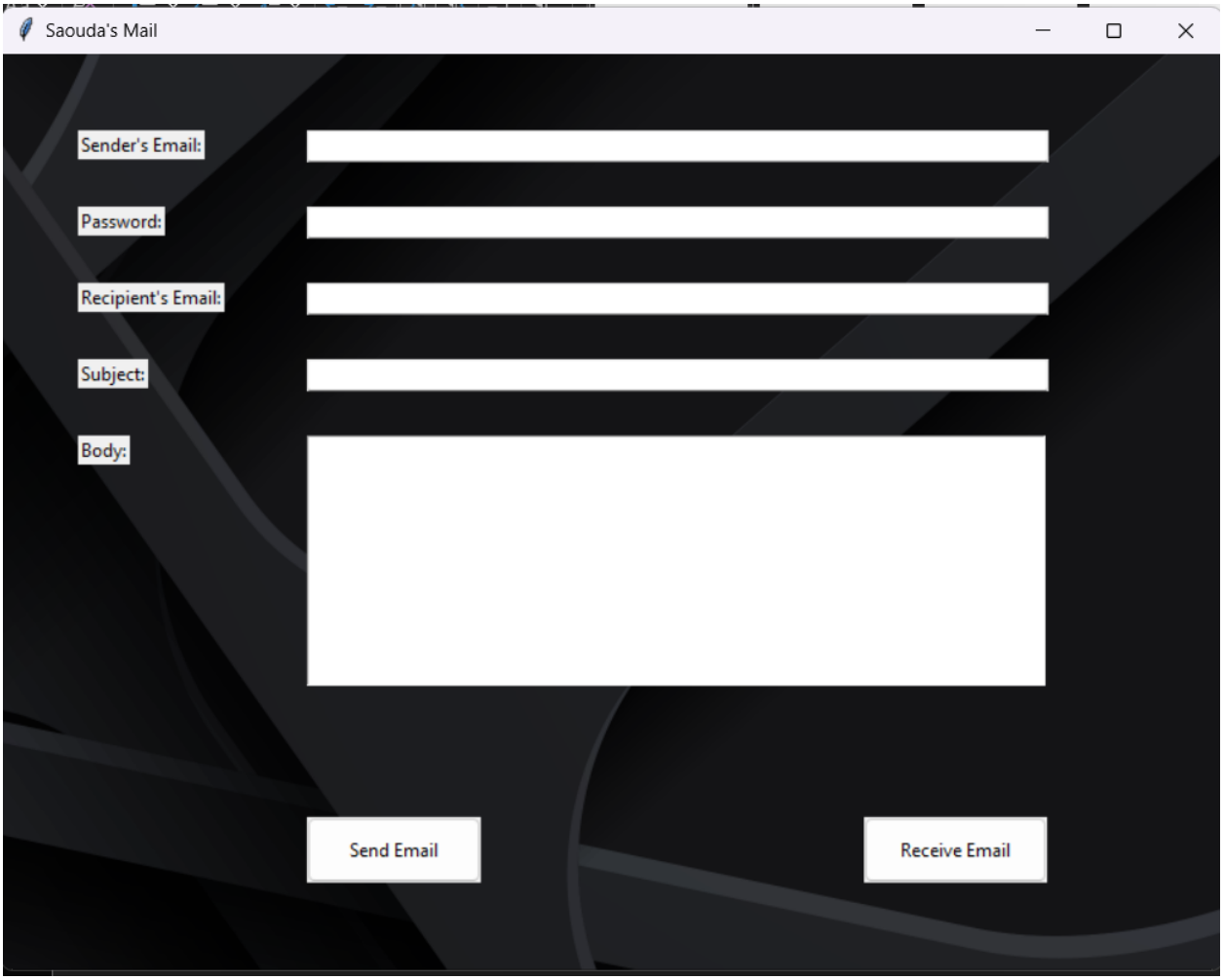
Sender Code:

```python
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

def send_email(sender_email, password, recipient_email, subject, body):
    # Create a MIMEText object
    msg = MIMEMultipart()
    msg['From'] = sender_email
    msg['To'] = recipient_email
    msg['Subject'] = subject
    msg.attach(MIMEText(body, 'plain'))

    # Connect to the SMTP server
    with smtplib.SMTP_SSL('smtp.gmail.com', 465) as server:
        server.login(sender_email, password)
        server.sendmail(sender_email, recipient_email, msg.as_string())
```

