project

Email Client Project

Team Members

Muhammad Ashraf Fathy Ahmed   **2305181**

Muhammad Elsyed Muhammad Ahmed   **2305273**

Yasser Ali Muhammad Muhammad   **2305268**

Muhammad Hazem Hafez Mustafa   **2305534**

Crime Rate Prediction

Project Overview:

This project implements a simple Email Client Application using Python that allows users to send and receive emails securely. The application uses SMTP for sending emails and IMAP for retrieving emails over encrypted SSL/TLS connections to ensure secure communication.

SMTP communication is performed on ports 465 or 587 using SSL or STARTTLS, while IMAP communication is handled on port 993 using SSL. The project also includes a TCP-based notification mechanism that sends a status message after each email operation.

Performance metrics such as latency, packet count, transmitted bytes, and throughput are measured for SMTP, IMAP, and notification TCP connections. Wireshark is used to capture and analyze the network traffic generated by the application.

As a bonus feature, a graphical user interface (GUI) is implemented using Tkinter, along with push notifications using the Plyer library to notify users when new emails arrive.

Part 1 — Email Client (SMTP + IMAP):

This part of the program provides a simple text-based menu that allows the user to choose between sending an email, reading the latest email, or exiting the application.
The program keeps running and displaying the menu until the user selects the exit option.

```
(dl_env) PS C:\University\Semester 5\Computer Network\Computer_Network_Projects\Email_Client_Project> python email_client.py
1. Send Email (SMTP)
2. Read Latest Email (IMAP)
3. Exit
Choose an option [1-3]: 1
=== Send Email (SMTP) ===
SMTP server: smtp.ethereal.email
SMTP port (465 for SSL, 587 for TLS) [587]: 587
SMTP username: antonia.hagenes@ethereal.email
SMTP password:
Receiver email: antonia.hagenes@ethereal.email
Subject: Wireshark
Body: Wireshark 221B
Email sent successfully!
SMTP Latency: 4.114 seconds

Simple Email Client
1. Send Email (SMTP)
2. Read Latest Email (IMAP)
3. Exit
Choose an option [1-3]: 2
=== Read Latest Email (IMAP) ===
IMAP server: smtp.ethereal.email
IMAP port [993]: 993
IMAP username: antonia.hagenes@ethereal.email
IMAP password:
Subject: Wireshark
Body:
 Wireshark 221B

IMAP Latency: 3.277 seconds

Simple Email Client
1. Send Email (SMTP)
2. Read Latest Email (IMAP)
3. Exit
Choose an option [1-3]: 3
Goodbye.
(dl_env) PS C:\University\Semester 5\Computer Network\Computer_Network_Projects\Email_Client_Project>
```

Note: The GUI was added in the final stage

Part 2 — TCP Notification Server:

The notification server listens for TCP messages from the email client and displays them as user notifications.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

