



**Alexandria National University**  
**Faculty of Computing and Data Science**  
**Computer Networks – Final Project**

**Project Title:**

**Email Client Application**

**Submitted by:**

- Ahmed Hesham – ID: 2305277
- Youssef Hesham – ID: 2305205
- Hager Amr – ID: 2305197
- Adel Maged – ID: 2305291

**Instructor:** Dr. Mohamed Rezk

**TA:** Eng. Ahmed Ashraf / Eng. Miar Mamdouh / Eng. Salma Magdy

**Date:** 18/12/2025

# 1- Introduction

- This project implements an email notification system that integrates **SMTP, IMAP**, and **TCP socket programming**.

The system allows sending emails using SMTP and monitoring incoming emails using IMAP.

Whenever a new email is detected, a **real-time TCP notification** is sent from the server to the client.

- The goal of this project is to demonstrate practical usage of **application layer protocols** and **transport layer communication**.

## 2- System Architecture

The system consists of four main components:

- **SMTP Client:** Sends emails using Gmail SMTP server.
- **IMAP Client:** Monitors the inbox and detects new incoming emails.
- **TCP Notification Server:** Sends notifications to connected clients.
- **TCP Notification Client:** Receives real-time notifications

### Architecture Flow:

1. Email is sent using SMTP.
2. IMAP client checks the inbox.
3. If a new email is detected, the TCP server sends a notification.
4. The TCP client receives and displays the notification.

