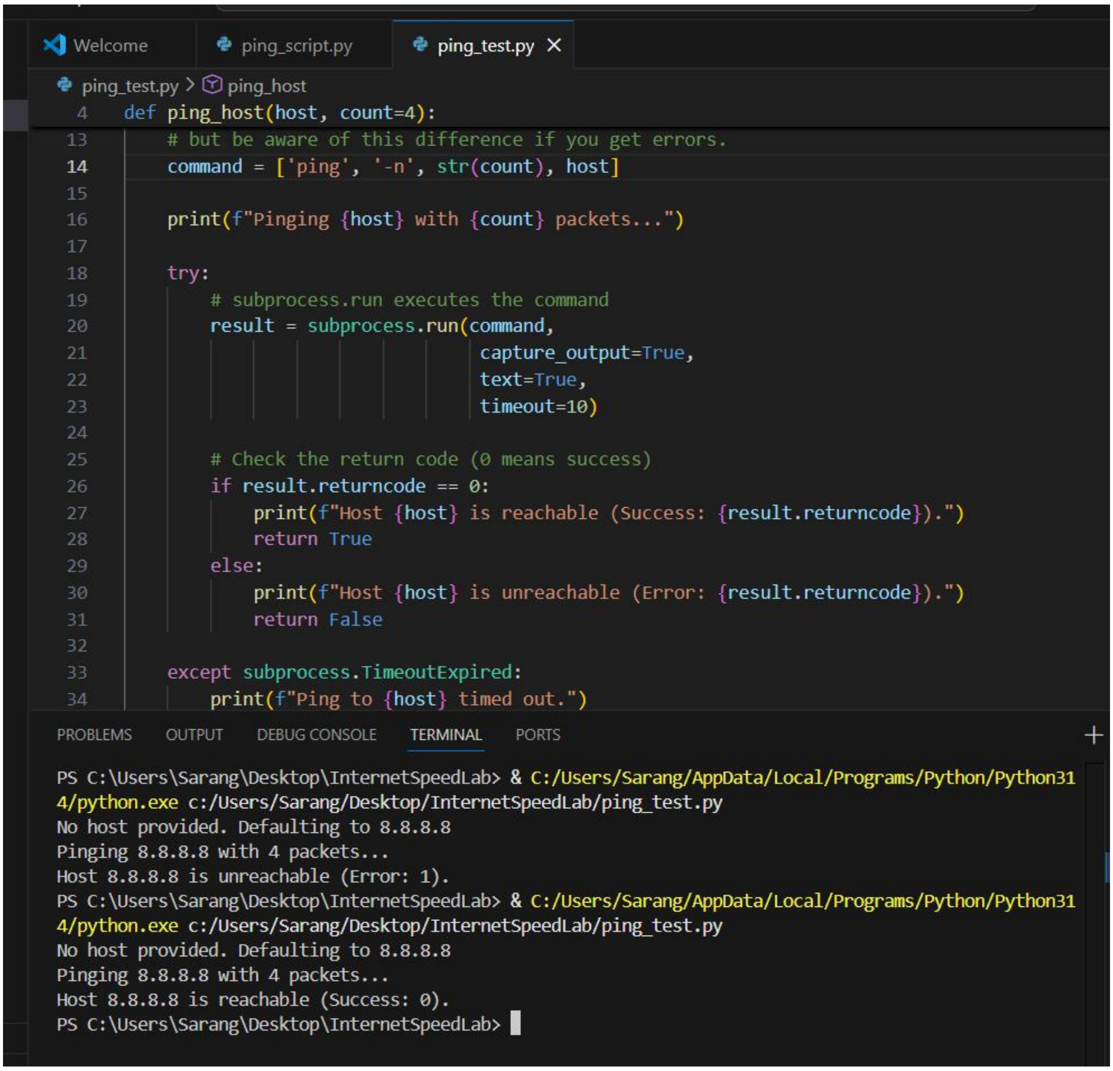
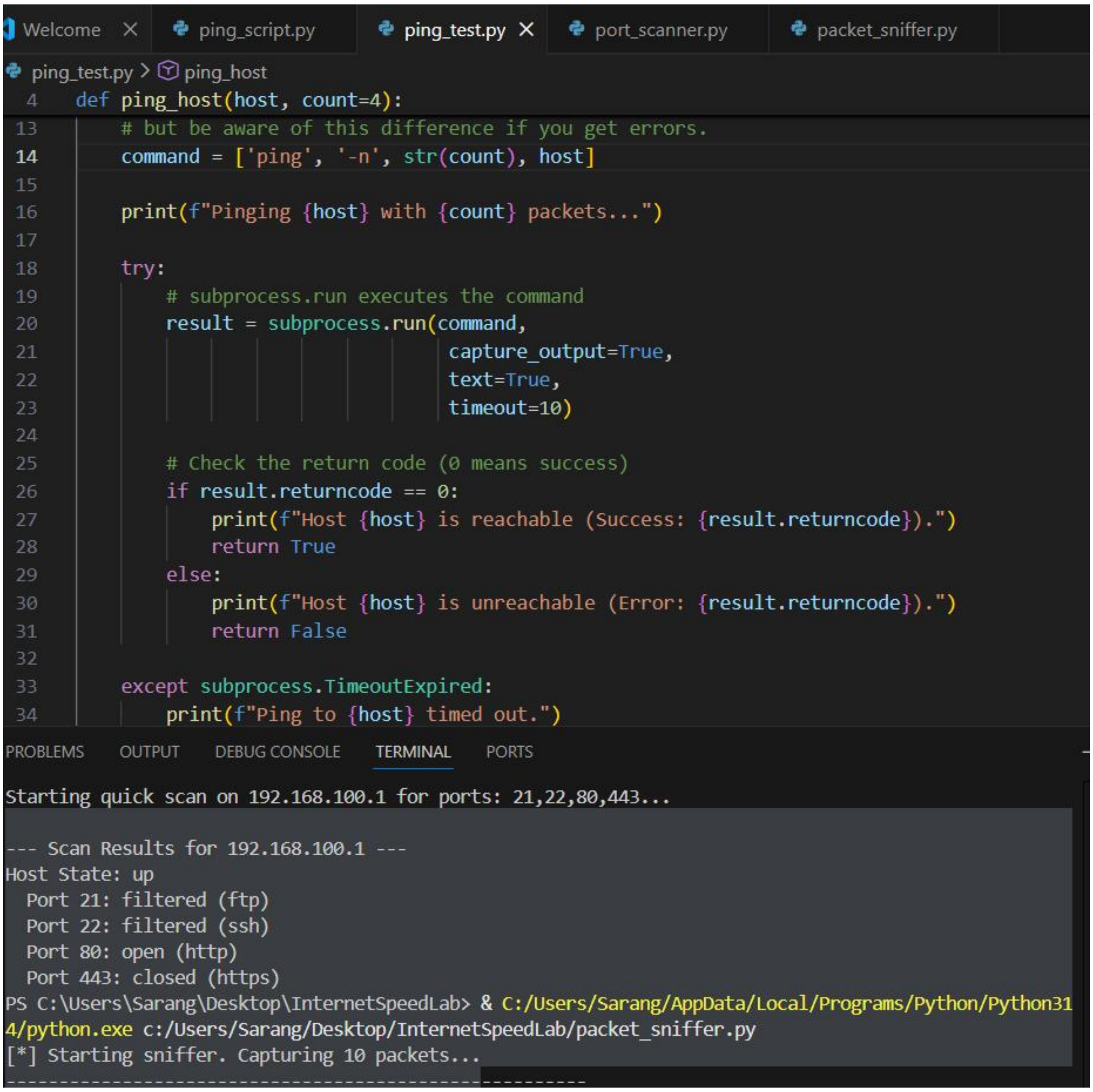
TASK 1.1 Application of Python in Networking



```python
def ping_host(host, count=4):
    # but be aware of this difference if you get errors.
    command = ['ping', '-n', str(count), host]

    print(f"Pinging {host} with {count} packets...")

    try:
        # subprocess.run executes the command
        result = subprocess.run(command,
                                capture_output=True,
                                text=True,
                                timeout=10)

        # Check the return code (0 means success)
        if result.returncode == 0:
            print(f"Host {host} is reachable (Success: {result.returncode}).")
            return True
        else:
            print(f"Host {host} is unreachable (Error: {result.returncode}).")
            return False

    except subprocess.TimeoutExpired:
        print(f"Ping to {host} timed out.")
```

```
PS C:\Users\Sarang\Desktop\InternetSpeedLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python31
4/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/ping_test.py
No host provided. Defaulting to 8.8.8.8
Pinging 8.8.8.8 with 4 packets...
Host 8.8.8.8 is unreachable (Error: 1).
PS C:\Users\Sarang\Desktop\InternetSpeedLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python31
4/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/ping_test.py
No host provided. Defaulting to 8.8.8.8
Pinging 8.8.8.8 with 4 packets...
Host 8.8.8.8 is reachable (Success: 0).
PS C:\Users\Sarang\Desktop\InternetSpeedLab>
```

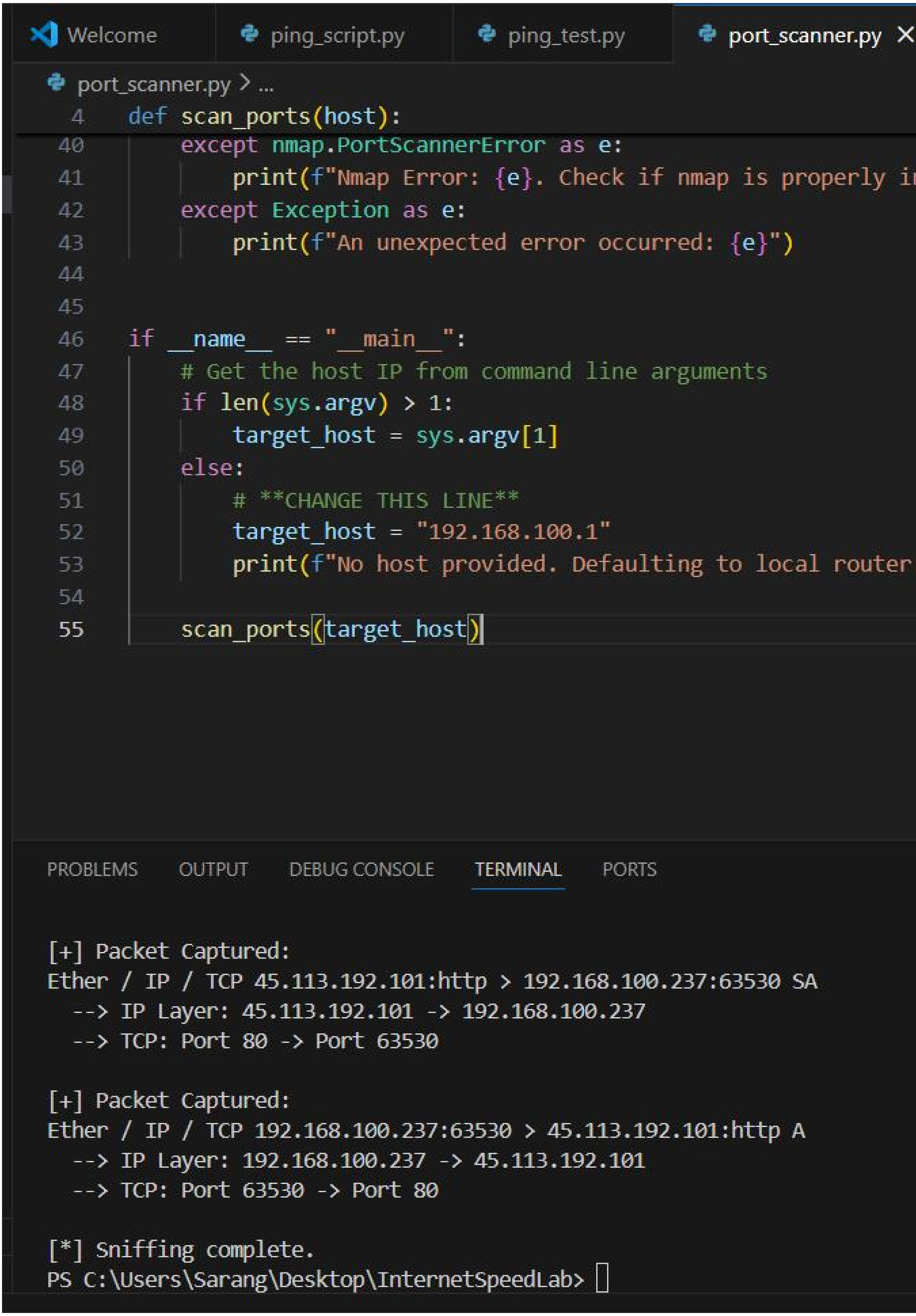
TASK 1.2



```python
def ping_host(host, count=4):
    # but be aware of this difference if you get errors.
    command = ['ping', '-n', str(count), host]

    print(f"Pinging {host} with {count} packets...")

    try:
        # subprocess.run executes the command
        result = subprocess.run(command,
                                capture_output=True,
                                text=True,
                                timeout=10)

        # Check the return code (0 means success)
        if result.returncode == 0:
            print(f"Host {host} is reachable (Success: {result.returncode}).")
            return True
        else:
            print(f"Host {host} is unreachable (Error: {result.returncode}).")
            return False

    except subprocess.TimeoutExpired:
        print(f"Ping to {host} timed out.")
```

```
Starting quick scan on 192.168.100.1 for ports: 21,22,80,443...

--- Scan Results for 192.168.100.1 ---
Host State: up
  Port 21: filtered (ftp)
  Port 22: filtered (ssh)
  Port 80: open (http)
  Port 443: closed (https)
PS C:\Users\Sarang\Desktop\InternetSpeedLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python31
4/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/packet_sniffer.py
[*] Starting sniffer. Capturing 10 packets...
-------------------------------------------------------
```

