Email Client Application

Objective

Your task 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.

Requirements

- Sending Emails: Your application should be able to send emails using the SMTP protocol. You should use the smtplib and email.mime.multipart libraries in Python for this purpose. The email should include a subject and a body. The sender's email, password, recipient's email, subject, and body should be parameters to your function.
- Receiving Emails: Your application should also be able to receive emails using the IMAP protocol. You should use the imaplib and email libraries in Python for this purpose. The application should fetch the latest email from the mailbox and print its body.
- Mail Server: For testing purposes and to maintain privacy, use mail.tm or any other alternative to protect your personal email.
- Error Handling: Your code should include appropriate error handling. If an error occurs during the sending or receiving process, your application should catch the exception and print an appropriate error message.
- Code Structure: Your code should be well-structured and properly indented. Include comments where necessary to explain your code.
- Testing: Test your application by sending emails to different user accounts and through different servers.

Bonus

- GUI: Develop a graphical user interface for your application. The GUI should allow users to input their email, password, recipient's email, subject, and body, and have buttons to send and receive emails. [use tkinter]
- Push Notifications: Instead of printing the latest email to the console, implement a push notification system that alerts the user when a new email arrives in the mailbox. [use Plyer]

Deliverables

- A Python script for sending emails.
- A Python script for receiving emails.
- A comprehensive report documenting your code, explaining how to use your application, including how to install and run your application, any dependencies your application has, and your testing process and results.

Evaluation

Your application will be evaluated on the following criteria:

- Functionality: Does your application correctly send and receive emails?
- Code Quality: Is your code well-structured, readable, and commented where necessary?
- Error Handling: Does your application handle errors gracefully, providing useful error messages?
- Documentation: Have you provided clear instructions in your report on how to use your application?

What To Submit

- Your source code, documented, and your hard-coded test cases are ready to run (do not put them within the report).
- A PDF report and any screenshots you are required to attach.
- Enclose all of that into one folder compress this folder as *.zip then change the extension of the file from *.zip to *.zip.pdf and submit it via the form.
- Name the folder as follows. studentID-sp24-nw-L3.zip.pdf (eg: 1234-sp24-nw-L3.zip.pdf).
- Submission Link: https://forms.office.com/r/7wQnPp0G9y

Policies

- No late.
- Copies will be heavily penalized.
- Your code should be clean, readable, and highly documented.
- You must submit at least one paper explaining your work testing your code and stating your conclusions (if any conclusions exist).
- Online submission on time is a mandatory step for your submission.
- You will be penalized if you don't attend the discussion on time (or at all), so stick with your timeslot.
- You should obey the submission format explained above.