Network Systems: NSS370S Dr Angus Brandt
Department of Electrical, Electronic and Computer Engineering
Cape Peninsula University of Technology

Lab 4

Socket Programming SMTP – Five-Layer Model

Name, Surname	Mnqobi Jeza
Student Number:	230878369
Date:	27 February 2025

Network Systems (NSS370S) SMTP Practical

Objective:

To understand and demonstrate how the Simple Mail Transfer Protocol (SMTP) works, send an email using Python's smtplib. Each student will personalise their email using their student number to ensure uniqueness.

Instructions:

1. Prerequisites

Ensure you have:

- Python installed on your computer
- Access to a Gmail account
- An App Password is generated for Gmail (if 2FA is enabled)
- The smtplib and email.mime.text Python libraries (pre-installed with Python)

2. Modify the Python Script

Each student must replace <student_number> with their actual student number in the following parts of the script:

```
import smtplib
from email.mime.text import MIMEText

# SMTP Server details
SMTP_SERVER = "smtp.gmail.com"
SMTP_PORT = 587
EMAIL_ADDRESS = "your-email@gmail.com"  # Replace with your own Gmail address
EMAIL_PASSWORD = "your-app-password"  # Use App Password if MFA is enabled

# Create email message
msg = MIMEText(f"This is a test email sent via Gmail SMTP. My student number is <student_number>.")
msg["Subject"] = f"SMTP Practical <student_number>"
msg["From"] = EMAIL_ADDRESS
msg["To"] = "brandta@cput.ac.za"

# Send email
server = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
server.starttls()  # Upgrade connection to secure
server.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
server.sendmail(EMAIL_ADDRESS, "brandta@cput.ac.za", msg.as_string())
server.quit()

print("Email sent successfully!")
```

Submission

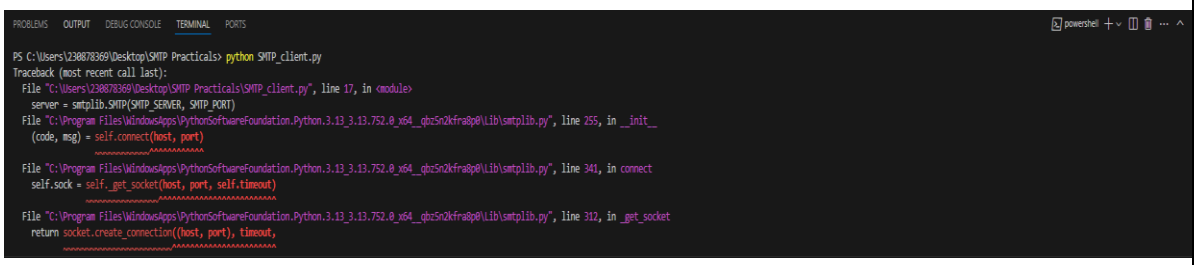
- A screenshot of your Python script**,** showing your student number in the email body and subject line.
- Screenshot the terminal showing the “Email sent successfully” response.
- Make sure your code is fully commented.

Python code:

SMTP_client.py

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

# SMTP Server details
SMTP_SERVER = "smtp.gmail.com"
SMTP_PORT = 587
EMAIL_ADDRESS = "mnqobijz41@gmail.com"  # Replace with your own Gmail address
EMAIL_PASSWORD = "gdsw wwwj aqlw wgiw"  # Use App Password if MFA is enabled

# Create email message
msg = MIMEText(f"This is a test email sent via Gmail SMTP. My student number is 230878369.")
msg["Subject"] = f"SMTP Practical 230878369"
msg["From"] = EMAIL_ADDRESS
msg["To"] = "brandta@cput.ac.za"

# Send email
server = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
server.starttls()  # Upgrade connection to secure
server.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
server.sendmail(EMAIL_ADDRESS, "brandta@cput.ac.za", msg.as_string())
server.quit()

print("Email sent successfully!")

```

Terminal output.



Authentication Setup

Gmail SMTP authentication

To authenticate and send emails using Gmail's SMTP server, follow these steps:

1. Gmail SMTP Server Details
   - SMTP Server: smtp.gmail.com
   - Port for SSL: 465
   - Port for TLS: 587
   - Authentication: Required (Username & Password)
   - Encryption: SSL or TLS
   - Username: Your Gmail address (your-email@gmail.com)
   - Password: Your Gmail password or App Password (if 2FA is enabled)

2. Generate an App Password (If 2FA is Enabled)
Gmail does not allow direct authentication with your normal password when using third-party apps or scripts. Instead, use an App Password:
Steps to Generate an App Password:
   1. Enable 2-Step Verification (Skip if already enabled):
      - Go to Google Account Security.
      - Under Signing in to Google, enable 2-Step Verification.
   2. Generate App Password:
      - Go to Security
      - Select 2-Step Verification
      - Scroll down to App passwords
      - Create a new app-specific password.
      - Click Generate and copy the 16-character password.
   3. Use this password instead of your Gmail password in SMTP authentication.

Offfice 365 SMTP authentication

2. Enable SMTP Authentication for Your Account
By default, SMTP authentication may be disabled for security reasons. You must enable it:
Enable SMTP Auth via Microsoft Admin Center
   1. Log in to Microsoft 365 Admin Center:
      - https://admin.microsoft.com
   2. Go to Users > Active Users.
   3. Select Your Account (or the user needing SMTP access).
   4. Click Mail > Manage email apps.
   5. Ensure Authenticated SMTP is enabled.
   6. Click Save.

Enable SMTP Authentication for the Organization (If Required)
   1. In the Admin Center, navigate to Settings > Org Settings.
   2. Click Modern Authentication.
   3. Scroll down to Authenticated SMTP and enable it.
   4. Click Save Changes.

END