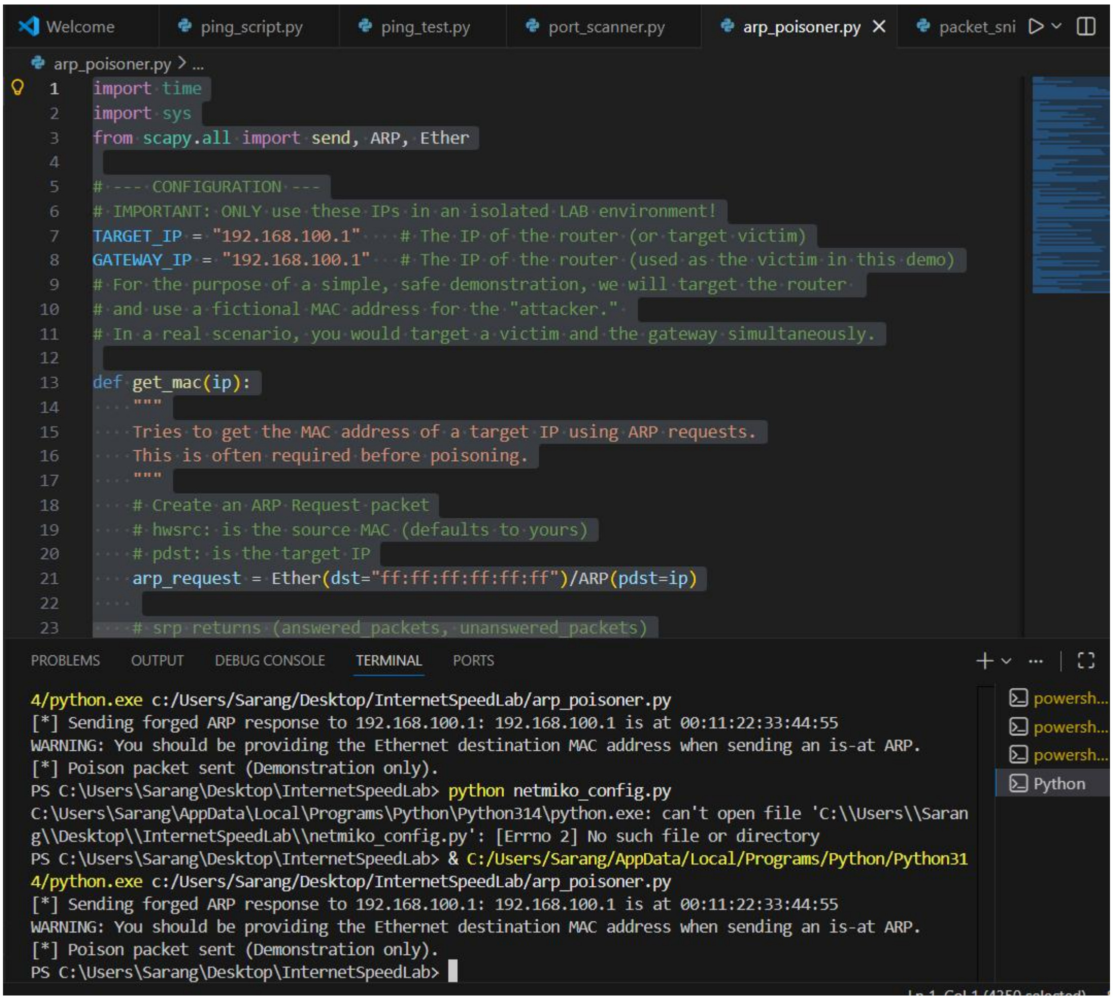
TASK 2.1



```
def scan_ports(host):
        except nmap.PortScannerError as e:
            print(f"Nmap Error: {e}. Check if nmap is properly i
        except Exception as e:
            print(f"An unexpected error occurred: {e}")


if __name__ == "__main__":
    # Get the host IP from command line arguments
    if len(sys.argv) > 1:
        target_host = sys.argv[1]
    else:
        # **CHANGE THIS LINE**
        target_host = "192.168.100.1"
        print(f"No host provided. Defaulting to local router

    scan_ports(target_host)
```

```
[+] Packet Captured:
Ether / IP / TCP 45.113.192.101:http > 192.168.100.237:63530 SA
  --> IP Layer: 45.113.192.101 -> 192.168.100.237
  --> TCP: Port 80 -> Port 63530

[+] Packet Captured:
Ether / IP / TCP 192.168.100.237:63530 > 45.113.192.101:http A
  --> IP Layer: 192.168.100.237 -> 45.113.192.101
  --> TCP: Port 63530 -> Port 80

[*] Sniffing complete.
PS C:\Users\Sarang\Desktop\InternetSpeedLab>
```

TASK 2.2

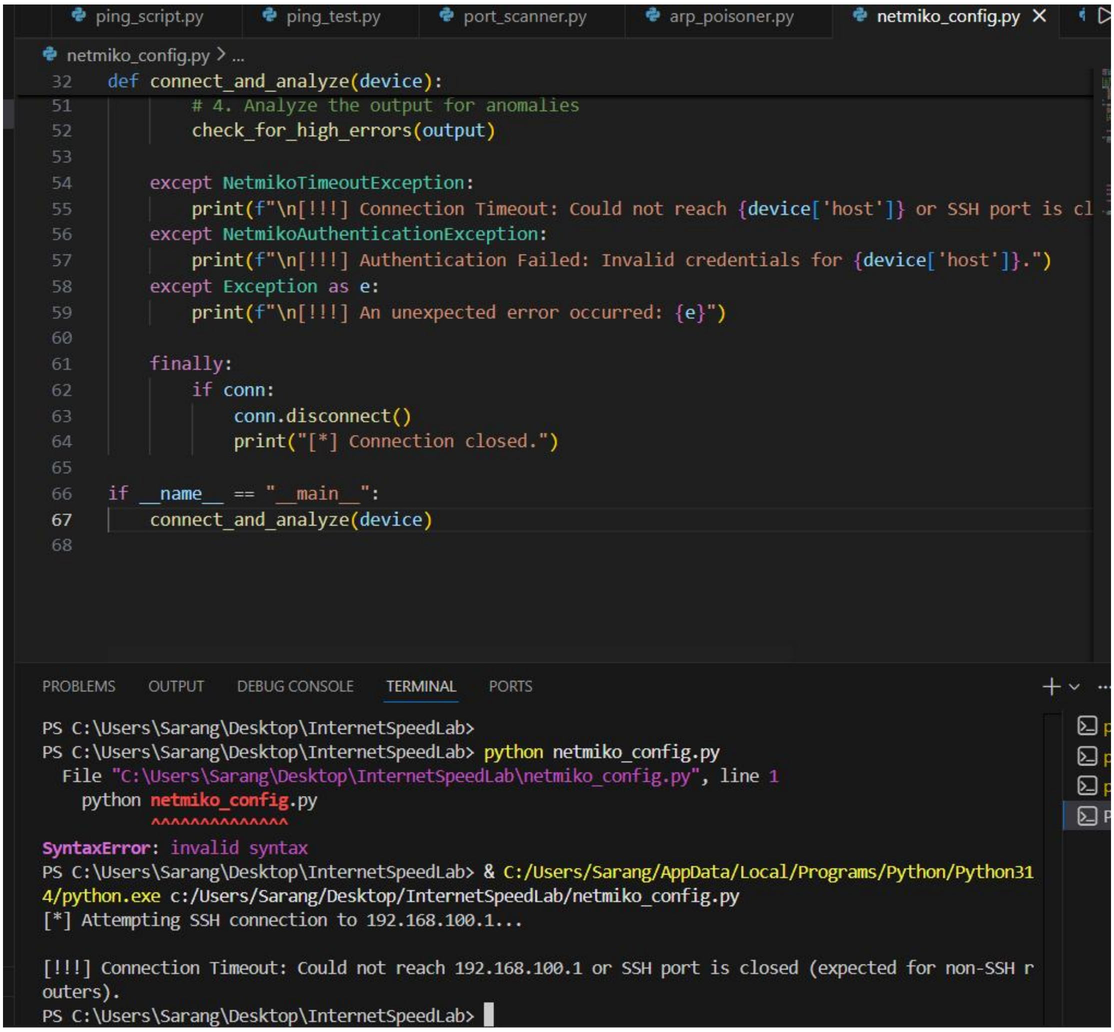


```python
import time
import sys
from scapy.all import send, ARP, Ether

# --- CONFIGURATION ---
# IMPORTANT: ONLY use these IPs in an isolated LAB environment!
TARGET_IP = "192.168.100.1"    # The IP of the router (or target victim)
GATEWAY_IP = "192.168.100.1"   # The IP of the router (used as the victim in this demo)
# For the purpose of a simple, safe demonstration, we will target the router
# and use a fictional MAC address for the "attacker."
# In a real scenario, you would target a victim and the gateway simultaneously.

def get_mac(ip):
    """
    Tries to get the MAC address of a target IP using ARP requests.
    This is often required before poisoning.
    """
    # Create an ARP Request packet
    # hwsrc: is the source MAC (defaults to yours)
    # pdst: is the target IP
    arp_request = Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=ip)
    
    # srp returns (answered packets, unanswered packets)
```

```
4/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/arp_poisoner.py
[*] Sending forged ARP response to 192.168.100.1: 192.168.100.1 is at 00:11:22:33:44:55
WARNING: You should be providing the Ethernet destination MAC address when sending an is-at ARP.
[*] Poison packet sent (Demonstration only).
PS C:\Users\Sarang\Desktop\InternetSpeedLab> python netmiko_config.py
C:\Users\Sarang\AppData\Local\Programs\Python\Python314\python.exe: can't open file 'C:\\Users\\Saran
g\\Desktop\\InternetSpeedLab\\netmiko_config.py': [Errno 2] No such file or directory
PS C:\Users\Sarang\Desktop\InternetSpeedLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python31
4/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/arp_poisoner.py
[*] Sending forged ARP response to 192.168.100.1: 192.168.100.1 is at 00:11:22:33:44:55
WARNING: You should be providing the Ethernet destination MAC address when sending an is-at ARP.
[*] Poison packet sent (Demonstration only).
PS C:\Users\Sarang\Desktop\InternetSpeedLab> 
```

Ln 1, Col 1 (4250 selected)

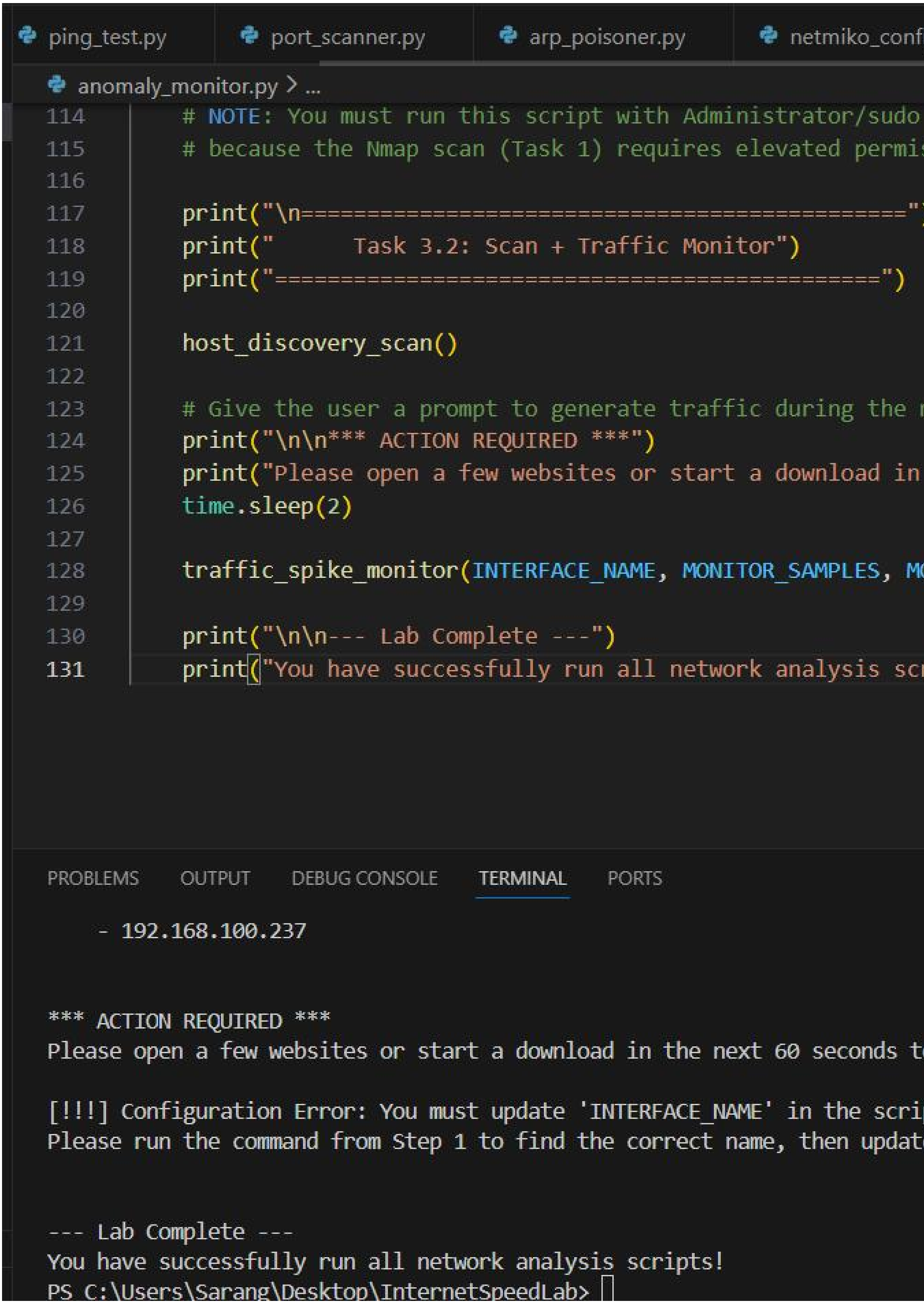
TASK 3.1



```python
def connect_and_analyze(device):
        # 4. Analyze the output for anomalies
        check_for_high_errors(output)

    except NetmikoTimeoutException:
        print(f"\n[!!!] Connection Timeout: Could not reach {device['host']} or SSH port is cl
    except NetmikoAuthenticationException:
        print(f"\n[!!!] Authentication Failed: Invalid credentials for {device['host']}.")
    except Exception as e:
        print(f"\n[!!!] An unexpected error occurred: {e}")

    finally:
        if conn:
            conn.disconnect()
            print("[*] Connection closed.")

if __name__ == "__main__":
    connect_and_analyze(device)

```

```
PS C:\Users\Sarang\Desktop\InternetSpeedLab>
PS C:\Users\Sarang\Desktop\InternetSpeedLab> python netmiko_config.py
  File "C:\Users\Sarang\Desktop\InternetSpeedLab\netmiko_config.py", line 1
    python netmiko_config.py
           ^^^^^^^^^^^^^^
SyntaxError: invalid syntax
PS C:\Users\Sarang\Desktop\InternetSpeedLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python31
4/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/netmiko_config.py
[*] Attempting SSH connection to 192.168.100.1...

[!!!] Connection Timeout: Could not reach 192.168.100.1 or SSH port is closed (expected for non-SSH r
outers).
PS C:\Users\Sarang\Desktop\InternetSpeedLab>
```

