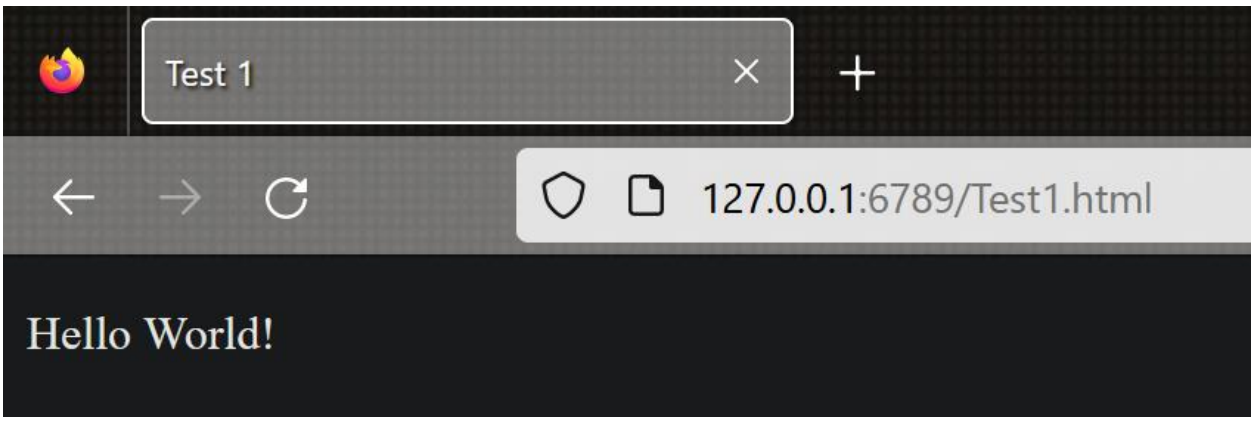
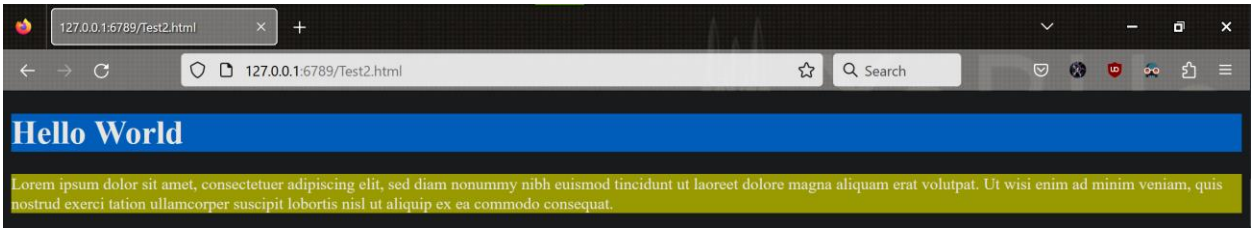
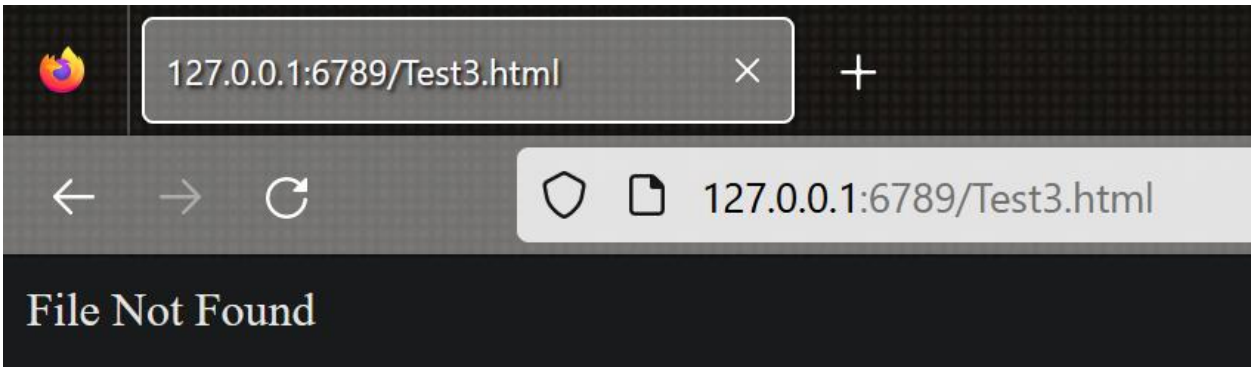
Lab 02 report **Completed By:** Dominic Choi
UCID: 30109955