(dl_env) PS C:\University\Semester 5\Computer Network\Computer_Network_Projects\Email_Client_Project> python notification_server.py
[Notification Server] Listening on 127.0.0.1:9999
[+] Connection from ('127.0.0.1', 53260)
[NOTIFICATION] Email Sent Successfully
[+] Connection from ('127.0.0.1', 53467)
[NOTIFICATION] Email Received Successfully
```

Part 3 — Wireshark Packet Capture & Analysis:

SMTP traffic: tcp.port == 465 or tcp.port == 587.

No.	Time	Source	Destination	Protocol	Length	Info
3085	156.442178	192.168.1.13	95.216.108.161	TCP	66	51972 → 587 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
3086	156.542173	95.216.108.161	192.168.1.13	TCP	66	587 → 51972 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1400 SACK_PERM WS=128
3087	156.542279	192.168.1.13	95.216.108.161	TCP	54	51972 → 587 [ACK] Seq=1 Ack=1 Win=65280 Len=0
3088	156.740847	95.216.108.161	192.168.1.13	SMTP	109	S: 220 smtp.ethereal.email ESMTP Welcome to Ethereal MSA
3089	156.788992	192.168.1.13	95.216.108.161	TCP	54	51972 → 587 [ACK] Seq=1 Ack=56 Win=65280 Len=0
3090	157.015552	192.168.1.13	95.216.108.161	SMTP	75	C: ehlo [192.168.1.13]
3091	157.114433	95.216.108.161	192.168.1.13	SMTP	195	S: 250-smtp.ethereal.email Nice to meet you, [154.237.216.143]	PIPELINING	8BITMIME	SMTPUTF8	AUTH LOGIN PLAIN	STARTTLS
3092	157.114640	95.216.108.161	192.168.1.13	TCP	54	587 → 51972 [ACK] Seq=56 Ack=22 Win=64256 Len=0
3093	157.114663	192.168.1.13	95.216.108.161	TCP	54	51972 → 587 [ACK] Seq=22 Ack=197 Win=65280 Len=0
3094	157.114887	192.168.1.13	95.216.108.161	SMTP	64	C: STARTTLS
3095	157.212549	95.216.108.161	192.168.1.13	SMTP	78	S: 220 Ready to start TLS
3096	157.213127	192.168.1.13	95.216.108.161	TLSv1.3	571	Client Hello (SNI=smtp.ethereal.email)
3097	157.315852	95.216.108.161	192.168.1.13	TLSv1.3	1454	Server Hello, Change Cipher Spec, Application Data
3098	157.315852	95.216.108.161	192.168.1.13	TLSv1.3	1073	Application Data, Application Data, Application Data
3099	157.315955	192.168.1.13	95.216.108.161	TCP	54	51972 → 587 [ACK] Seq=549 Ack=2640 Win=65280 Len=0
3100	157.321412	192.168.1.13	95.216.108.161	TLSv1.3	134	Change Cipher Spec, Application Data
3101	157.419146	95.216.108.161	192.168.1.13	TLSv1.3	628	Application Data, Application Data
3102	157.419224	192.168.1.13	95.216.108.161	TLSv1.3	97	Application Data
3103	157.516483	95.216.108.161	192.168.1.13	TLSv1.3	203	Application Data
3104	157.516793	192.168.1.13	95.216.108.161	TLSv1.3	157	Application Data
3105	157.625490	95.216.108.161	192.168.1.13	TLSv1.3	107	Application Data
3106	157.626330	192.168.1.13	95.216.108.161	TLSv1.3	120	Application Data
3107	157.723694	95.216.108.161	192.168.1.13	TLSv1.3	90	Application Data
3108	157.724044	192.168.1.13	95.216.108.161	TLSv1.3	118	Application Data
3110	157.821809	95.216.108.161	192.168.1.13	TLSv1.3	90	Application Data
3111	157.822091	192.168.1.13	95.216.108.161	TLSv1.3	82	Application Data
3113	157.918700	95.216.108.161	192.168.1.13	TLSv1.3	113	Application Data
3114	157.919050	192.168.1.13	95.216.108.161	TLSv1.3	301	Application Data
3116	158.025870	95.216.108.161	192.168.1.13	TLSv1.3	169	Application Data