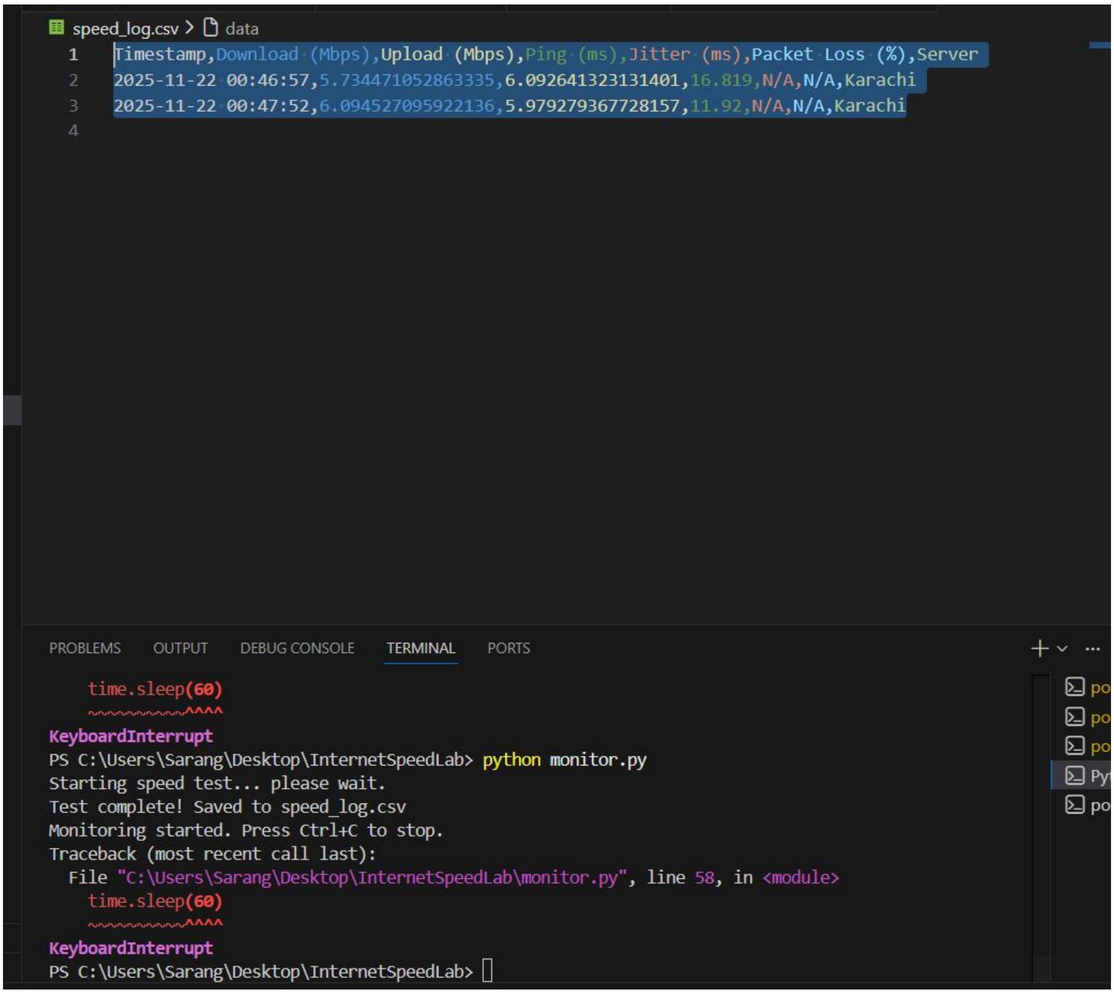
TASK 3.2



```
    # NOTE: You must run this script with Administrator/sudo
    # because the Nmap scan (Task 1) requires elevated permis

    print("\n==============================================="
    print("      Task 3.2: Scan + Traffic Monitor")
    print("==============================================")

    host_discovery_scan()

    # Give the user a prompt to generate traffic during the r
    print("\n\n*** ACTION REQUIRED ***")
    print("Please open a few websites or start a download in
    time.sleep(2)

    traffic_spike_monitor(INTERFACE_NAME, MONITOR_SAMPLES, MO

    print("\n\n--- Lab Complete ---")
    print("You have successfully run all network analysis scr
```

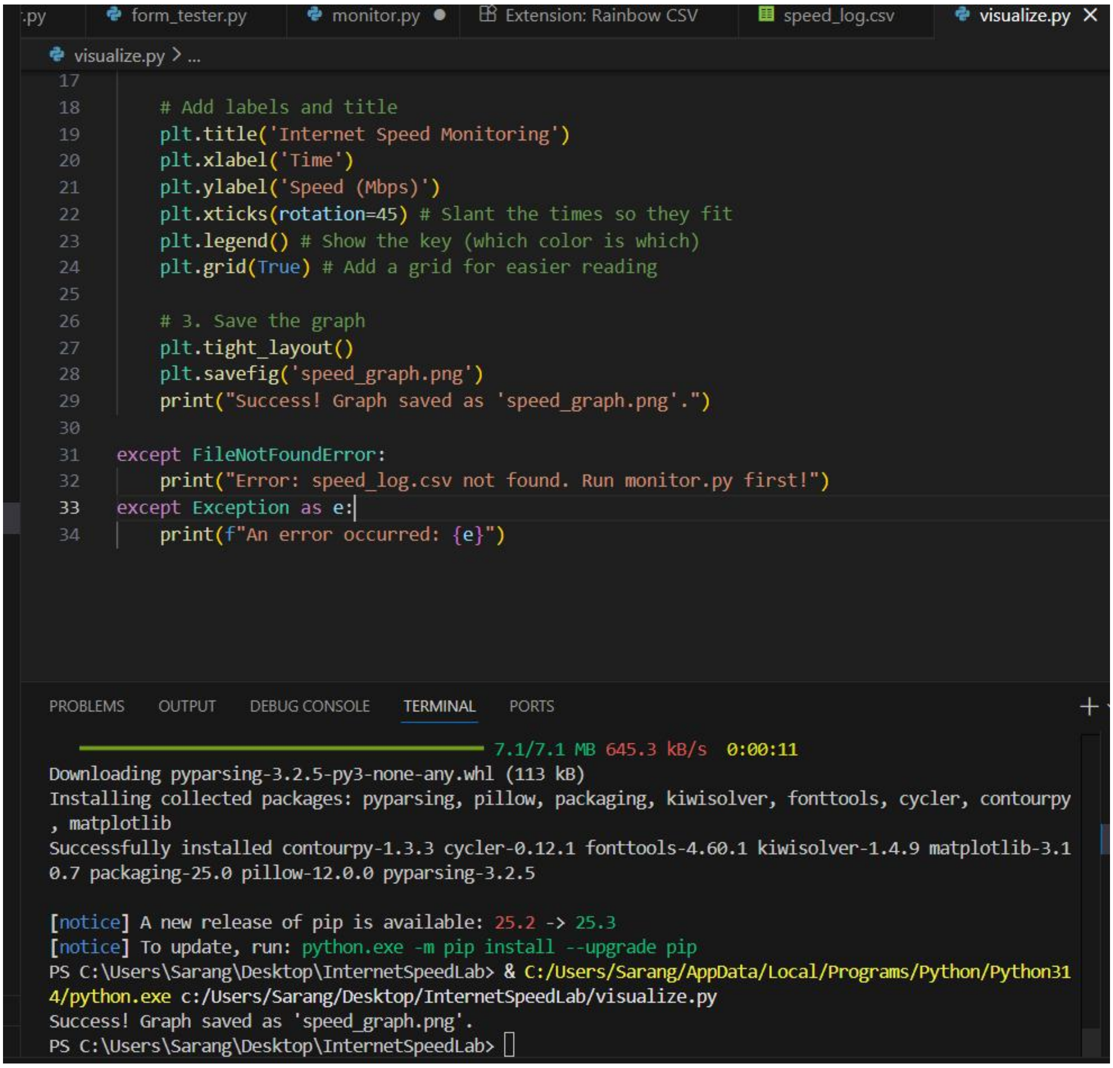
```
    - 192.168.100.237


*** ACTION REQUIRED ***
Please open a few websites or start a download in the next 60 seconds t

[!!!] Configuration Error: You must update 'INTERFACE_NAME' in the scri
Please run the command from Step 1 to find the correct name, then updat


--- Lab Complete ---
You have successfully run all network analysis scripts!
PS C:\Users\Sarang\Desktop\InternetSpeedLab>
```

LAB -2 Internet Speed Monitoring Report



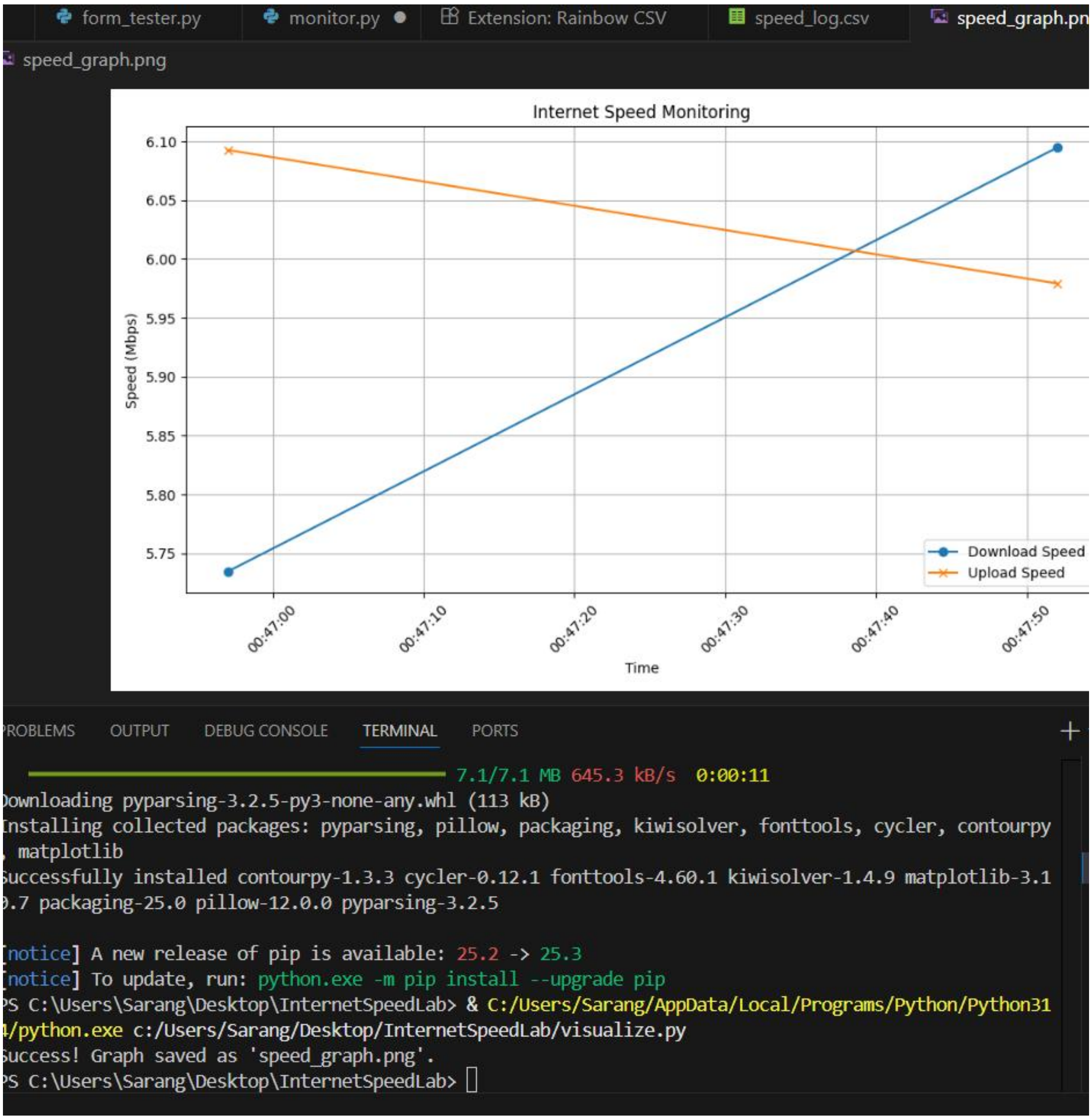
```
Timestamp,Download (Mbps),Upload (Mbps),Ping (ms),Jitter (ms),Packet Loss (%),Server
2025-11-22 00:46:57,5.734471052863335,6.092641323131401,16.819,N/A,N/A,Karachi
2025-11-22 00:47:52,6.094527095922136,5.979279367728157,11.92,N/A,N/A,Karachi
```

```
    time.sleep(60)
    ~~~~~~~~~~^^^^
KeyboardInterrupt
PS C:\Users\Sarang\Desktop\InternetSpeedLab> python monitor.py
Starting speed test... please wait.
Test complete! Saved to speed_log.csv
Monitoring started. Press Ctrl+C to stop.
Traceback (most recent call last):
  File "C:\Users\Sarang\Desktop\InternetSpeedLab\monitor.py", line 58, in <module>
    time.sleep(60)
    ~~~~~~~~~~^^^^
KeyboardInterrupt
PS C:\Users\Sarang\Desktop\InternetSpeedLab> 
```