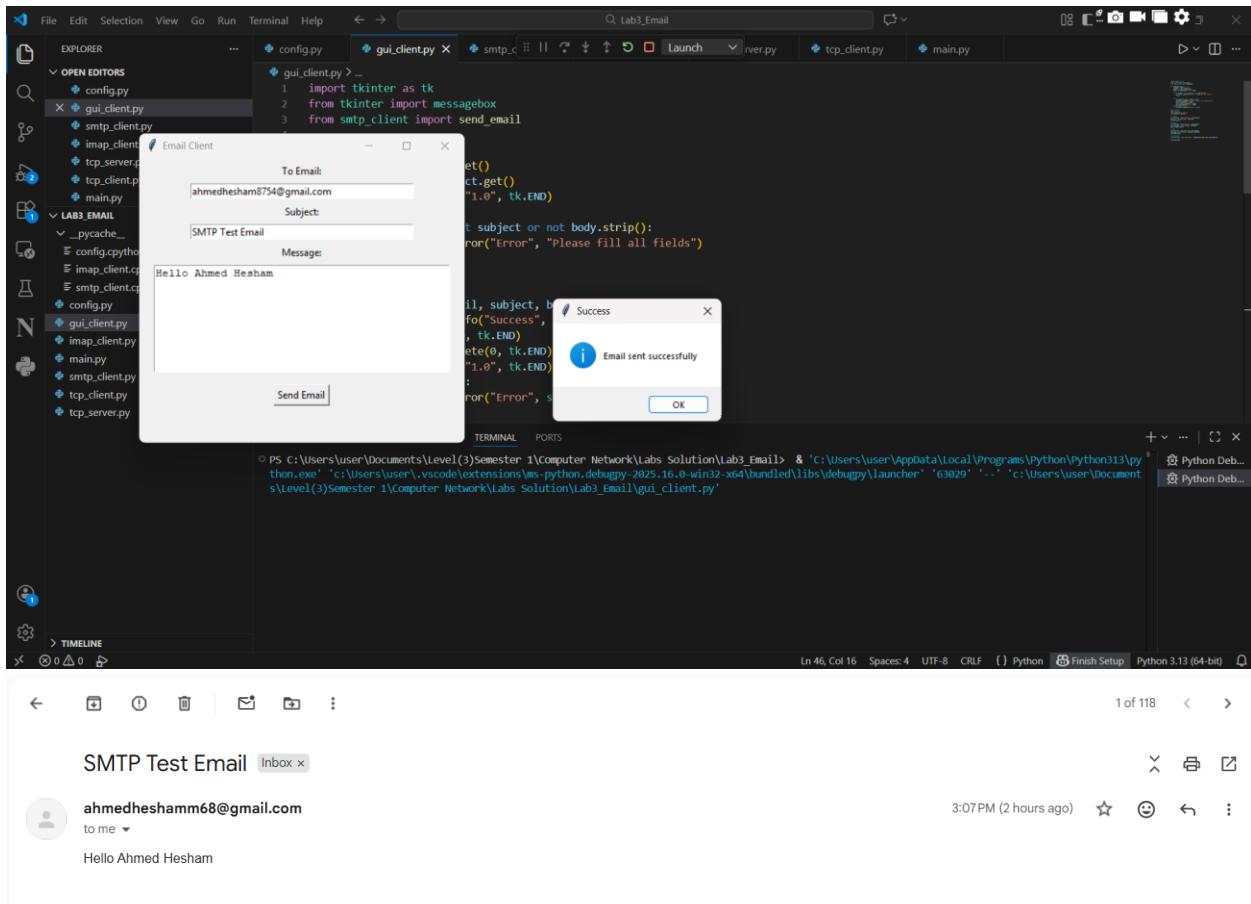
## 3- Technologies Used

- Programming Language: **Python**
- Protocols:
  - SMTP (Email Sending)
  - IMAP (Email Receiving)
  - TCP (Notification System)
- Tools:
  - Wireshark
  - Gmail (App Password)
  - Python socket library

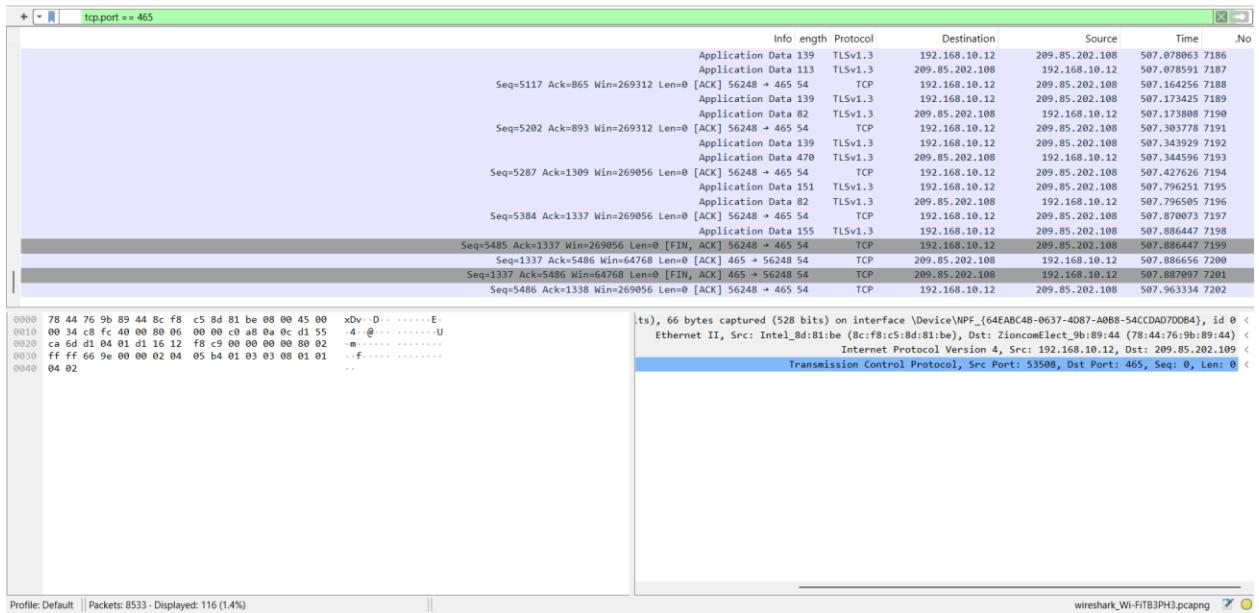
## 4- Implementation Details

### 4.1 SMTP Client

- Uses Gmail SMTP server.
- Sends email securely using SSL.
- App Password is used instead of the real Gmail password.



**smtp\_capture (Wire shark)>>(tcp.port == 465)**



## 4.2 IMAP Client

- Connects to Gmail IMAP server.
- Checks the inbox for the latest email.
- Extracts the email subject.