3117	158.026163	192.168.1.13	95.216.108.161	TLSv1.3	82	Application Data
3119	158.122999	192.168.1.13	95.216.108.161	TLSv1.3	85	Application Data
3120	158.123293	95.216.108.161	192.168.1.13	TLSv1.3	78	Application Data
3121	158.123296	192.168.1.13	95.216.108.161	TCP	54	51972 → 587 [FIN, ACK] Seq=1208 Ack=3693 Win=64256 Len=0
3122	158.123326	192.168.1.13	95.216.108.161	TCP	54	51972 → 587 [RST, ACK] Seq=1209 Ack=3717 Win=0 Len=0
3123	158.221142	95.216.108.161	192.168.1.13	TCP	54	587 → 51972 [ACK] Seq=3718 Ack=1209 Win=64128 Len=0

IMAP traffic: tcp.port == 993.

No.	Time	Source	Destination	Protocol	Length	Info
4404	314.856912	192.168.1.13	95.216.108.161	TCP	66	53390 → 993 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
4405	314.958436	95.216.108.161	192.168.1.13	TCP	66	993 → 53390 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1400 SACK_PERM WS=128
4406	314.958599	192.168.1.13	95.216.108.161	TCP	54	53390 → 993 [ACK] Seq=1 Ack=1 Win=65280 Len=0
4407	314.959259	192.168.1.13	95.216.108.161	TLSv1.3	571	Client Hello (SNI=smtp.ethereal.email)
4408	315.062668	95.216.108.161	192.168.1.13	TCP	54	993 → 53390 [ACK] Seq=1 Ack=518 Win=64128 Len=0
4409	315.067366	95.216.108.161	192.168.1.13	TLSv1.3	1454	Server Hello, Change Cipher Spec, Application Data
4410	315.067366	95.216.108.161	192.168.1.13	TLSv1.3	1074	Application Data, Application Data, Application Data
4411	315.067435	192.168.1.13	95.216.108.161	TCP	54	53390 → 993 [ACK] Seq=518 Ack=2421 Win=65280 Len=0
4412	315.068561	192.168.1.13	95.216.108.161	TLSv1.3	134	Change Cipher Spec, Application Data
4413	315.165832	95.216.108.161	192.168.1.13	TLSv1.3	628	Application Data, Application Data
4414	315.218137	192.168.1.13	95.216.108.161	TCP	54	53390 → 993 [ACK] Seq=598 Ack=2995 Win=64768 Len=0
4415	315.313742	95.216.108.161	192.168.1.13	TLSv1.3	166	Application Data
4416	315.314157	192.168.1.13	95.216.108.161	TLSv1.3	94	Application Data
4417	315.409914	95.216.108.161	192.168.1.13	TLSv1.3	197	Application Data
4418	315.458364	192.168.1.13	95.216.108.161	TCP	54	53390 → 993 [ACK] Seq=638 Ack=3250 Win=64512 Len=0
4419	315.554499	95.216.108.161	192.168.1.13	TLSv1.3	107	Application Data
4420	315.554935	192.168.1.13	95.216.108.161	TLSv1.3	141	Application Data
4421	315.686835	192.168.1.13	95.216.108.161	TLSv1.3	223	Application Data
4422	315.731272	192.168.1.13	95.216.108.161	TCP	54	53390 → 993 [ACK] Seq=725 Ack=3472 Win=64256 Len=0
4423	315.826914	95.216.108.161	192.168.1.13	TLSv1.3	131	Application Data
4424	315.827196	192.168.1.13	95.216.108.161	TLSv1.3	96	Application Data
4426	315.923834	95.216.108.161	192.168.1.13	TLSv1.3	128	Application Data
4429	315.970501	192.168.1.13	95.216.108.161	TCP	54	53390 → 993 [ACK] Seq=767 Ack=3623 Win=64256 Len=0
4430	316.079269	95.216.108.161	192.168.1.13	TLSv1.3	336	Application Data
4431	316.079652	192.168.1.13	95.216.108.161	TLSv1.3	94	Application Data
4435	316.176518	95.216.108.161	192.168.1.13	TLSv1.3	90	Application Data
4436	316.226176	192.168.1.13	95.216.108.161	TCP	54	53390 → 993 [ACK] Seq=807 Ack=3941 Win=65280 Len=0
4439	316.322524	95.216.108.161	192.168.1.13	TLSv1.3	103	Application Data