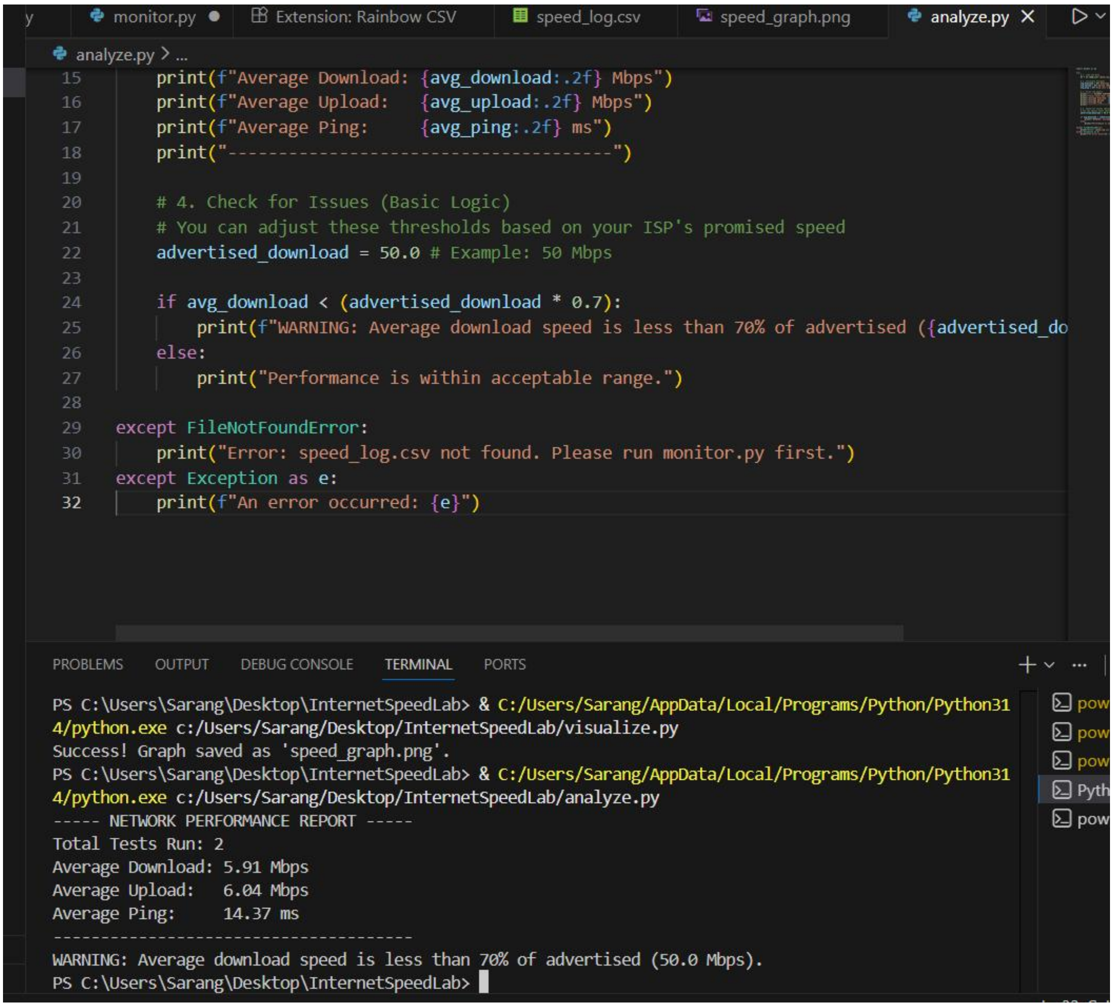


```python

    # Add labels and title
    plt.title('Internet Speed Monitoring')
    plt.xlabel('Time')
    plt.ylabel('Speed (Mbps)')
    plt.xticks(rotation=45) # Slant the times so they fit
    plt.legend() # Show the key (which color is which)
    plt.grid(True) # Add a grid for easier reading

    # 3. Save the graph
    plt.tight_layout()
    plt.savefig('speed_graph.png')
    print("Success! Graph saved as 'speed_graph.png'.")

except FileNotFoundError:
    print("Error: speed_log.csv not found. Run monitor.py first!")
except Exception as e:
    print(f"An error occurred: {e}")
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.1/7.1 MB 645.3 kB/s  0:00:11
Downloading pyparsing-3.2.5-py3-none-any.whl (113 kB)
Installing collected packages: pyparsing, pillow, packaging, kiwisolver, fonttools, cycler, contourpy
, matplotlib
Successfully installed contourpy-1.3.3 cycler-0.12.1 fonttools-4.60.1 kiwisolver-1.4.9 matplotlib-3.1
0.7 packaging-25.0 pillow-12.0.0 pyparsing-3.2.5

[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\Sarang\Desktop\InternetSpeedLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python31
4/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/visualize.py
Success! Graph saved as 'speed_graph.png'.
PS C:\Users\Sarang\Desktop\InternetSpeedLab>
```



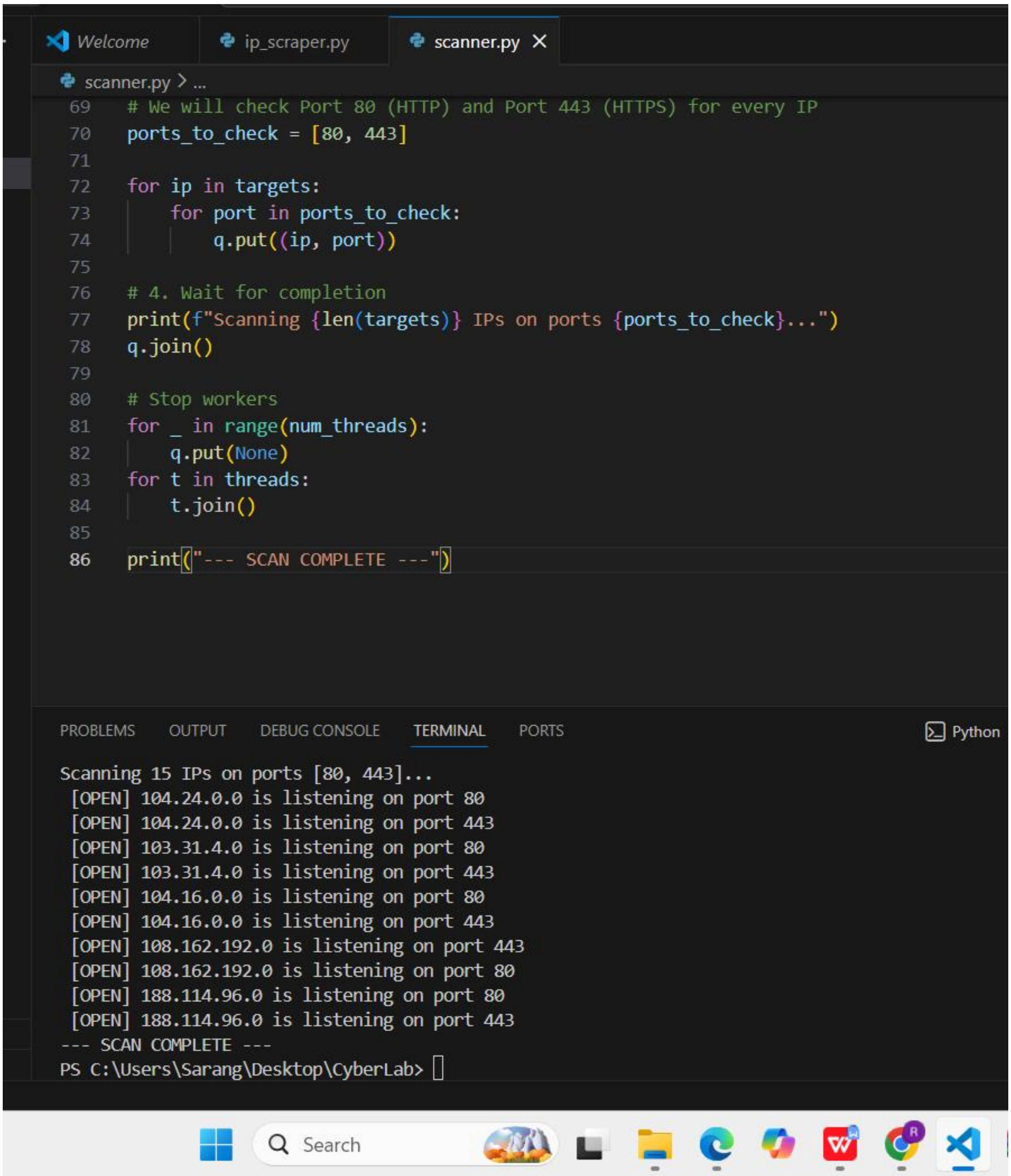
```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.1/7.1 MB 645.3 kB/s  0:00:11
ownloading pyparsing-3.2.5-py3-none-any.whl (113 kB)
nstalling collected packages: pyparsing, pillow, packaging, kiwisolver, fonttools, cycler, contourpy
 matplotlib
uccessfully installed contourpy-1.3.3 cycler-0.12.1 fonttools-4.60.1 kiwisolver-1.4.9 matplotlib-3.1
.7 packaging-25.0 pillow-12.0.0 pyparsing-3.2.5

notice] A new release of pip is available: 25.2 -> 25.3
notice] To update, run: python.exe -m pip install --upgrade pip
S C:\Users\Sarang\Desktop\InternetSpeedLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python31
/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/visualize.py
uccess! Graph saved as 'speed_graph.png'.
S C:\Users\Sarang\Desktop\InternetSpeedLab> 
```



```
        print(f"Average Download: {avg_download:.2f} Mbps")
        print(f"Average Upload:   {avg_upload:.2f} Mbps")
        print(f"Average Ping:     {avg_ping:.2f} ms")
        print("--------------------------------------")

        # 4. Check for Issues (Basic Logic)
        # You can adjust these thresholds based on your ISP's promised speed
        advertised_download = 50.0 # Example: 50 Mbps

        if avg_download < (advertised_download * 0.7):
            print(f"WARNING: Average download speed is less than 70% of advertised ({advertised_do
        else:
            print("Performance is within acceptable range.")

    except FileNotFoundError:
        print("Error: speed_log.csv not found. Please run monitor.py first.")
    except Exception as e:
        print(f"An error occurred: {e}")
```

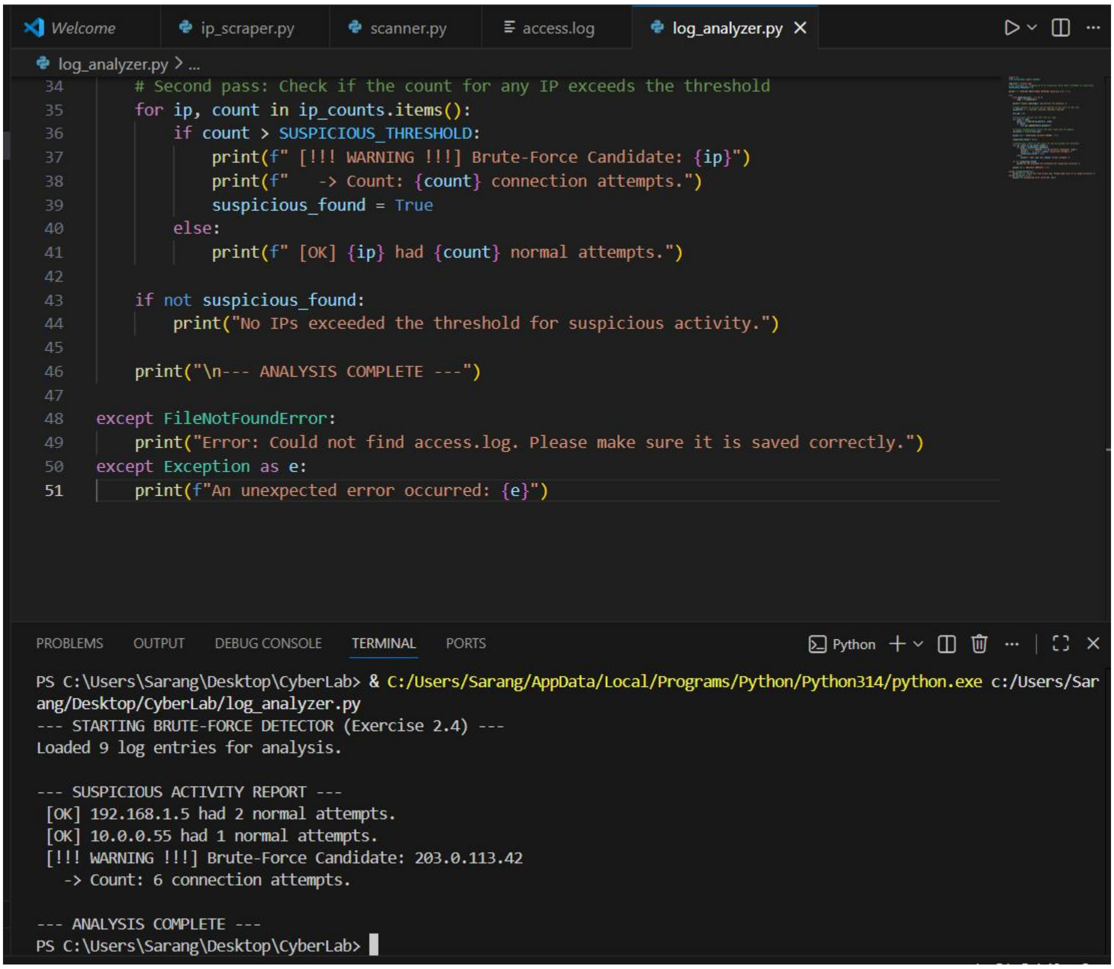
```
PS C:\Users\Sarang\Desktop\InternetSpeedLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python31
4/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/visualize.py
Success! Graph saved as 'speed_graph.png'.
PS C:\Users\Sarang\Desktop\InternetSpeedLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python31
4/python.exe c:/Users/Sarang/Desktop/InternetSpeedLab/analyze.py
----- NETWORK PERFORMANCE REPORT -----
Total Tests Run: 2
Average Download: 5.91 Mbps
Average Upload:   6.04 Mbps
Average Ping:     14.37 ms
--------------------------------------
WARNING: Average download speed is less than 70% of advertised (50.0 Mbps).
PS C:\Users\Sarang\Desktop\InternetSpeedLab> 
```

LAB-03 Lab Handout: Python Shell Scripting for Cyber security- Web Scraping and Forensics



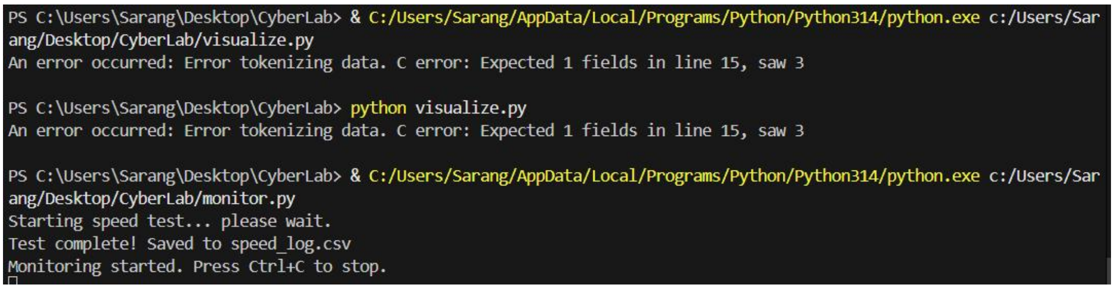
```python
# We will check Port 80 (HTTP) and Port 443 (HTTPS) for every IP
ports_to_check = [80, 443]

for ip in targets:
    for port in ports_to_check:
        q.put((ip, port))

# 4. Wait for completion
print(f"Scanning {len(targets)} IPs on ports {ports_to_check}...")
q.join()

# Stop workers
for _ in range(num_threads):
    q.put(None)
for t in threads:
    t.join()

print("--- SCAN COMPLETE ---")
```

```
Scanning 15 IPs on ports [80, 443]...
 [OPEN] 104.24.0.0 is listening on port 80
 [OPEN] 104.24.0.0 is listening on port 443
 [OPEN] 103.31.4.0 is listening on port 80
 [OPEN] 103.31.4.0 is listening on port 443
 [OPEN] 104.16.0.0 is listening on port 80
 [OPEN] 104.16.0.0 is listening on port 443
 [OPEN] 108.162.192.0 is listening on port 443
 [OPEN] 108.162.192.0 is listening on port 80
 [OPEN] 188.114.96.0 is listening on port 80
 [OPEN] 188.114.96.0 is listening on port 443
--- SCAN COMPLETE ---
PS C:\Users\Sarang\Desktop\CyberLab>
```