Receiver Code:

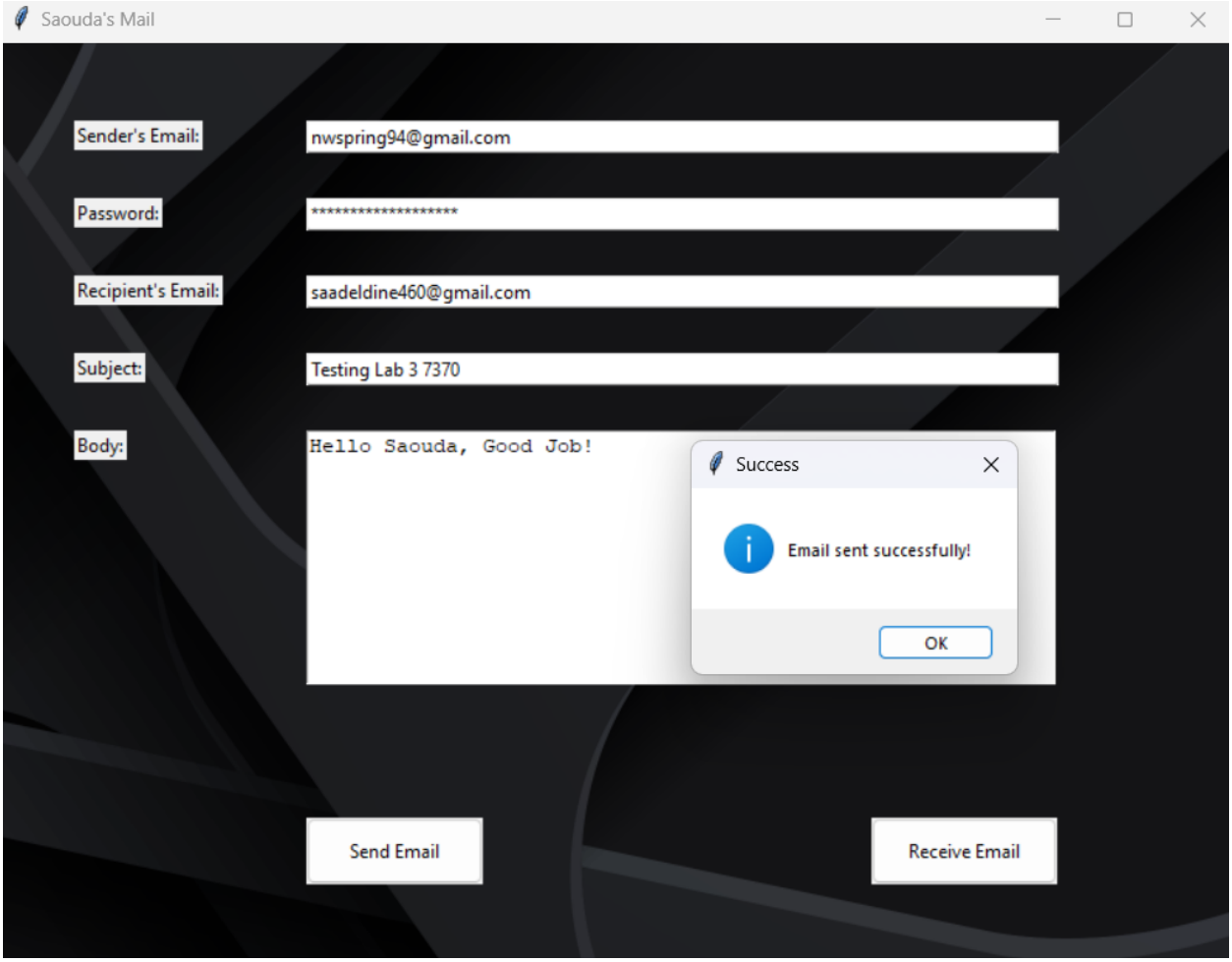
```python
import imaplib
import email

def receive_email(email, password):
    # Connect to the IMAP server
    mail = imaplib.IMAP4_SSL('imap.gmail.com')
    mail.login(email, password)
    mail.select('inbox')
    result, data = mail.search(None, 'ALL')
    latest_email_id = data[0].split()[-1]
    result, data = mail.fetch(latest_email_id, '(RFC822)')
    raw_email = data[0][1]
    msg = email.message_from_bytes(raw_email)
    mail.close()
    mail.logout()
    return msg.get_payload()
```