4441	316.323036	192.168.1.13	95.216.108.161	TLSv1.3	100	Application Data
4443	316.422579	192.168.1.13	95.216.108.161	TLSv1.3	101	Application Data
4444	316.466561	192.168.1.13	95.216.108.161	TCP	54	53390 → 993 [ACK] Seq=853 Ack=4037 Win=65280 Len=0
4453	316.562560	95.216.108.161	192.168.1.13	TLSv1.3	512	Application Data, Application Data, Application Data, Application Data, Application Data, Application Data
4454	316.609493	192.168.1.13	95.216.108.161	TCP	54	53390 → 993 [ACK] Seq=853 Ack=4495 Win=64768 Len=0

Notification server traffic: tcp.port == 9999

No.	Time	Source	Destination	Protocol	Length	Info
3186	216.752632	127.0.0.1	127.0.0.1	TCP	56	53260 → 9999 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
3187	216.752706	127.0.0.1	127.0.0.1	TCP	56	9999 → 53260 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
3188	216.752740	127.0.0.1	127.0.0.1	TCP	44	53260 → 9999 [ACK] Seq=1 Ack=1 Win=65280 Len=0
3189	216.752816	127.0.0.1	127.0.0.1	TCP	67	53260 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=23
3190	216.752838	127.0.0.1	127.0.0.1	TCP	44	9999 → 53260 [ACK] Seq=1 Ack=24 Win=65280 Len=0
3191	216.752873	127.0.0.1	127.0.0.1	TCP	44	53260 → 9999 [FIN, ACK] Seq=24 Ack=1 Win=65280 Len=0
3192	216.752891	127.0.0.1	127.0.0.1	TCP	44	9999 → 53260 [ACK] Seq=1 Ack=25 Win=65280 Len=0
3193	216.753258	127.0.0.1	127.0.0.1	TCP	44	9999 → 53260 [FIN, ACK] Seq=1 Ack=25 Win=65280 Len=0
3194	216.753292	127.0.0.1	127.0.0.1	TCP	44	53260 → 9999 [ACK] Seq=25 Ack=2 Win=65280 Len=0
4914	342.382601	127.0.0.1	127.0.0.1	TCP	56	53467 → 9999 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
4916	342.382680	127.0.0.1	127.0.0.1	TCP	56	9999 → 53467 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
4918	342.382731	127.0.0.1	127.0.0.1	TCP	44	53467 → 9999 [ACK] Seq=1 Ack=1 Win=65280 Len=0
4919	342.382811	127.0.0.1	127.0.0.1	TCP	71	53467 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=27
4921	342.382842	127.0.0.1	127.0.0.1	TCP	44	9999 → 53467 [ACK] Seq=1 Ack=28 Win=65280 Len=0
4924	342.382877	127.0.0.1	127.0.0.1	TCP	44	53467 → 9999 [FIN, ACK] Seq=28 Ack=1 Win=65280 Len=0
4926	342.382895	127.0.0.1	127.0.0.1	TCP	44	9999 → 53467 [ACK] Seq=1 Ack=29 Win=65280 Len=0
4927	342.383609	127.0.0.1	127.0.0.1	TCP	44	9999 → 53467 [FIN, ACK] Seq=1 Ack=29 Win=65280 Len=0
4928	342.383648	127.0.0.1	127.0.0.1	TCP	44	53467 → 9999 [ACK] Seq=29 Ack=2 Win=65280 Len=0

Part 4 — Performance Measurements:

**Number of packets = last packet num – first packet num +1**
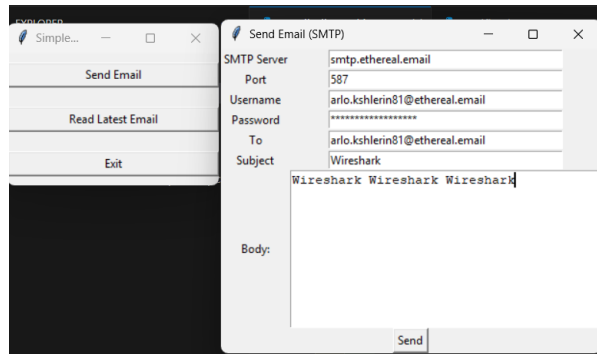
Latency = End Time – Start Time

Throughput = total_bytes / Latency

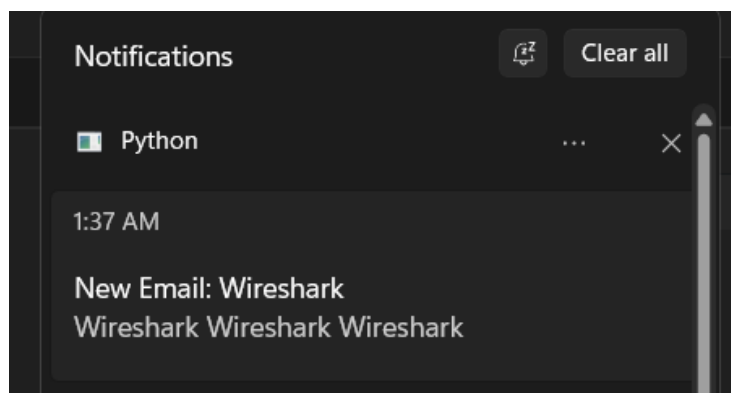| Operation	Time	Packets	Bytes	Throughput
SMTP Send	1.778	39	6837	3843.39
IMAP Fetch	1.752	51	6987	3988.01
Notification TCP	$6.6*10^{-4}$	8	443	671212.12

The application has a Tkinter GUI for sending and receiving emails and uses Plyer to notify users of new emails.

GUI (Tkinter):
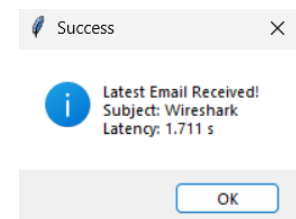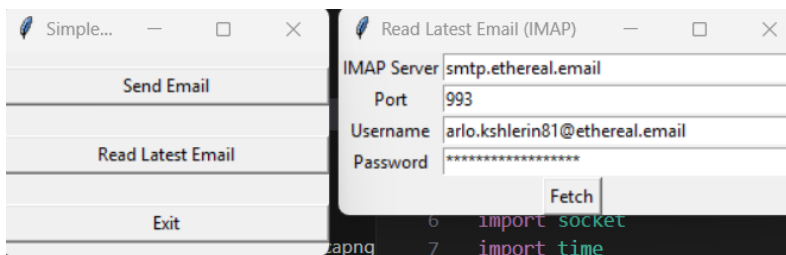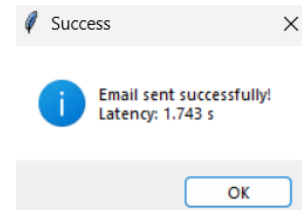
Push Notifications (Plyer):

🔸 **Source code:**
https://github.com/MuhammadAshraf221B/Computer_Network_Projects/tree/main/Email_Client_Project