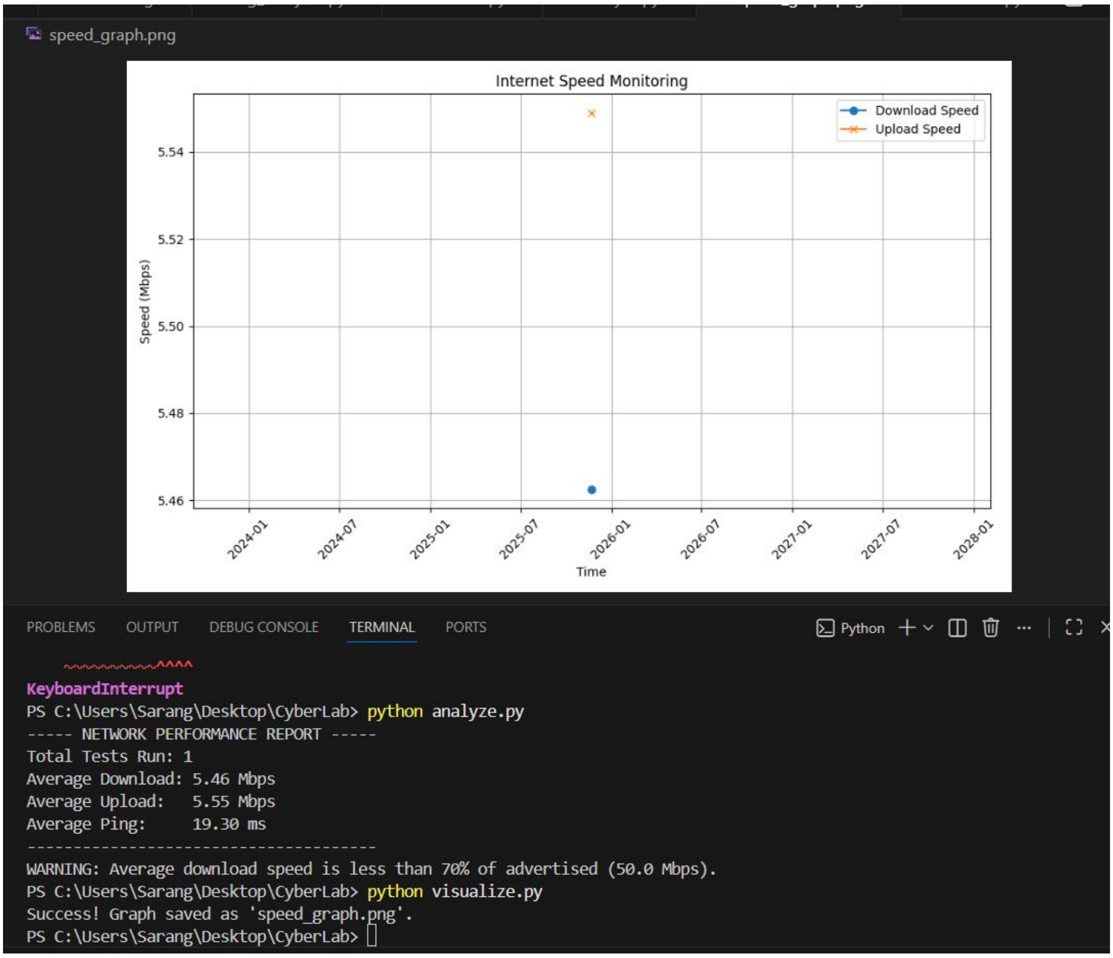


```python
    # Second pass: Check if the count for any IP exceeds the threshold
    for ip, count in ip_counts.items():
        if count > SUSPICIOUS_THRESHOLD:
            print(f" [!!! WARNING !!!] Brute-Force Candidate: {ip}")
            print(f"   -> Count: {count} connection attempts.")
            suspicious_found = True
        else:
            print(f" [OK] {ip} had {count} normal attempts.")

    if not suspicious_found:
        print("No IPs exceeded the threshold for suspicious activity.")

    print("\n--- ANALYSIS COMPLETE ---")

except FileNotFoundError:
    print("Error: Could not find access.log. Please make sure it is saved correctly.")
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

```
PS C:\Users\Sarang\Desktop\CyberLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python314/python.exe c:/Users/Sar
ang/Desktop/CyberLab/log_analyzer.py
--- STARTING BRUTE-FORCE DETECTOR (Exercise 2.4) ---
Loaded 9 log entries for analysis.

--- SUSPICIOUS ACTIVITY REPORT ---
 [OK] 192.168.1.5 had 2 normal attempts.
 [OK] 10.0.0.55 had 1 normal attempts.
 [!!! WARNING !!!] Brute-Force Candidate: 203.0.113.42
   -> Count: 6 connection attempts.

--- ANALYSIS COMPLETE ---
PS C:\Users\Sarang\Desktop\CyberLab>
```



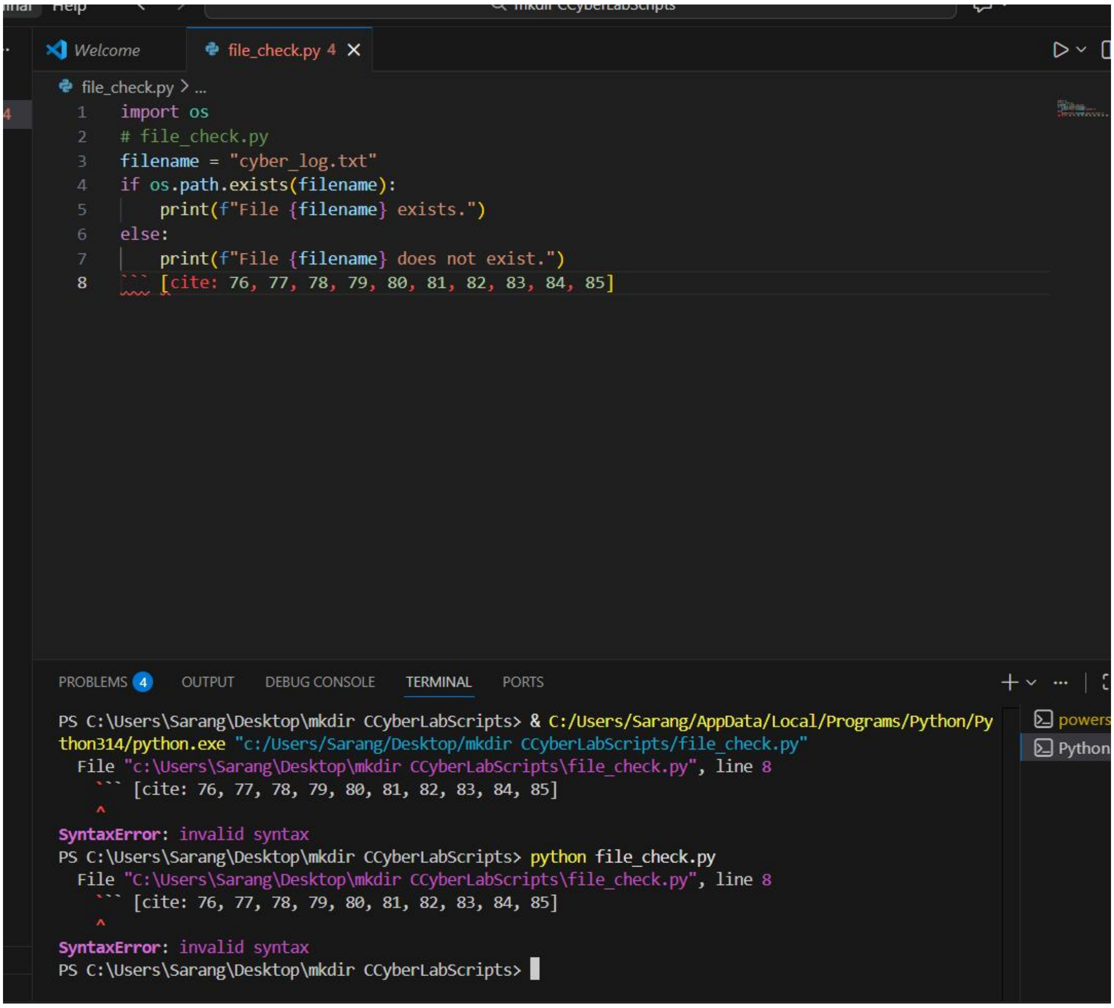
```
PS C:\Users\Sarang\Desktop\CyberLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python314/python.exe c:/Users/Sar
ang/Desktop/CyberLab/visualize.py
An error occurred: Error tokenizing data. C error: Expected 1 fields in line 15, saw 3

PS C:\Users\Sarang\Desktop\CyberLab> python visualize.py
An error occurred: Error tokenizing data. C error: Expected 1 fields in line 15, saw 3

PS C:\Users\Sarang\Desktop\CyberLab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python314/python.exe c:/Users/Sar
ang/Desktop/CyberLab/monitor.py
Starting speed test... please wait.
Test complete! Saved to speed_log.csv
Monitoring started. Press Ctrl+C to stop.
```



```
    ~~~~~~~~~~~~^^^^
KeyboardInterrupt
PS C:\Users\Sarang\Desktop\CyberLab> python analyze.py
----- NETWORK PERFORMANCE REPORT -----
Total Tests Run: 1
Average Download: 5.46 Mbps
Average Upload:   5.55 Mbps
Average Ping:     19.30 ms
--------------------------------------
WARNING: Average download speed is less than 70% of advertised (50.0 Mbps).
PS C:\Users\Sarang\Desktop\CyberLab> python visualize.py
Success! Graph saved as 'speed_graph.png'.
PS C:\Users\Sarang\Desktop\CyberLab>
```

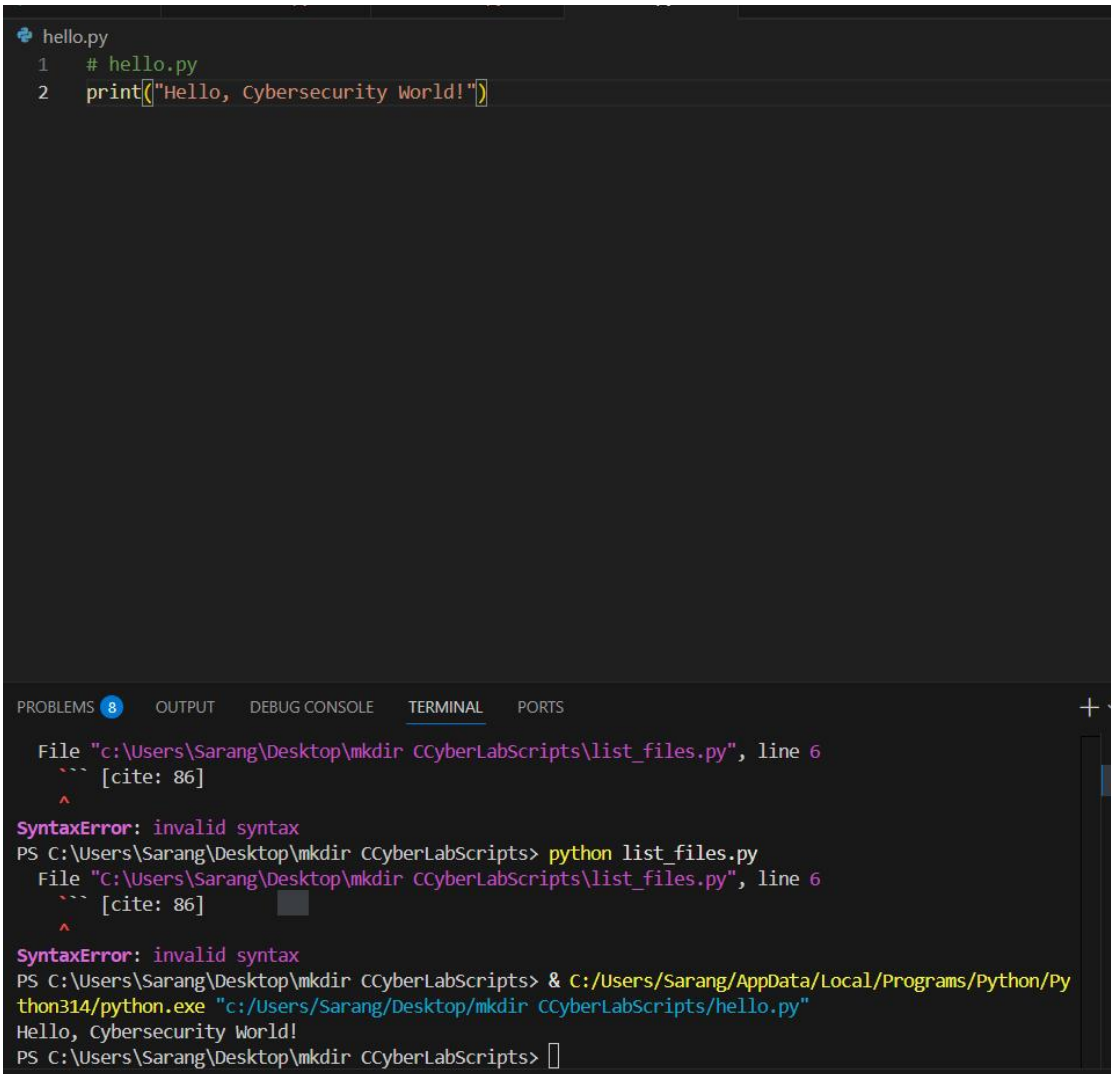
Conclusion: Your script successfully identified a significant issue where the actual measured performance **(5.46 Mbps)** is far below the assumed advertised rate **(50.0 Mbps)**, which is the primary goal of this lab section.

LAB-4 SHELL SCRIPTING WITH PYTHON FOR CYBERSECURITY



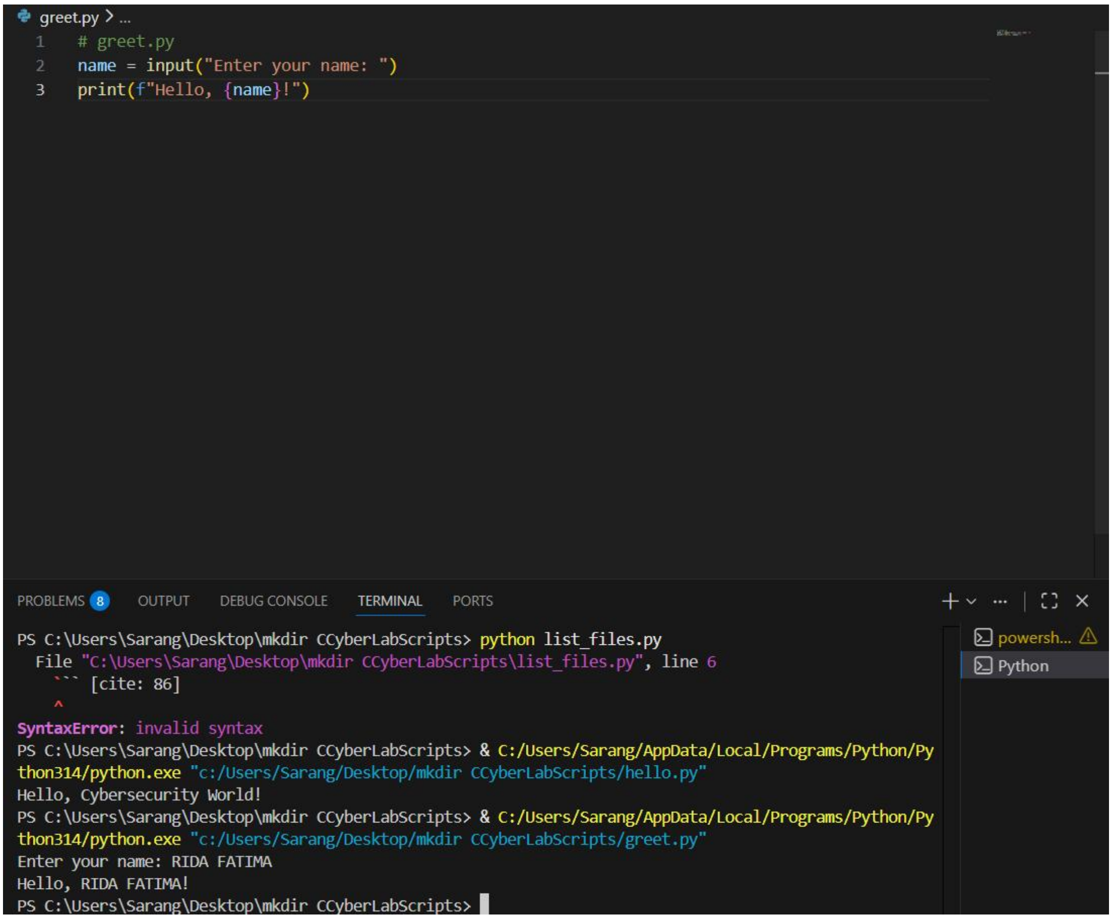
~~~~
import os
# file_check.py
filename = "cyber_log.txt"
if os.path.exists(filename):
    print(f"File {filename} exists.")
else:
    print(f"File {filename} does not exist.")
``` [cite: 76, 77, 78, 79, 80, 81, 82, 83, 84, 85]
~~~~

~~~~
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Py
thon314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/file_check.py"
  File "c:\Users\Sarang\Desktop\mkdir CCyberLabScripts\file_check.py", line 8
    ``` [cite: 76, 77, 78, 79, 80, 81, 82, 83, 84, 85]
    ^
SyntaxError: invalid syntax
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> python file_check.py
  File "C:\Users\Sarang\Desktop\mkdir CCyberLabScripts\file_check.py", line 8
    ``` [cite: 76, 77, 78, 79, 80, 81, 82, 83, 84, 85]
    ^
SyntaxError: invalid syntax
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts>
~~~~