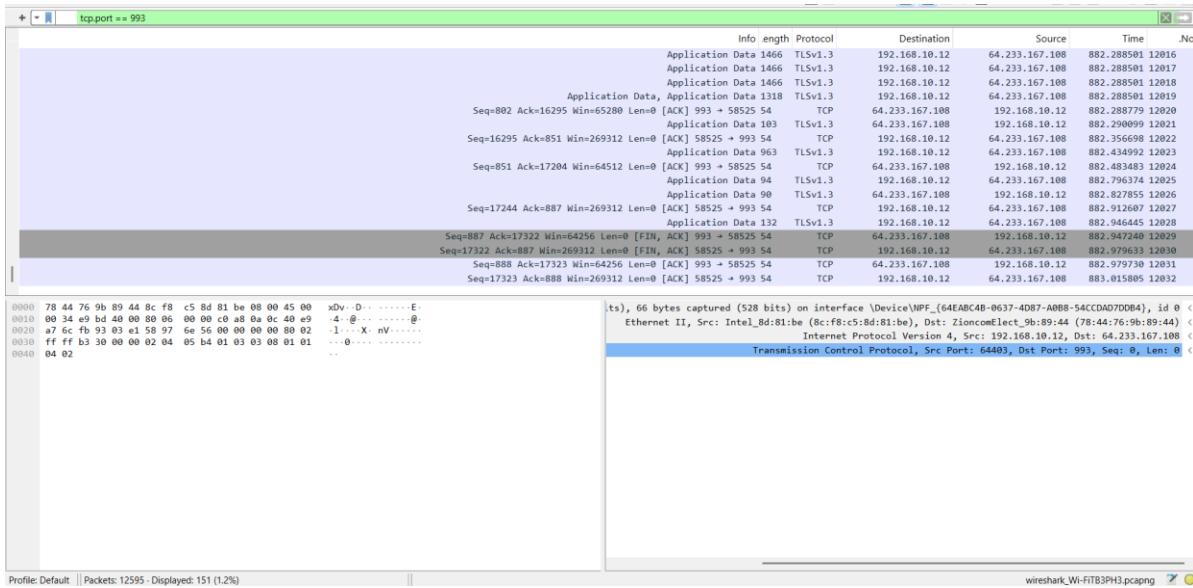
A screenshot of a code editor (VS Code) showing a Python project structure. The project includes files for SMTP, IMAP, and GUI clients, along with configuration and server scripts. The main.py file is open in the editor, containing the following code:

```

mainpy > ...
    1 from imap_client import check_latest_email
    2
    3 subject = check_latest_email()
    4 print("Latest email subject:", subject)

```

The terminal window shows command-line output related to the Python environment setup and execution of the script. The status bar at the bottom provides build and run information.



## 4.3 TCP Notification Server

- Listens on port **5050**.
- Waits for a client connection.
- Sends a notification only when a **new email** is detected.
- Prevents duplicate notifications by tracking the last received email.

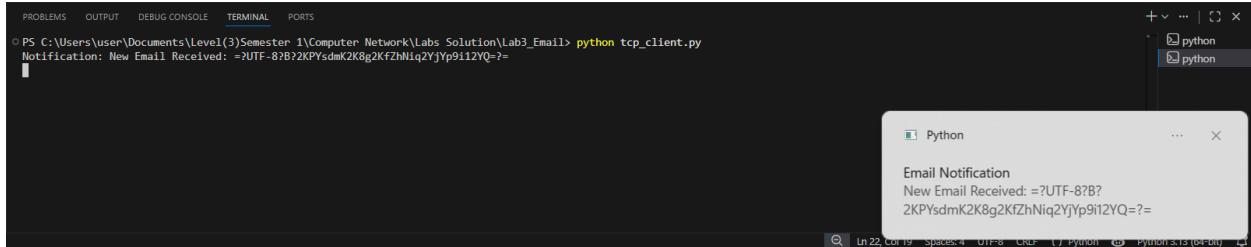
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\user\Documents\Level(3)Semester 1\Computer Network\Labs Solution\Lab3_Email> python tcp_server.py
Notification Server started...
Client connected: ('127.0.0.1', 50241)
```

## 4.4 TCP Notification Client

- Connects to the TCP server.
- Listens continuously for incoming notifications.
- Displays the received email subject.
- 

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\user\Documents\Level(3)Semester 1\Computer Network\Labs Solution\Lab3_Email> python tcp_client.py
Notification: New Email Received: SMTP Test Email
```

A push notification system was implemented using the Plyer library. When a new email arrives, the client displays a desktop notification instead of only printing the message to the console.