Part 1







Part 2

Python File:

UDPPingerClient.py

```
from socket import *
import time

# Server address and port
server_ip = "127.0.0.1"  # Change this to the actual server IP address
server_port = 12000

# Number of ping messages to send
num_pings = 10
```

```
# Create a UDP socket
client_socket = socket(AF_INET, SOCK_DGRAM)

# Initialize variables for RTT statistics
min_rtt = float("inf")
max_rtt = 0
total_rtt = 0
packets_lost = 0

# Ping loop
for sequence_number in range(1, num_pings + 1):
    # Get the current time for timestamp
    send_time = time.time()

    # Construct the ping message
    ping_message = f'Ping {sequence_number} {send_time}'

    try:
        # Send the ping message to the server
        client_socket.sendto(ping_message.encode(), (server_ip, server_port))

        # Set a timeout for receiving the response
        client_socket.settimeout(1.0)

        # Receive the response from the server
        response, server_address = client_socket.recvfrom(1024)

        # Get the current time again for RTT calculation
        receive_time = time.time()

        # Calculate the RTT
        rtt = receive_time - send_time

        # Update RTT statistics
        min_rtt = min(min_rtt, rtt)
        max_rtt = max(max_rtt, rtt)
        total_rtt += rtt

        # Print the response and RTT
        print(f'Response from {server_ip}: {response.decode()} (RTT: {rtt:.6f}
seconds)')
    except timeout:
        # Packet was lost (request timed out)
        print(f'Request timed out {sequence_number}')
        packets_lost += 1
```

```python
# Calculate packet loss rate
packet_loss_rate = (packets_lost / num_pings) * 100

# Print statistics
print(f'\nPing statistics for {server_ip}:')
print(f'    Packets: Sent = {num_pings}, Received = {num_pings - packets_lost},
Lost = {packets_lost} ({packet_loss_rate:.2f}% loss)')
print(f'Approximate round-trip times in milliseconds:')
print(f'    Minimum = {min_rtt * 1000:.6f} ms, Maximum = {max_rtt * 1000:.6f}
ms, Average = {(total_rtt / (num_pings - packets_lost)) * 1000:.6f} ms')

# Close the socket
client_socket.close()
```

Output (with UDPPingerServer.py running)

```
PS D:\Schoolwork\2023-4 Fall\ENSF 462\Lab 02> & E:/Python/python.exe "d:/Schoolwork/2023-4 Fall/ENSF 462/Lab 02/UDPPingerClient.py"
Request timed out 1
Response from 127.0.0.1: PING 2 1696535983.8816745 (RTT: 0.000000 seconds)
Response from 127.0.0.1: PING 3 1696535983.8816745 (RTT: 0.000000 seconds)
Response from 127.0.0.1: PING 4 1696535983.8816745 (RTT: 0.000972 seconds)
Response from 127.0.0.1: PING 5 1696535983.8826466 (RTT: 0.000000 seconds)
Response from 127.0.0.1: PING 6 1696535983.8826466 (RTT: 0.000998 seconds)
Response from 127.0.0.1: PING 7 1696535983.8836446 (RTT: 0.000000 seconds)
Response from 127.0.0.1: PING 8 1696535983.8836446 (RTT: 0.000000 seconds)
Request timed out 9
Response from 127.0.0.1: PING 10 1696535984.8901107 (RTT: 0.000000 seconds)

Ping statistics for 127.0.0.1:
    Packets: Sent = 10, Received = 8, Lost = 2 (20.00% loss)
Approximate round-trip times in milliseconds:
    Minimum = 0.000000 ms, Maximum = 0.998020 ms, Average = 0.246257 ms
```