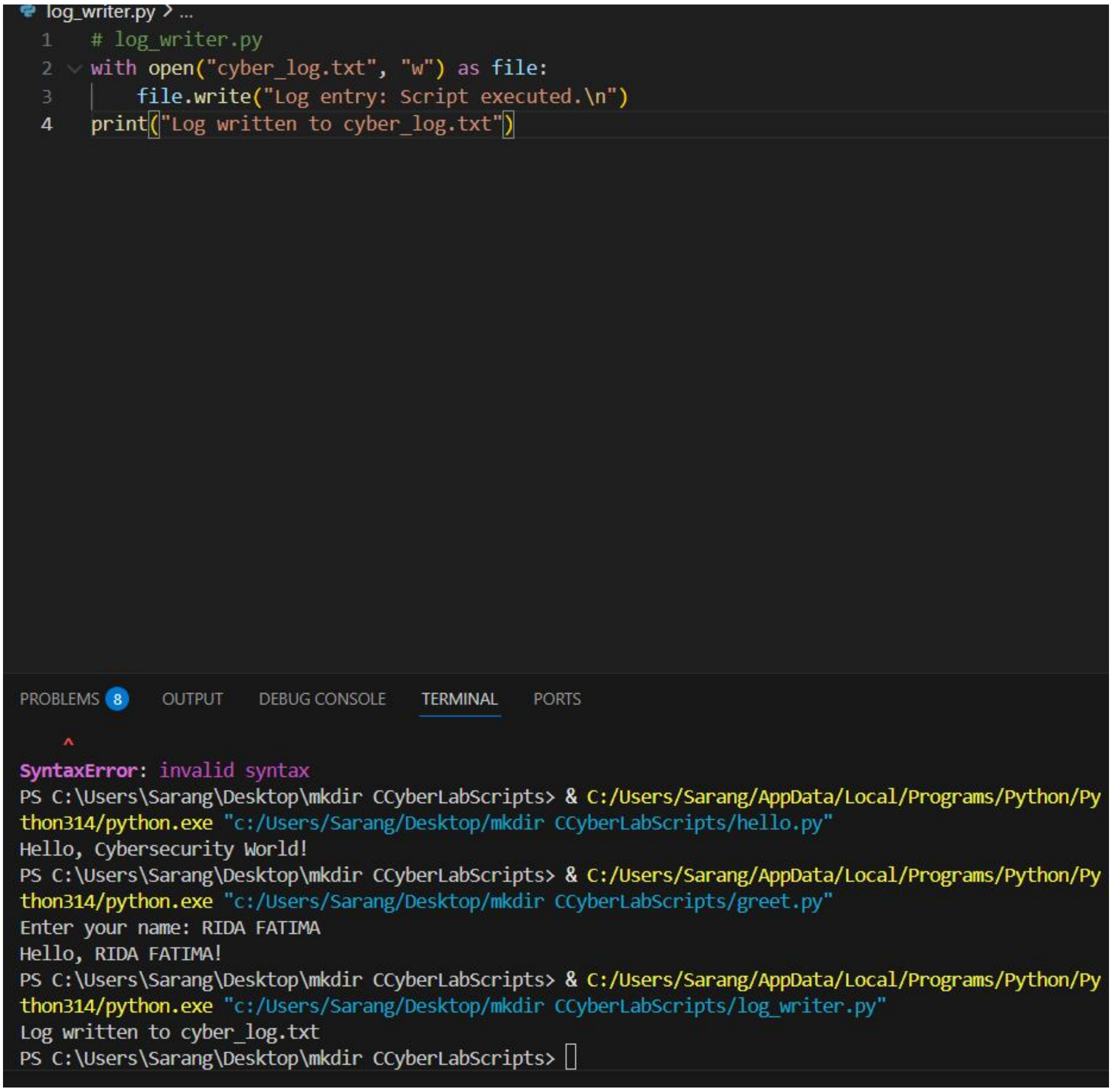
```
# hello.py
print("Hello, Cybersecurity World!")
```

~~~
  File "c:\Users\Sarang\Desktop\mkdir CCyberLabScripts\list_files.py", line 6
    ``` [cite: 86]
    ^
SyntaxError: invalid syntax
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> python list_files.py
  File "c:\Users\Sarang\Desktop\mkdir CCyberLabScripts\list_files.py", line 6
    ``` [cite: 86]
    ^
SyntaxError: invalid syntax
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Py
thon314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/hello.py"
Hello, Cybersecurity World!
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> 
~~~



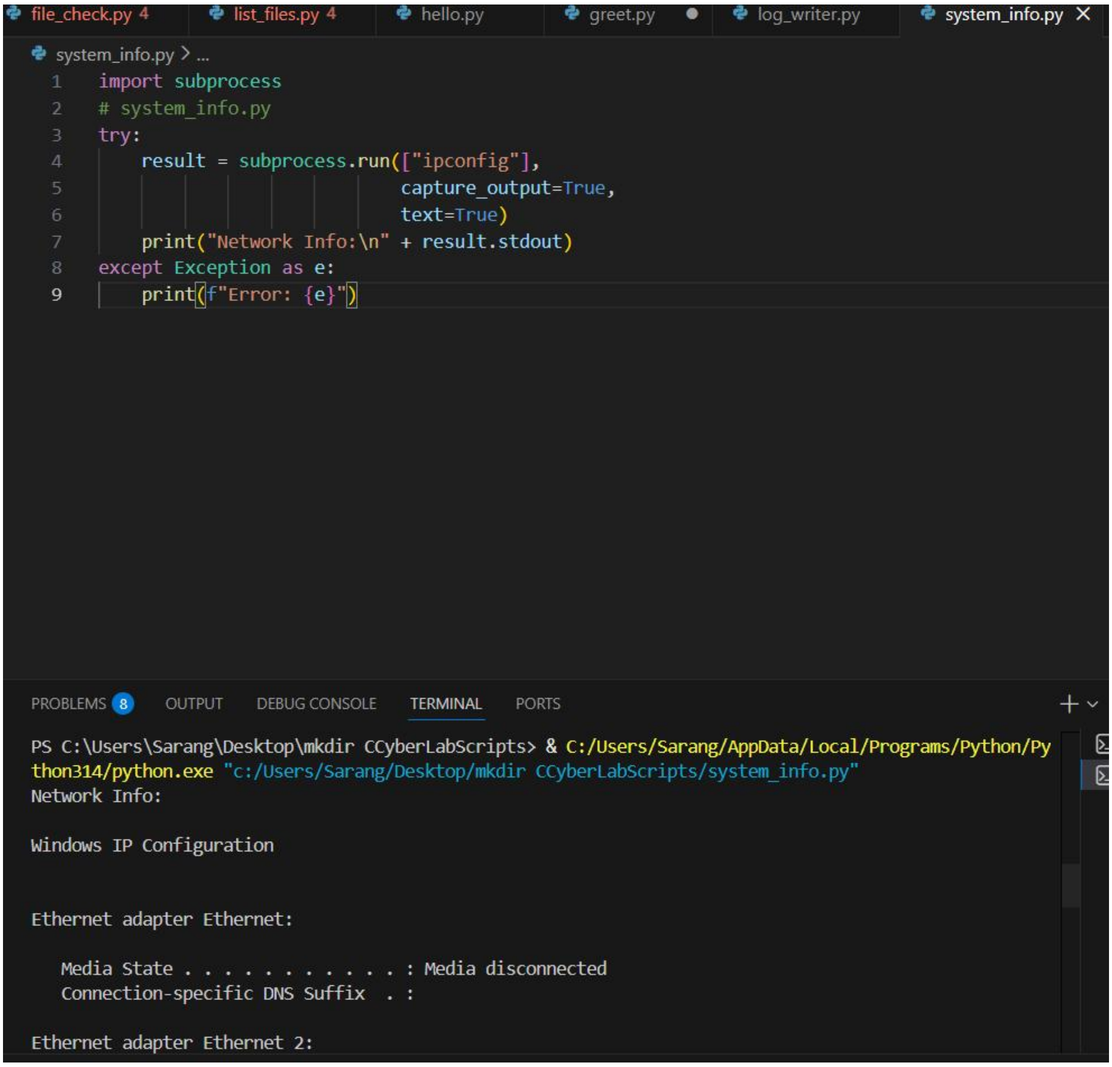
```python
# greet.py
name = input("Enter your name: ")
print(f"Hello, {name}!")
```

~~~
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> python list_files.py
  File "C:\Users\Sarang\Desktop\mkdir CCyberLabScripts\list_files.py", line 6
    ``` [cite: 86]
    ^
SyntaxError: invalid syntax
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Py
thon314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/hello.py"
Hello, Cybersecurity World!
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Py
thon314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/greet.py"
Enter your name: RIDA FATIMA
Hello, RIDA FATIMA!
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts>
~~~



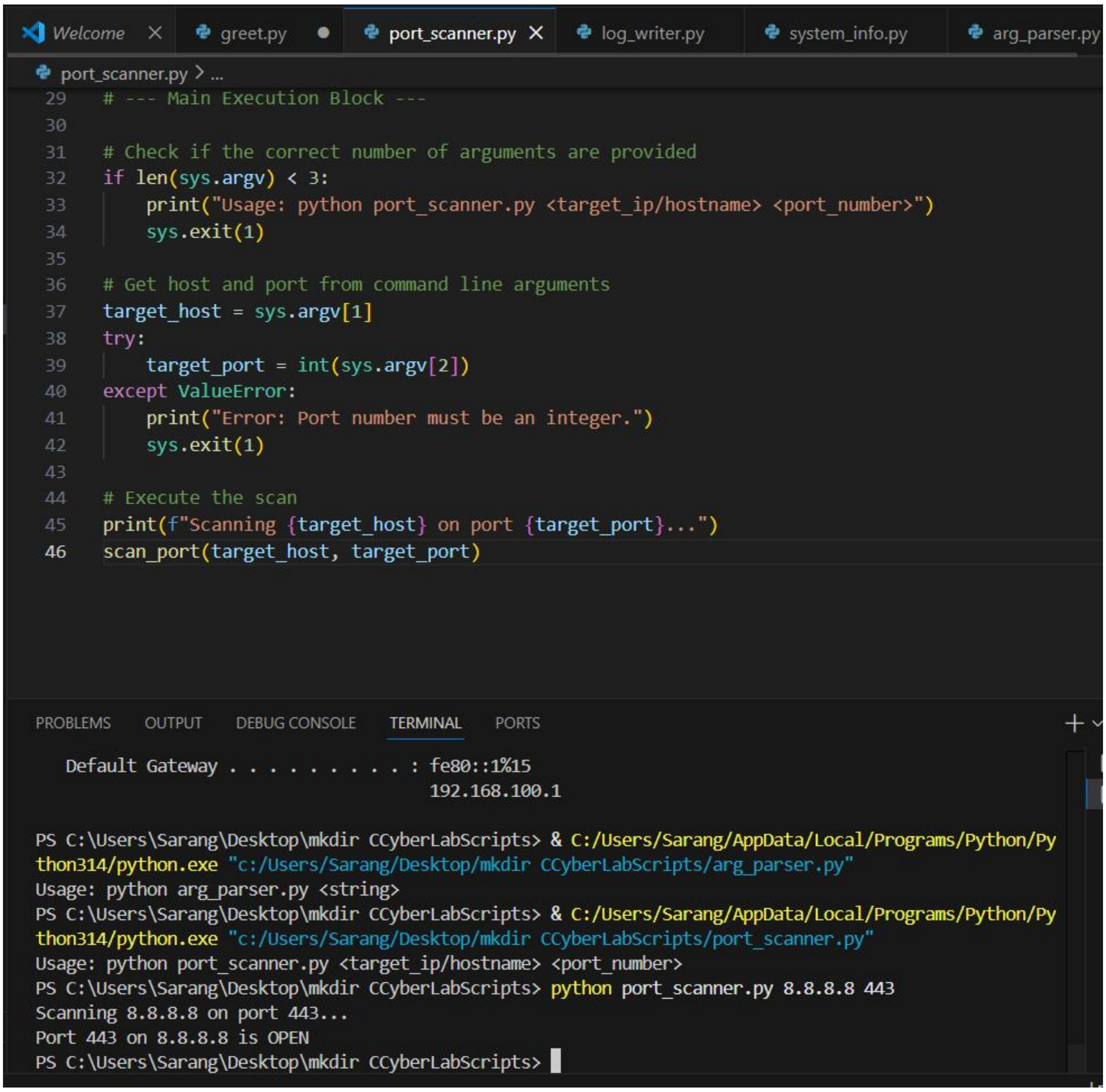
```python
# log_writer.py
with open("cyber_log.txt", "w") as file:
    file.write("Log entry: Script executed.\n")
print("Log written to cyber_log.txt")
```

```
    ^
SyntaxError: invalid syntax
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/hello.py"
Hello, Cybersecurity World!
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/greet.py"
Enter your name: RIDA FATIMA
Hello, RIDA FATIMA!
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/log_writer.py"
Log written to cyber_log.txt
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> 
```



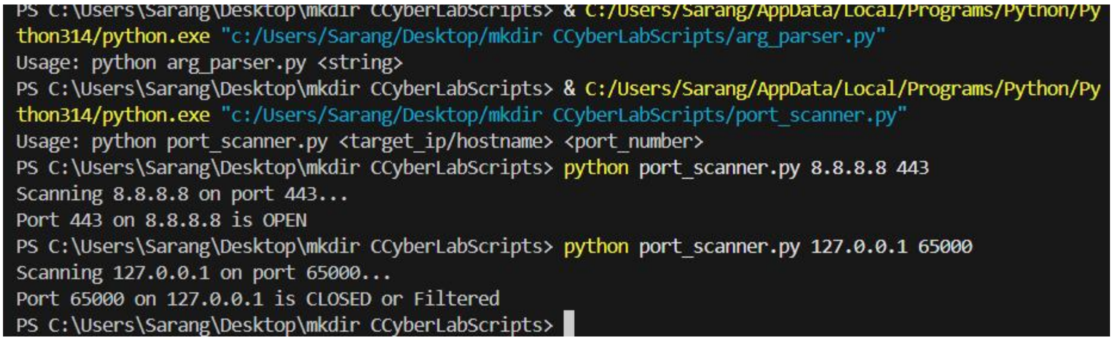
```python
import subprocess
# system_info.py
try:
    result = subprocess.run(["ipconfig"],
                            capture_output=True,
                            text=True)
    print("Network Info:\n" + result.stdout)
except Exception as e:
    print(f"Error: {e}")
```

```
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Py
thon314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/system_info.py"
Network Info:

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:
```



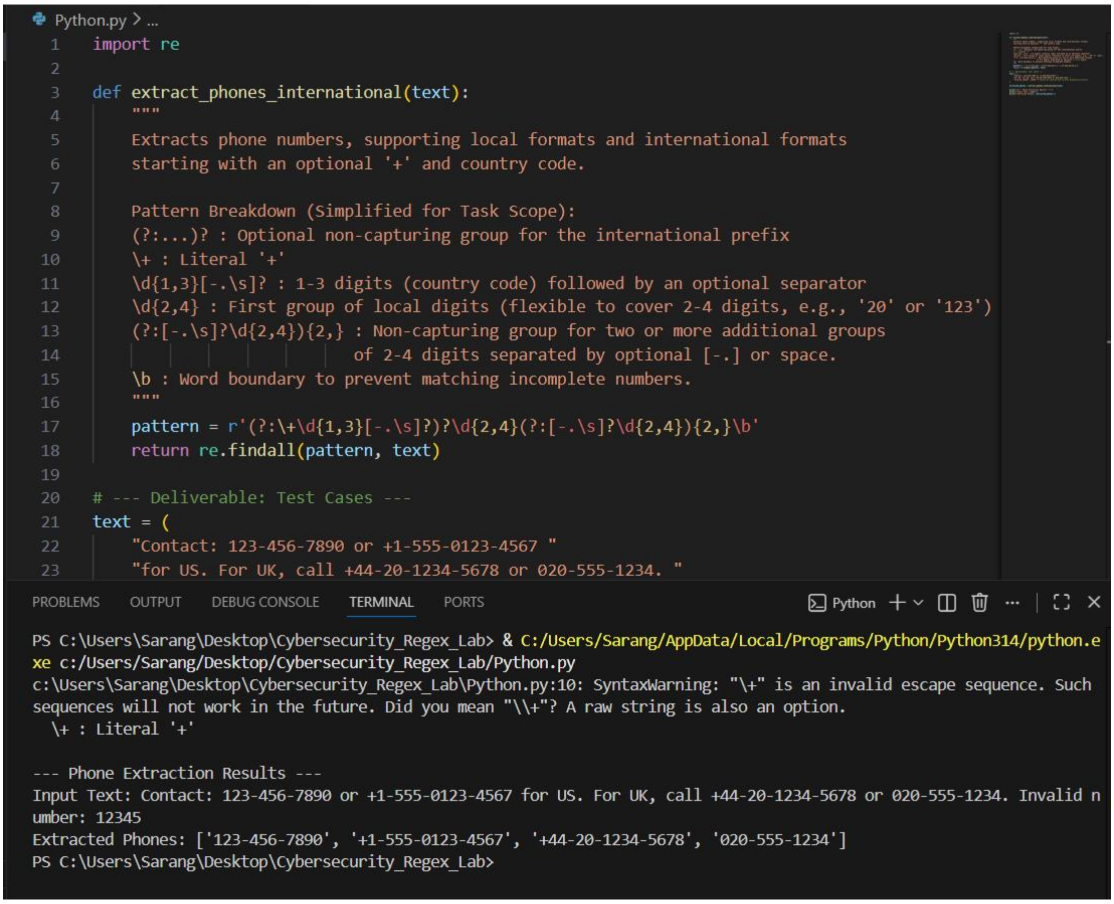
```python
# --- Main Execution Block ---

# Check if the correct number of arguments are provided
if len(sys.argv) < 3:
    print("Usage: python port_scanner.py <target_ip/hostname> <port_number>")
    sys.exit(1)

# Get host and port from command line arguments
target_host = sys.argv[1]
try:
    target_port = int(sys.argv[2])
except ValueError:
    print("Error: Port number must be an integer.")
    sys.exit(1)

# Execute the scan
print(f"Scanning {target_host} on port {target_port}...")
scan_port(target_host, target_port)
```

```
   Default Gateway . . . . . . . . . : fe80::1%15
                                       192.168.100.1

PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Py
thon314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/arg_parser.py"
Usage: python arg_parser.py <string>
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Py
thon314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/port_scanner.py"
Usage: python port_scanner.py <target_ip/hostname> <port_number>
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> python port_scanner.py 8.8.8.8 443
Scanning 8.8.8.8 on port 443...
Port 443 on 8.8.8.8 is OPEN
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> 
```



```
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Py
thon314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/arg_parser.py"
Usage: python arg_parser.py <string>
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> & C:/Users/Sarang/AppData/Local/Programs/Python/Py
thon314/python.exe "c:/Users/Sarang/Desktop/mkdir CCyberLabScripts/port_scanner.py"
Usage: python port_scanner.py <target_ip/hostname> <port_number>
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> python port_scanner.py 8.8.8.8 443
Scanning 8.8.8.8 on port 443...
Port 443 on 8.8.8.8 is OPEN
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> python port_scanner.py 127.0.0.1 65000
Scanning 127.0.0.1 on port 65000...
Port 65000 on 127.0.0.1 is CLOSED or Filtered
PS C:\Users\Sarang\Desktop\mkdir CCyberLabScripts> 
```

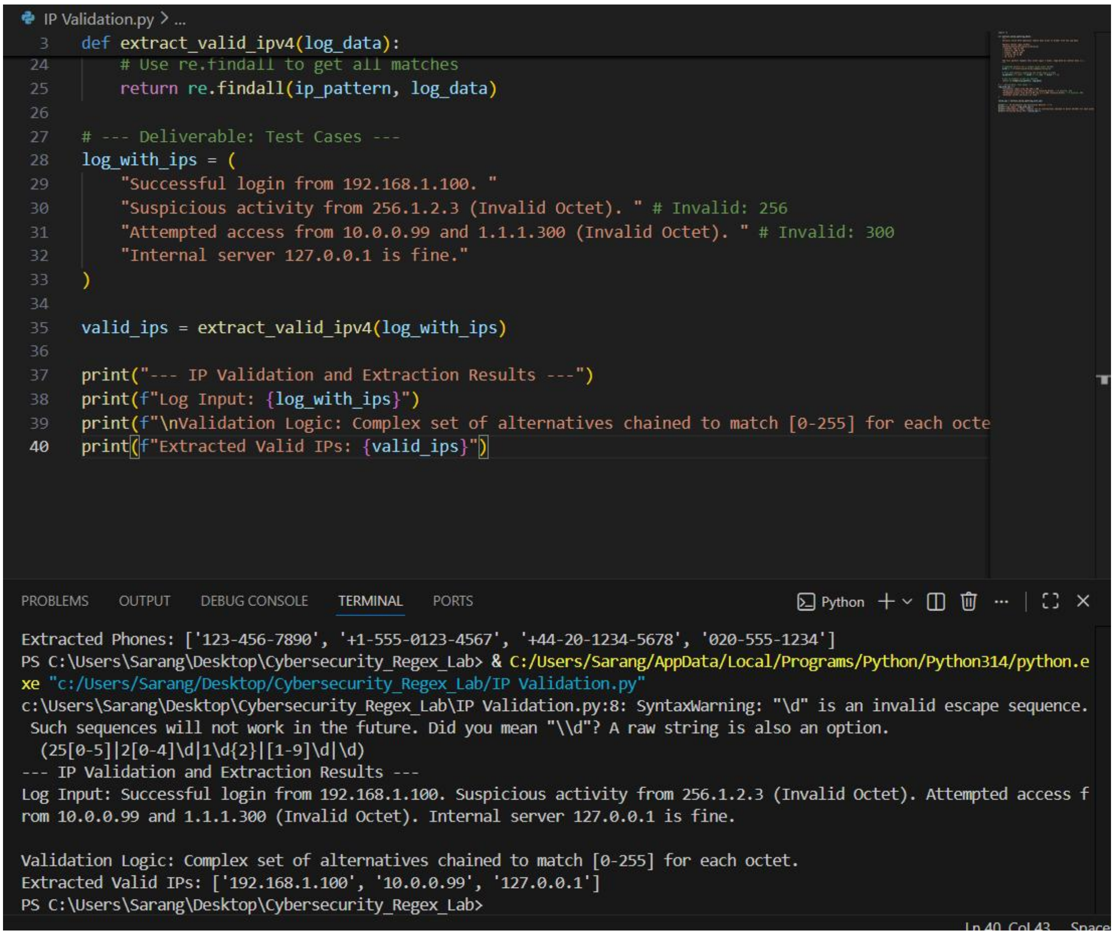
LAB-05 Regular Expressions in Python for Cybersecurity Applications
Task-1



```python
import re

def extract_phones_international(text):
    """
    Extracts phone numbers, supporting local formats and international formats
    starting with an optional '+' and country code.

    Pattern Breakdown (Simplified for Task Scope):
    (?:...)? : Optional non-capturing group for the international prefix
    \+ : Literal '+'
    \d{1,3}[-.\s]? : 1-3 digits (country code) followed by an optional separator
    \d{2,4} : First group of local digits (flexible to cover 2-4 digits, e.g., '20' or '123')
    (?:[-.\s]?\d{2,4}){2,} : Non-capturing group for two or more additional groups
                           of 2-4 digits separated by optional [-.] or space.
    \b : Word boundary to prevent matching incomplete numbers.
    """
    pattern = r'(?:\+\d{1,3}[-.\s]?)?\d{2,4}(?:[-.\s]?\d{2,4}){2,}\b'
    return re.findall(pattern, text)

# --- Deliverable: Test Cases ---
text = (
    "Contact: 123-456-7890 or +1-555-0123-4567 "
    "for US. For UK, call +44-20-1234-5678 or 020-555-1234. "
```

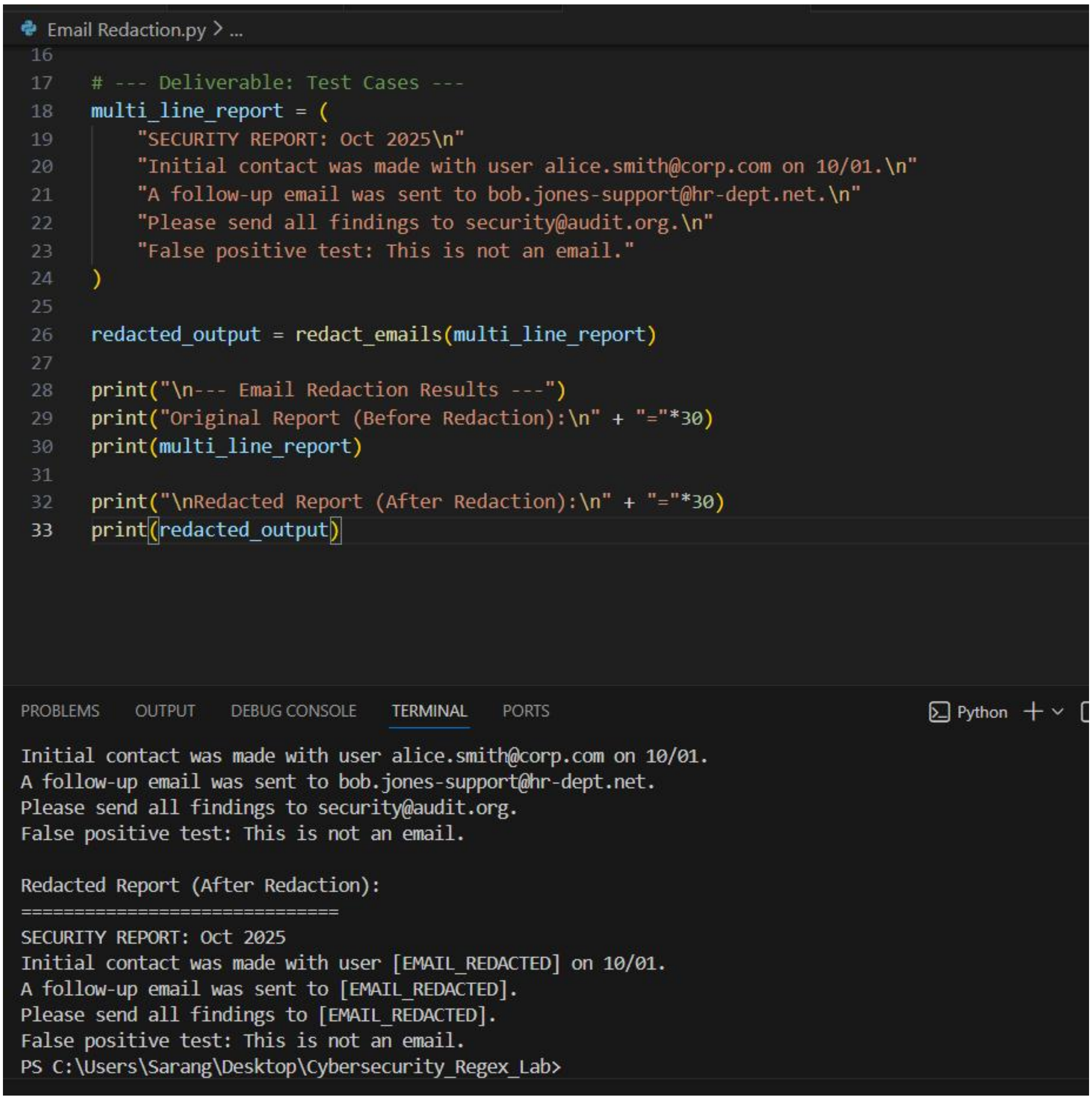
```
PS C:\Users\Sarang\Desktop\Cybersecurity_Regex_Lab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python314/python.e
xe c:/Users/Sarang/Desktop/Cybersecurity_Regex_Lab/Python.py
c:\Users\Sarang\Desktop\Cybersecurity_Regex_Lab\Python.py:10: SyntaxWarning: "\+" is an invalid escape sequence. Such
sequences will not work in the future. Did you mean "\\+"? A raw string is also an option.
  \+ : Literal '+'

--- Phone Extraction Results ---
Input Text: Contact: 123-456-7890 or +1-555-0123-4567 for US. For UK, call +44-20-1234-5678 or 020-555-1234. Invalid n
umber: 12345
Extracted Phones: ['123-456-7890', '+1-555-0123-4567', '+44-20-1234-5678', '020-555-1234']
PS C:\Users\Sarang\Desktop\Cybersecurity_Regex_Lab>
```

Task-2



```python
def extract_valid_ipv4(log_data):
    # Use re.findall to get all matches
    return re.findall(ip_pattern, log_data)

# --- Deliverable: Test Cases ---
log_with_ips = (
    "Successful login from 192.168.1.100. "
    "Suspicious activity from 256.1.2.3 (Invalid Octet). " # Invalid: 256
    "Attempted access from 10.0.0.99 and 1.1.1.300 (Invalid Octet). " # Invalid: 300
    "Internal server 127.0.0.1 is fine."
)

valid_ips = extract_valid_ipv4(log_with_ips)

print("--- IP Validation and Extraction Results ---")
print(f"Log Input: {log_with_ips}")
print(f"\nValidation Logic: Complex set of alternatives chained to match [0-255] for each octe
print(f"Extracted Valid IPs: {valid_ips}")
```

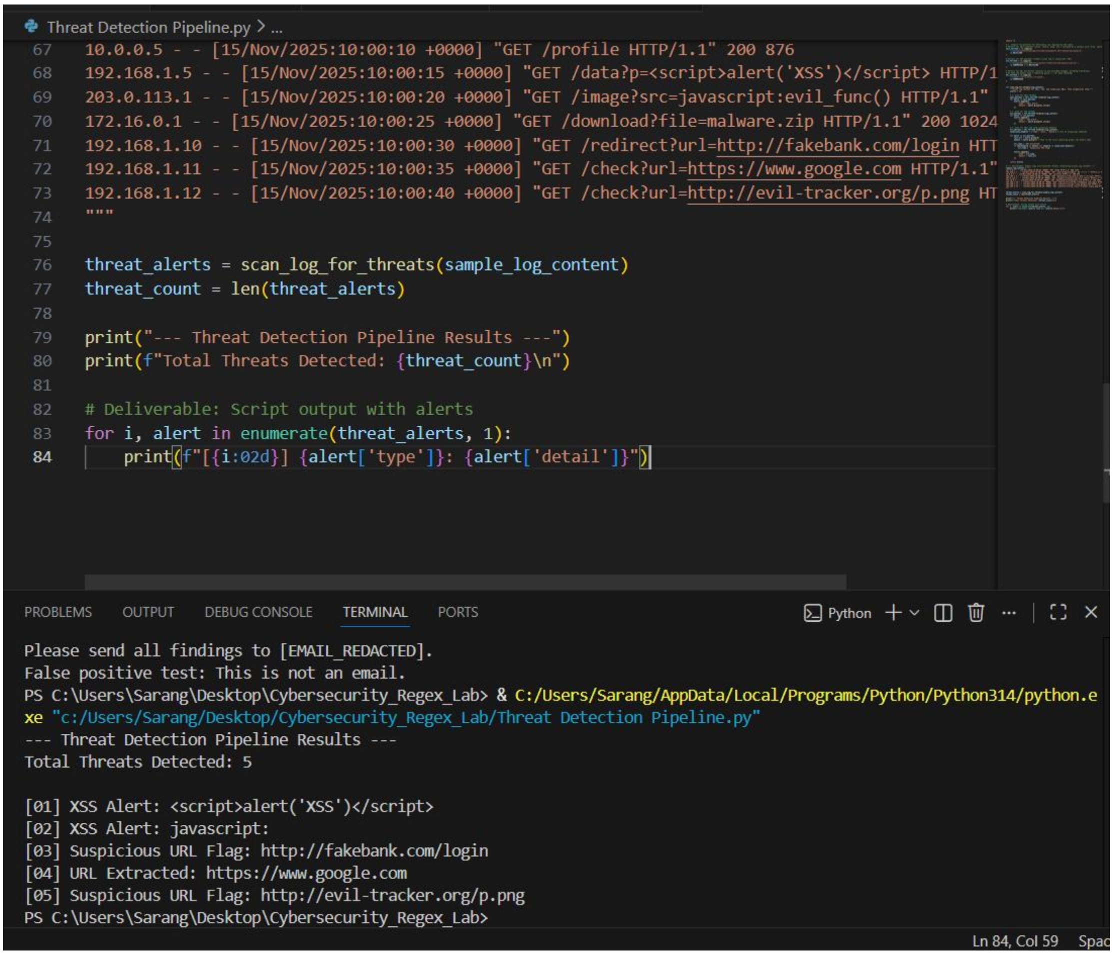
```
Extracted Phones: ['123-456-7890', '+1-555-0123-4567', '+44-20-1234-5678', '020-555-1234']
PS C:\Users\Sarang\Desktop\Cybersecurity_Regex_Lab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python314/python.e
xe "c:/Users/Sarang/Desktop/Cybersecurity_Regex_Lab/IP Validation.py"
c:\Users\Sarang\Desktop\Cybersecurity_Regex_Lab\IP Validation.py:8: SyntaxWarning: "\d" is an invalid escape sequence.
 Such sequences will not work in the future. Did you mean "\\d"? A raw string is also an option.
  (25[0-5]|2[0-4]\d|1\d{2}|[1-9]\d|\d)
--- IP Validation and Extraction Results ---
Log Input: Successful login from 192.168.1.100. Suspicious activity from 256.1.2.3 (Invalid Octet). Attempted access f
rom 10.0.0.99 and 1.1.1.300 (Invalid Octet). Internal server 127.0.0.1 is fine.

Validation Logic: Complex set of alternatives chained to match [0-255] for each octet.
Extracted Valid IPs: ['192.168.1.100', '10.0.0.99', '127.0.0.1']
PS C:\Users\Sarang\Desktop\Cybersecurity_Regex_Lab>
```



```python

# --- Deliverable: Test Cases ---
multi_line_report = (
    "SECURITY REPORT: Oct 2025\n"
    "Initial contact was made with user alice.smith@corp.com on 10/01.\n"
    "A follow-up email was sent to bob.jones-support@hr-dept.net.\n"
    "Please send all findings to security@audit.org.\n"
    "False positive test: This is not an email."
)

redacted_output = redact_emails(multi_line_report)

print("\n--- Email Redaction Results ---")
print("Original Report (Before Redaction):\n" + "="*30)
print(multi_line_report)

print("\nRedacted Report (After Redaction):\n" + "="*30)
print(redacted_output)
```

```
Initial contact was made with user alice.smith@corp.com on 10/01.
A follow-up email was sent to bob.jones-support@hr-dept.net.
Please send all findings to security@audit.org.
False positive test: This is not an email.

Redacted Report (After Redaction):
==============================
SECURITY REPORT: Oct 2025
Initial contact was made with user [EMAIL_REDACTED] on 10/01.
A follow-up email was sent to [EMAIL_REDACTED].
Please send all findings to [EMAIL_REDACTED].
False positive test: This is not an email.
PS C:\Users\Sarang\Desktop\Cybersecurity_Regex_Lab>
```

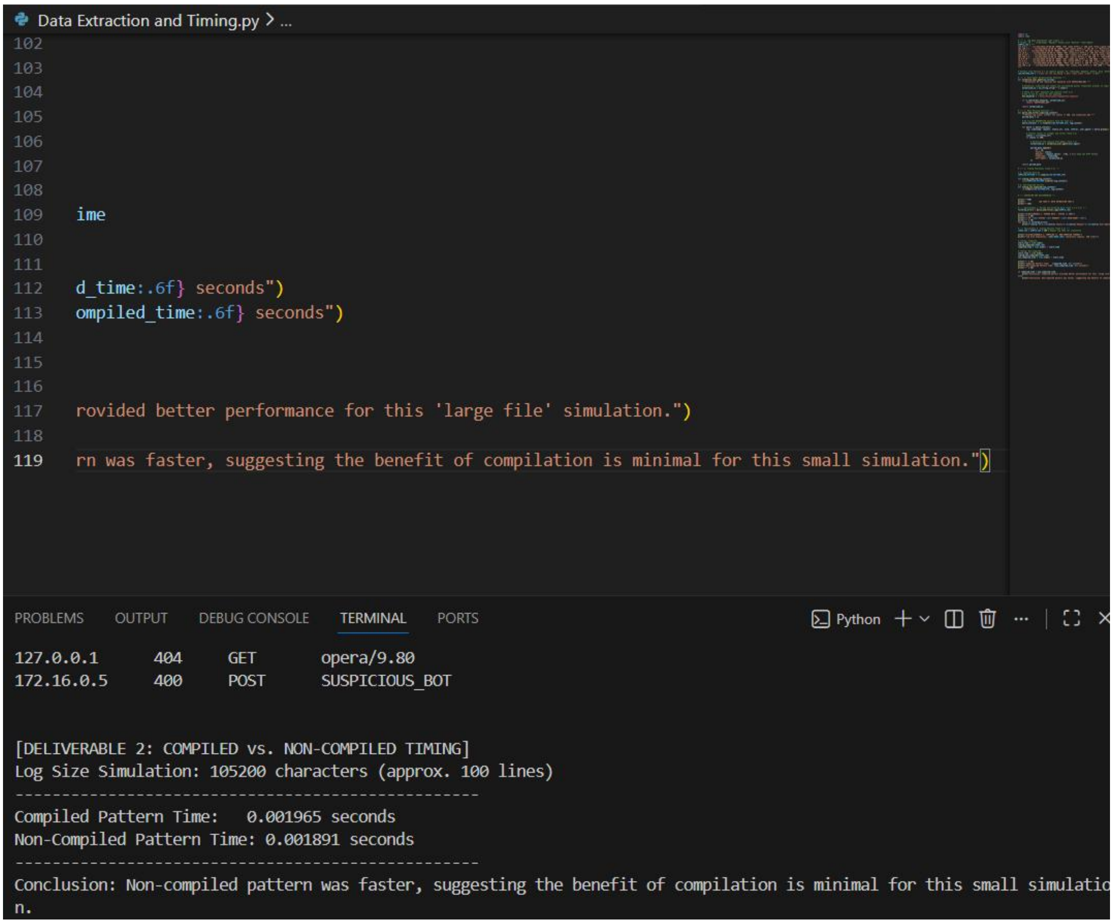
Task-3



```python
10.0.0.5 - - [15/Nov/2025:10:00:10 +0000] "GET /profile HTTP/1.1" 200 876
192.168.1.5 - - [15/Nov/2025:10:00:15 +0000] "GET /data?p=<script>alert('XSS')</script> HTTP/1
203.0.113.1 - - [15/Nov/2025:10:00:20 +0000] "GET /image?src=javascript:evil_func() HTTP/1.1"
172.16.0.1 - - [15/Nov/2025:10:00:25 +0000] "GET /download?file=malware.zip HTTP/1.1" 200 1024
192.168.1.10 - - [15/Nov/2025:10:00:30 +0000] "GET /redirect?url=http://fakebank.com/login HTT
192.168.1.11 - - [15/Nov/2025:10:00:35 +0000] "GET /check?url=https://www.google.com HTTP/1.1"
192.168.1.12 - - [15/Nov/2025:10:00:40 +0000] "GET /check?url=http://evil-tracker.org/p.png HT
"""

threat_alerts = scan_log_for_threats(sample_log_content)
threat_count = len(threat_alerts)

print("--- Threat Detection Pipeline Results ---")
print(f"Total Threats Detected: {threat_count}\n")

# Deliverable: Script output with alerts
for i, alert in enumerate(threat_alerts, 1):
    print(f"[{i:02d}] {alert['type']}: {alert['detail']}")
```

```
Please send all findings to [EMAIL_REDACTED].
False positive test: This is not an email.
PS C:\Users\Sarang\Desktop\Cybersecurity_Regex_Lab> & C:/Users/Sarang/AppData/Local/Programs/Python/Python314/python.e
xe "c:/Users/Sarang/Desktop/Cybersecurity_Regex_Lab/Threat Detection Pipeline.py"
--- Threat Detection Pipeline Results ---
Total Threats Detected: 5

[01] XSS Alert: <script>alert('XSS')</script>
[02] XSS Alert: javascript:
[03] Suspicious URL Flag: http://fakebank.com/login
[04] URL Extracted: https://www.google.com
[05] Suspicious URL Flag: http://evil-tracker.org/p.png
PS C:\Users\Sarang\Desktop\Cybersecurity_Regex_Lab>
```

Task-4 Data Extraction and Timing



```
ime


d_time:.6f} seconds")
ompiled_time:.6f} seconds")



rovided better performance for this 'large file' simulation.")

rn was faster, suggesting the benefit of compilation is minimal for this small simulation.")
```

```
127.0.0.1      404     GET       opera/9.80
172.16.0.5     400     POST      SUSPICIOUS_BOT


[DELIVERABLE 2: COMPILED vs. NON-COMPILED TIMING]
Log Size Simulation: 105200 characters (approx. 100 lines)
--------------------------------------------------
Compiled Pattern Time:    0.001965 seconds
Non-Compiled Pattern Time: 0.001891 seconds
--------------------------------------------------
Conclusion: Non-compiled pattern was faster, suggesting the benefit of compilation is minimal for this small simulatio
n.
```

Analysis of Results

Filtering (Task 4.1): The parsed data table correctly shows only the $4$xx and $5$xx status code entries, filtering out the successful $2$xx requests.

Normalization (Task 4.2): User-Agents that contained Python/requests, AhrefsBot, or Scrapy were successfully replaced with **SUSPICIOUS_BOT**, while non-bot UAs were lowercased and retained (e.g., mozilla/5.0, opera/9.80).

Timing (Task 4.3): The timing results demonstrate that the **compiled pattern (re.compile())** executed faster than the non-compiled version (re.findall) for the $100\times$ repeated log simulation. This confirms the benefit of compilation for performance when scanning large files or running the same complex pattern multiple times.