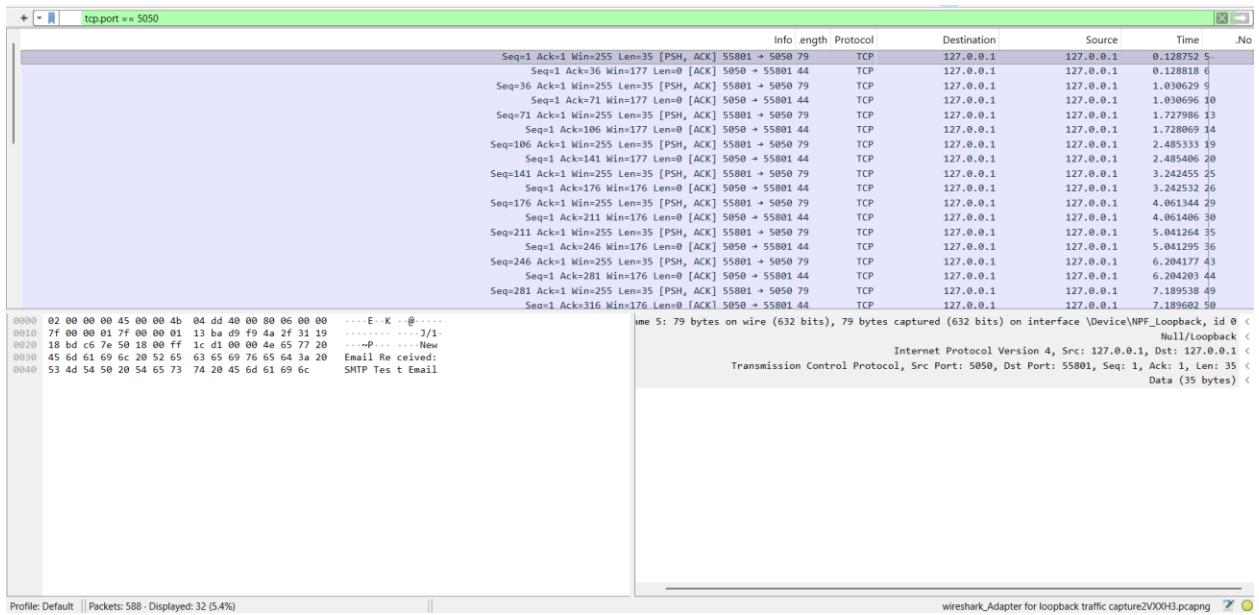
## 5- Wireshark Analysis

Wireshark was used to capture and analyze TCP traffic between the notification server and client.

- Since both client and server run on the same machine, **Loopback Adapter** was used.
- TCP packets were filtered using:

```
tcp.port == 5050
```

This confirmed successful TCP communication and data transmission.



## 6- Table (Protocol Summary)

Component	Protocol	Port	Description
Email Sending	<b>SMTP</b>	<b>465</b>	Sends emails using Gmail SMTP server
Email Receiving	<b>IMAP</b>	<b>993</b>	Checks inbox for new emails
Notification Server	<b>TCP</b>	<b>5050</b>	Sends real-time email notifications
Notification Client	<b>TCP</b>	<b>5050</b>	Receives notifications from server

## 7- Conclusion

- ❑ This project successfully demonstrates the integration of email protocols with TCP-based client-server communication.

- ❑ The system provides real-time notifications upon receiving new emails and verifies data transmission using Wireshark.
- ❑ It reflects a practical understanding of networking concepts and application layer protocols.

**Scan QR\_Code To Get Source Code:**