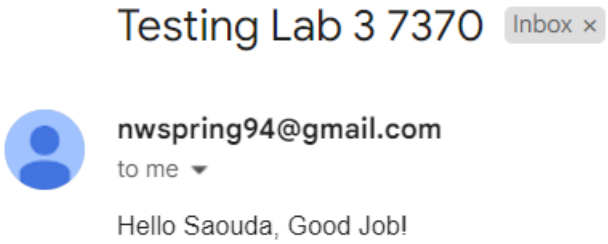
Application:



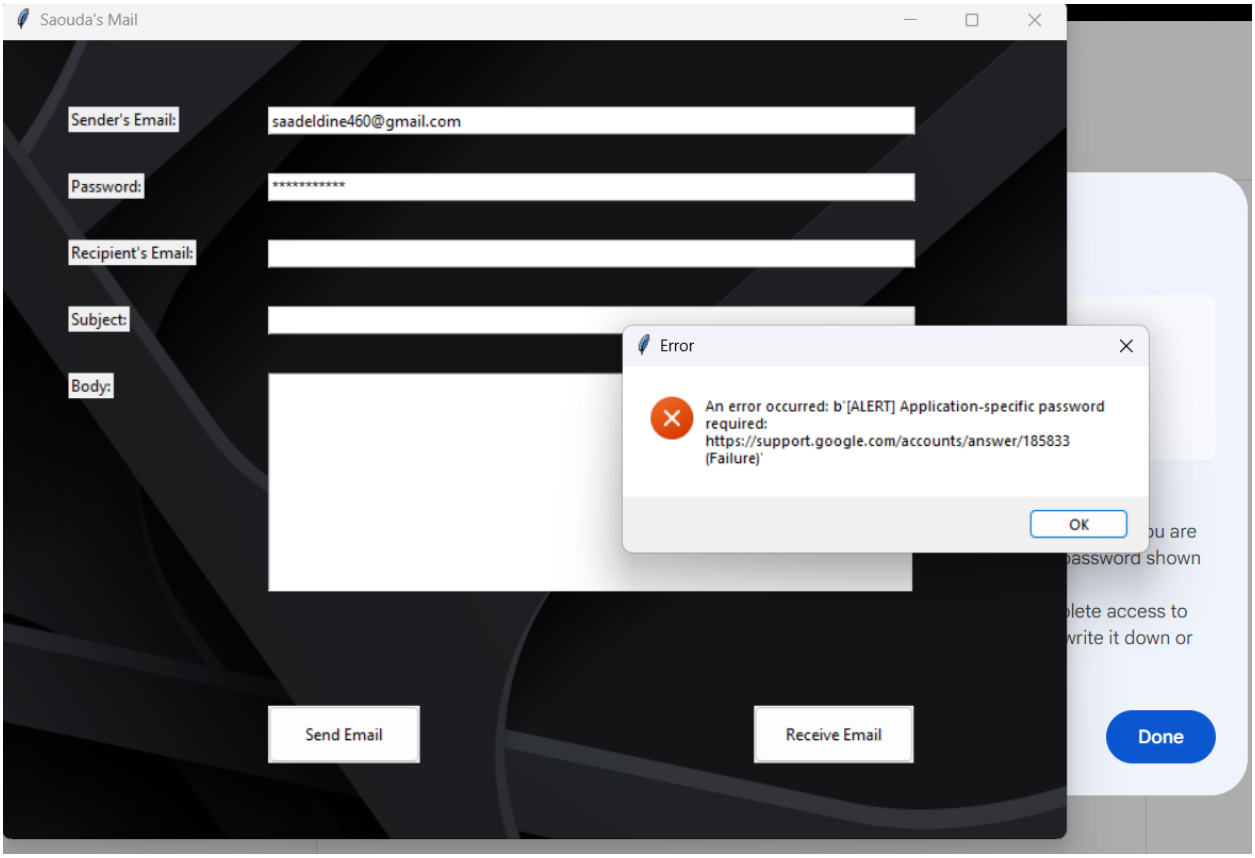
Test cases:

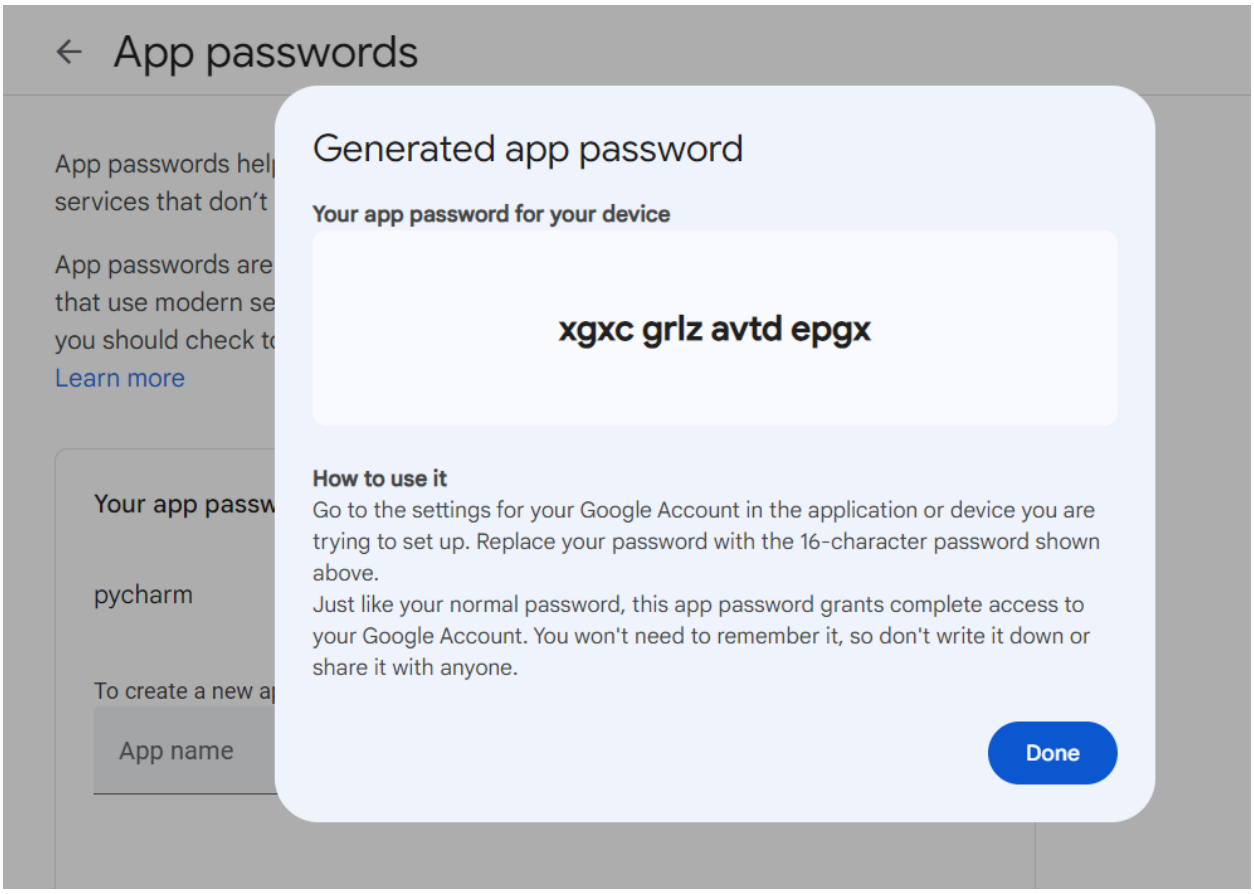
Send:

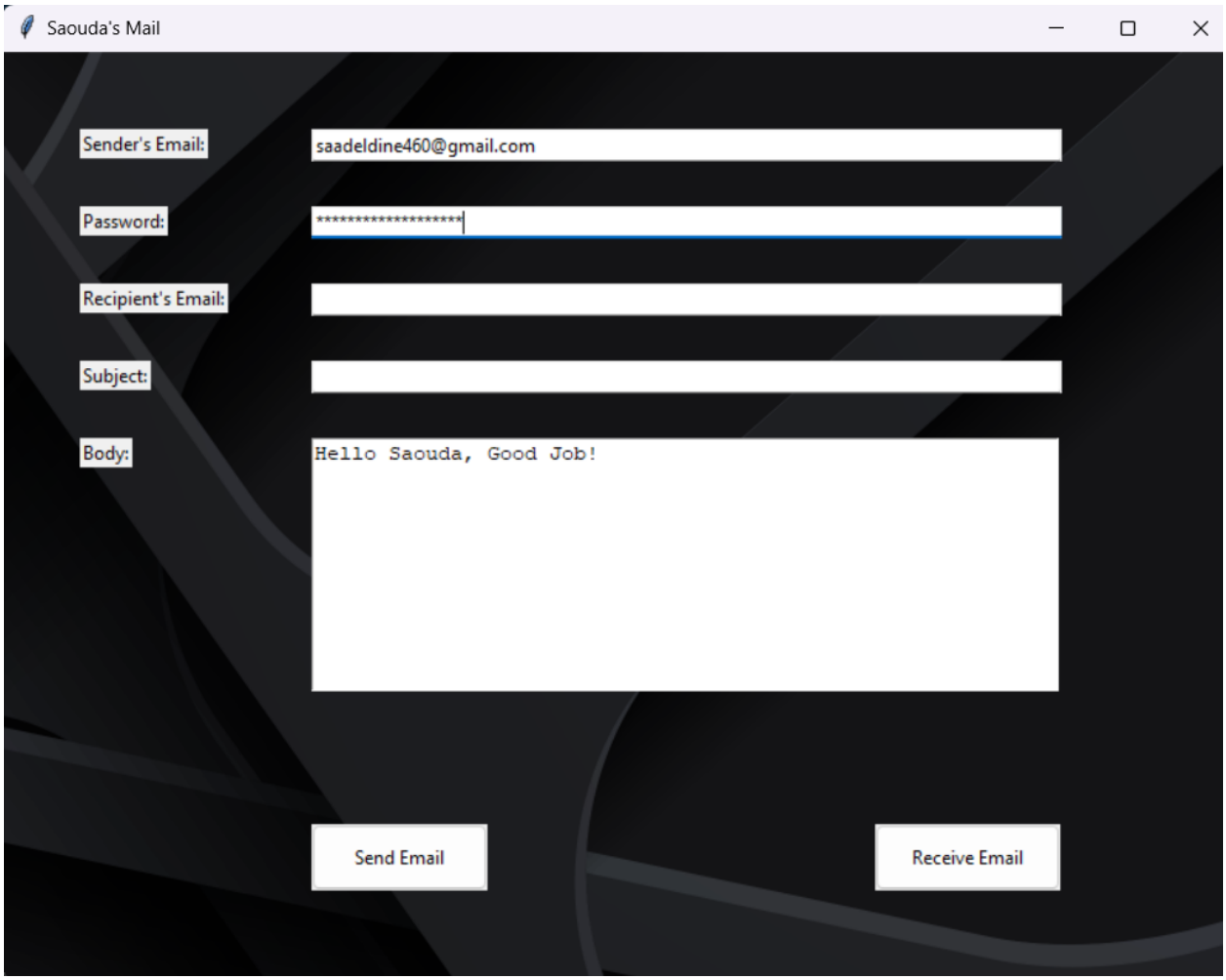




Receive:


Saouda's Mail
Sender's Email:
saadeldine460@gmail.com
Password:
****
Recipient's Email:
Subject:
Body:
Send Email
Receive Email
Error
An error occurred: b'[ALERT] Application-specific password required: https://support.google.com/accounts/answer/185833 (Failure)'
OK
Done


App passwords
Generated app password
Your app password for your device
xgxc grlz avtd epgx
How to use it
Go to the settings for your Google Account in the application or device you are trying to set up. Replace your password with the 16-character password shown above.
Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.
Done
Learn more
pycharm
App name



Handled cases: