CSE 476 F20 TERM PROJECT FINAL REPORT

ESRA NUR ARICAN

161044028

Lab 1: Web Server Lab

In this lab, we learned the basics of socket programming for TCP connections in Python: how to create a socket, bind it to a specific address and port, as well as send and receive a HTTP packet. We asked to develop a web server that handles one HTTP request at a time. Web server should accept and parse the HTTP request, get the requested file from the server's file system, create an HTTP response message consisting of the requested file preceded by header lines, and then send the response directly to the client. If the requested file is not present in the server, the server should send an HTTP "404 Not Found" message back to the client.

For the Python part, I completed the web server code by filling in the #Fill in start and #Fill in end tags. Later I created an html page called HelloWorld.

HelloWorld.html | webServerLab.py | abc.py

```python
#import socket module
from socket import *

serverPort=6789
serverSocket = socket(AF_INET, SOCK_STREAM)

#Prepare a sever socket
#Fill in start
serverSocket.bind(('',serverPort))
serverSocket.listen(1)

print ("the web server is up on port:",serverPort)
#Fill in end
while True:
    #Establish the connection
    print ('Ready to serve...')
    connectionSocket, addr = serverSocket.accept()

    try:
        message = connectionSocket.recv(1024)
        print (message,'::',message.split()[0],':',message.split()[1])
        filename = message.split()[1]
        print (filename,'||',filename[1:])
        f = open(filename[1:])
        outputdata = f.read()
        print( outputdata)
        #Send one HTTP header line into socket
        #Fill in start
        deneme = "HTTP/1.1 200 OK\r\n\r\n"
        connectionSocket.send(deneme.encode())
        #connectionSocket.send(b"HTTP/1.1 200 OK\r\n\r\n")
        connectionSocket.send(outputdata.encode())
        #Fill in end
        #Send the content of the requested file to the client
```

My code for webServer

```python
        #Send the content of the requested file to the client
        for i in range(0, len(outputdata)):
            connectionSocket.send(outputdata[i].encode())
        connectionSocket.close()
    except IOError:
        #Send response message for file not found
        #Fill in start
        abc = "\nHTTP/1.1 404 Not Found\n\n"
        connectionSocket.send(abc.encode())
```

webServer code contd.

```html
<!DOCTYPE html>
<html>
    <head>
        <title>HELLO WORLD!</title>
    </head>
    <body>
        <p>  HELLO WORLD!
             SERVER IS RUNNNING!</p>
    </body>
</html>
```

Basic html code

To see the results, I run my python code with python 3 and python 2.7 after putting the html and python files in the same directory. Since the Skeleton code is written in python 2, I got some errors when I compiled my code with python3. I corrected these errors later by compiling with python 2.7, and when I made some changes in my code, I was able to run it with python3.

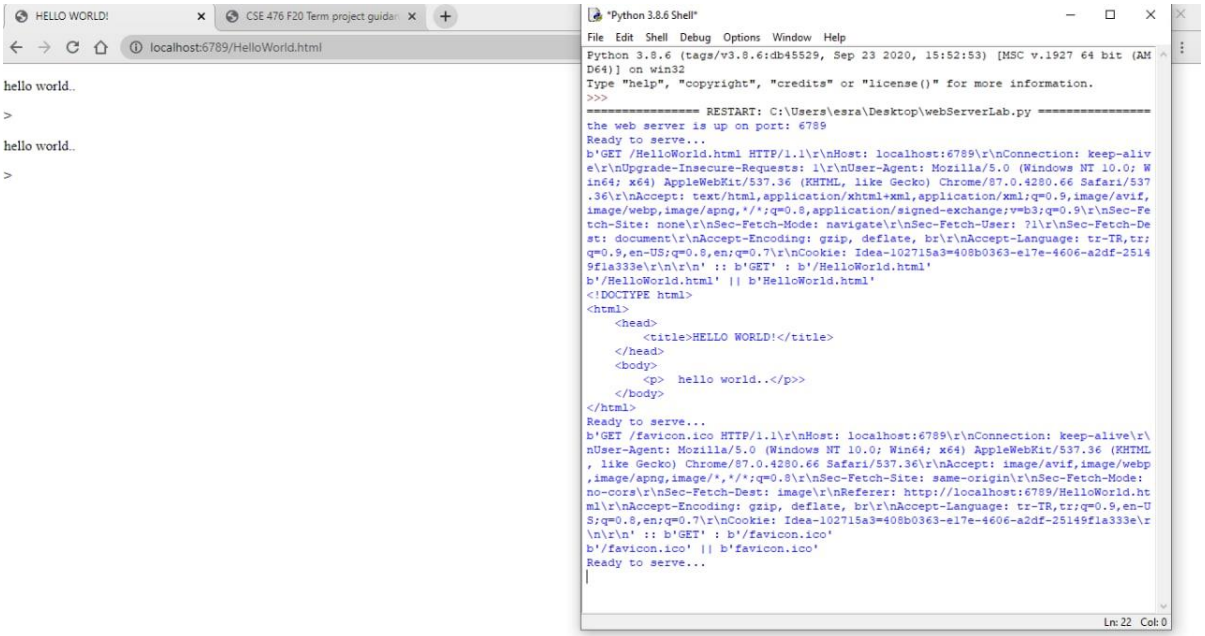
After successfully run the code I got server ready message.

```
Python 3.8.6 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:\Users\esra\Desktop\webServerLab.py ==============
the web server is up on port: 6789
Ready to serve...
```

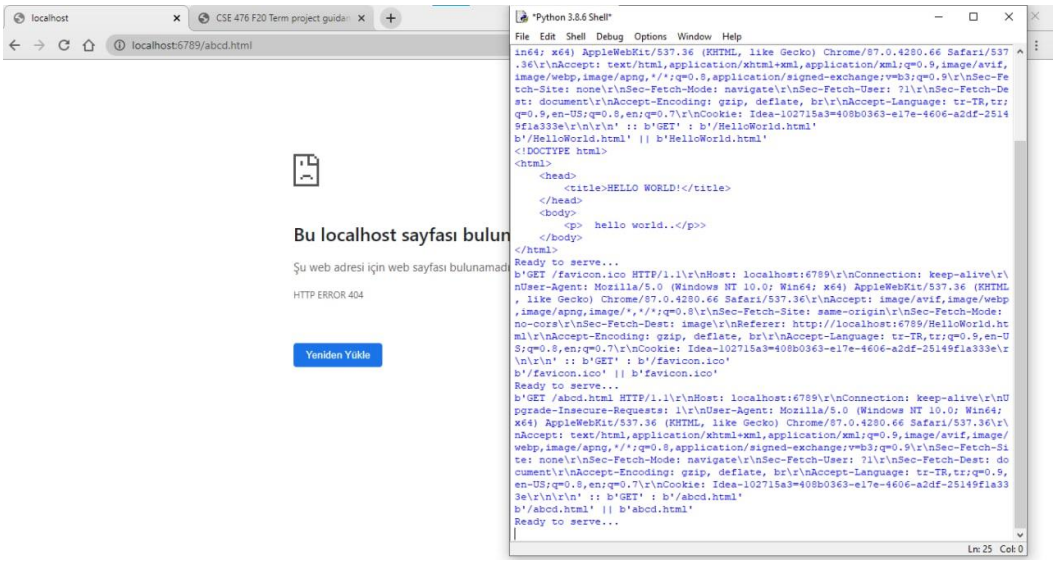
After this message I opened a Google chrome tab and write the html file name in a way that:

localhost:6789/HelloWorld.html

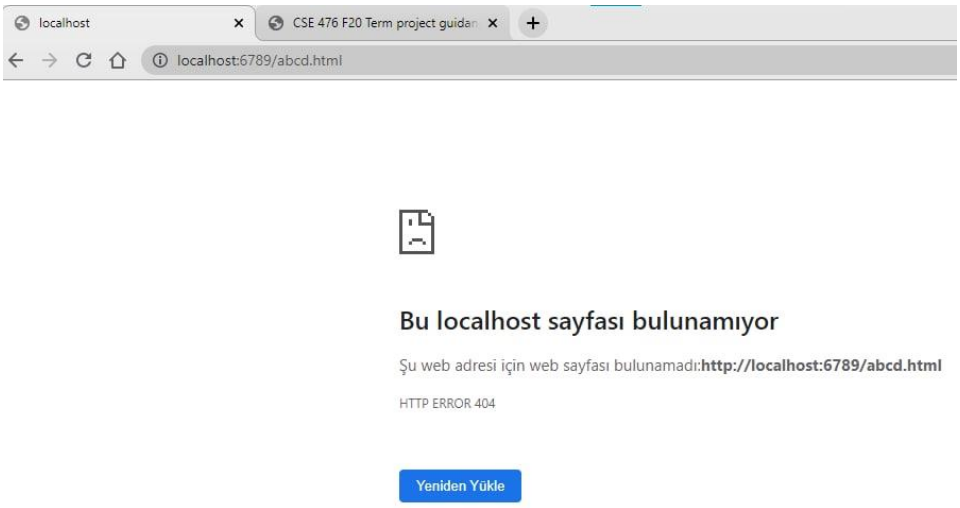
TEST RESULTS – LAB 1



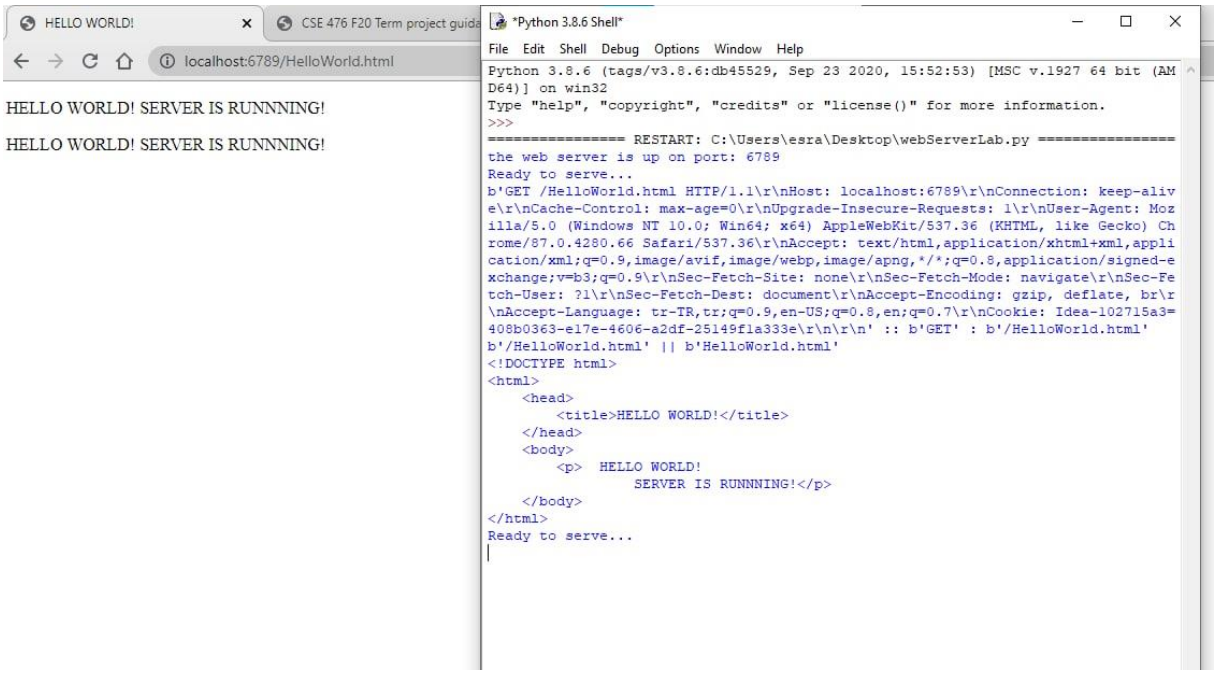
Result of running server on Windows machine, with correct html filename



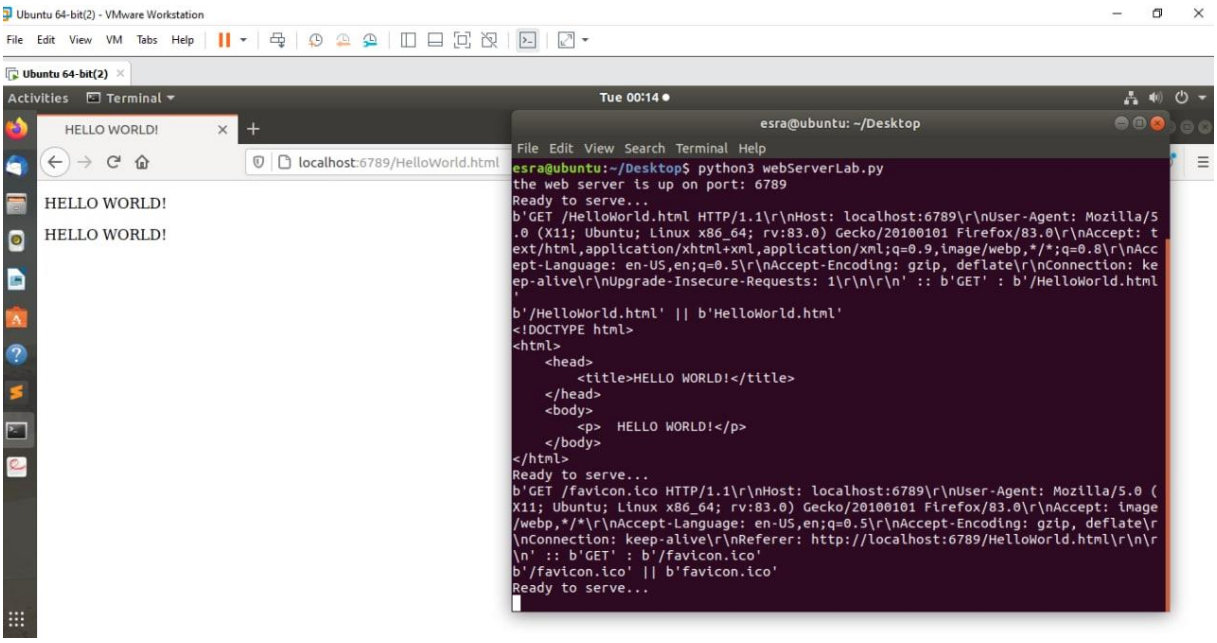
Result of running server with wrong html file named "abcd.html" which does not exists.



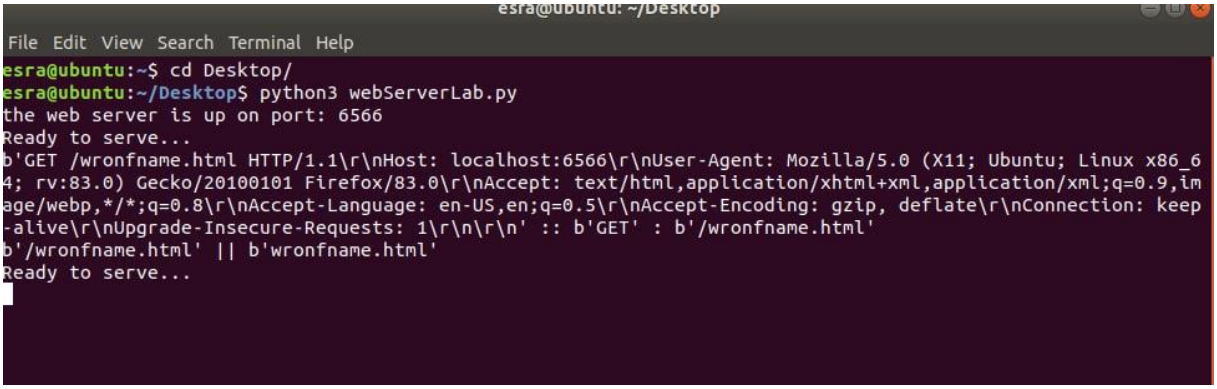
404 error given for wrong html file name



Another result of Windows machine with same HelloWorld.html file, I just changed the contents of the file.



Result of the webServer.py file from ubuntu machine



Result of webServer.py file with non-existing html file, from ubuntu machine

Lab 2: UDP Pinger Lab

For this lab assignment, I take the given server code and implement asked client code on python.

I send the ping message using UDP. Then print the response message from server if exists. Then I calculated time by substracting begin and end time. Finally I printed “Time out!” message if server doesn’t response.

TEST RESULTS – LAB 2

```
Komut İstemi

C:\Users\esra\Desktop>py -2 UDP_client.py
Sent Ping number: 1 Sat Jan  2 14:56:27 2021
Received PING NUMBER: 1 SAT JAN  2 14:56:27 2021
RTT: 0.00800013542175 seconds

Sent Ping number: 2 Sat Jan  2 14:56:27 2021
Number: 2  Time out!

Sent Ping number: 3 Sat Jan  2 14:56:28 2021
Number: 3  Time out!

Sent Ping number: 4 Sat Jan  2 14:56:29 2021
Received PING NUMBER: 4 SAT JAN  2 14:56:29 2021
RTT: 0.00299978256226 seconds

Sent Ping number: 5 Sat Jan  2 14:56:29 2021
Received PING NUMBER: 5 SAT JAN  2 14:56:29 2021
RTT: 0.000999927520752 seconds

Sent Ping number: 6 Sat Jan  2 14:56:29 2021
Received PING NUMBER: 6 SAT JAN  2 14:56:29 2021
RTT: 0.000999927520752 seconds

Sent Ping number: 7 Sat Jan  2 14:56:29 2021
Received PING NUMBER: 7 SAT JAN  2 14:56:29 2021
RTT: 0.00100016593933 seconds

Sent Ping number: 8 Sat Jan  2 14:56:29 2021
Received PING NUMBER: 8 SAT JAN  2 14:56:29 2021
RTT: 0.000999927520752 seconds

Sent Ping number: 9 Sat Jan  2 14:56:29 2021
Received PING NUMBER: 9 SAT JAN  2 14:56:29 2021
RTT: 0.000999927520752 seconds

Sent Ping number: 10 Sat Jan  2 14:56:29 2021
Received PING NUMBER: 10 SAT JAN  2 14:56:29 2021
RTT: 0.000999927520752 seconds

closing socket

C:\Users\esra\Desktop>_
```

```
Komut İstemi - py -2 UDPPingerServer.py

Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\esra>cd Desktop

C:\Users\esra\Desktop>py -2 UDPPingerServer.py
Started UDP server on port 12000
```

Result of running UDP_client.py file and UDPPingerServer.py file on Windows machine

```
Komut İstemi

closing socket

C:\Users\esra\Desktop>py -2 UDP_client.py
Sent Ping number: 1 Sat Jan  2 14:58:38 2021
Received PING NUMBER: 1 SAT JAN  2 14:58:38 2021
RTT: 0.0130000114441 seconds

Sent Ping number: 2 Sat Jan  2 14:58:38 2021
Received PING NUMBER: 2 SAT JAN  2 14:58:38 2021
RTT: 0.0139999389648 seconds

Sent Ping number: 3 Sat Jan  2 14:58:38 2021
Received PING NUMBER: 3 SAT JAN  2 14:58:38 2021
RTT: 0.0019998550415 seconds

Sent Ping number: 4 Sat Jan  2 14:58:38 2021
Number: 4  Time out!

Sent Ping number: 5 Sat Jan  2 14:58:39 2021
Number: 5  Time out!

Sent Ping number: 6 Sat Jan  2 14:58:40 2021
Number: 6  Time out!

Sent Ping number: 7 Sat Jan  2 14:58:41 2021
Number: 7  Time out!

Sent Ping number: 8 Sat Jan  2 14:58:42 2021
Received PING NUMBER: 8 SAT JAN  2 14:58:42 2021
RTT: 0.0019998550415 seconds

Sent Ping number: 9 Sat Jan  2 14:58:42 2021
Received PING NUMBER: 9 SAT JAN  2 14:58:42 2021
RTT: 0.00300002098083 seconds

Sent Ping number: 10 Sat Jan  2 14:58:42 2021
Received PING NUMBER: 10 SAT JAN  2 14:58:42 2021
RTT: 0.00399994850159 seconds

closing socket

C:\Users\esra\Desktop>_
```

```
Komut İstemi - py -2 UDPPingerServer.py

Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\esra>cd Desktop

C:\Users\esra\Desktop>py -2 UDPPingerServer.py
Started UDP server on port 12000
```

Result of running UDP_client.py file and UDPPingerServer.py file on Windows machine

```
esra@ubuntu: ~/Desktop/cse476-term-project-codes
File  Edit  View  Search  Terminal  Help

closing socket
esra@ubuntu:~/Desktop/cse476-term-project-codes$ python UDP_client.py
Sent Ping number: 1 Sat Jan  2 16:04:00 2021
Number1 Time out!

Sent Ping number: 2 Sat Jan  2 16:04:01 2021
Received PING NUMBER: 2 SAT JAN  2 16:04:01 2021
RTT: 0.000322103500366 seconds

Sent Ping number: 3 Sat Jan  2 16:04:01 2021
Received PING NUMBER: 3 SAT JAN  2 16:04:01 2021
RTT: 0.000262022018433 seconds

Sent Ping number: 4 Sat Jan  2 16:04:01 2021
Number4 Time out!

Sent Ping number: 5 Sat Jan  2 16:04:02 2021
Number5 Time out!

Sent Ping number: 6 Sat Jan  2 16:04:03 2021
Number6 Time out!

Sent Ping number: 7 Sat Jan  2 16:04:04 2021
Number7 Time out!

Sent Ping number: 8 Sat Jan  2 16:04:05 2021
Received PING NUMBER: 8 SAT JAN  2 16:04:05 2021
RTT: 0.00069785118103 seconds

Sent Ping number: 9 Sat Jan  2 16:04:05 2021
Received PING NUMBER: 9 SAT JAN  2 16:04:05 2021
RTT: 0.000239133834839 seconds

Sent Ping number: 10 Sat Jan  2 16:04:05 2021
Received PING NUMBER: 10 SAT JAN  2 16:04:05 2021
```

```
esra@ubuntu: ~/Desktop/cse476-term-project-codes
File  Edit  View  Search  Terminal  Help
esra@ubuntu:~$ cd Desktop/cse476-term-project-codes/
esra@ubuntu:~/Desktop/cse476-term-project-codes$ python  UDPPingerServer.py
Started UDP server on port 12000
```

Result from running codes on ubuntu machine

Lab 3: SMTP Lab

In this lab we develop a simple mail client that sends email to any recipient.

To be able to use gmail address, I add following lines to given skeleton code;

```
mailserver = ('smtp.gmail.com', 587) #Fill in start #Fill in end
```

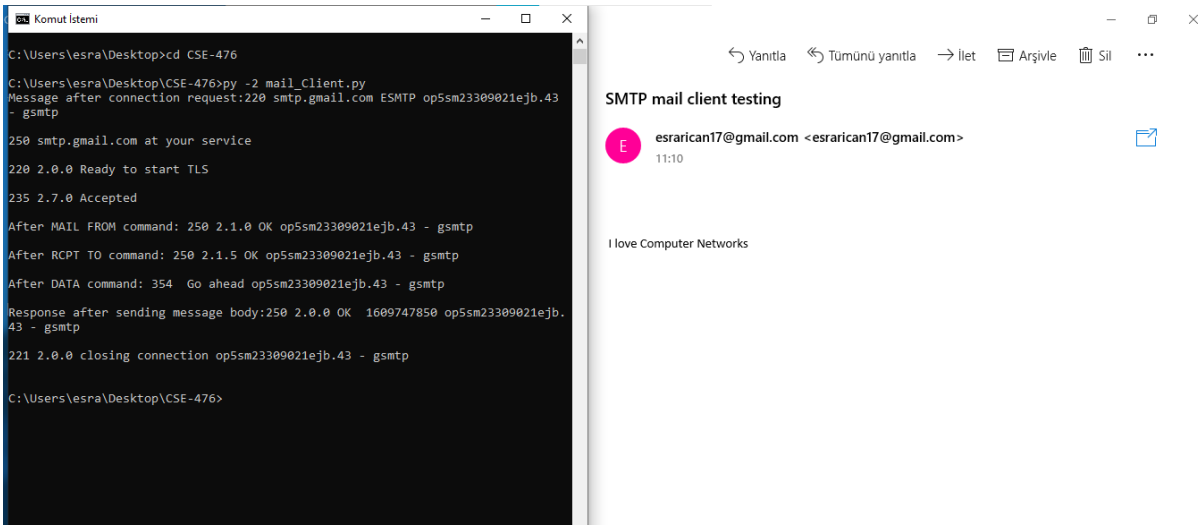
Then I add this lines of codes after HELO command

```
# Sending STARTTLS command
command="STARTTLS\r\n"
clientSocket.send(command.encode())
recv1 = clientSocket.recv(1024)
print(recv1)
if recv1[:3] != '250':
    print('250 reply not received from server.')
```

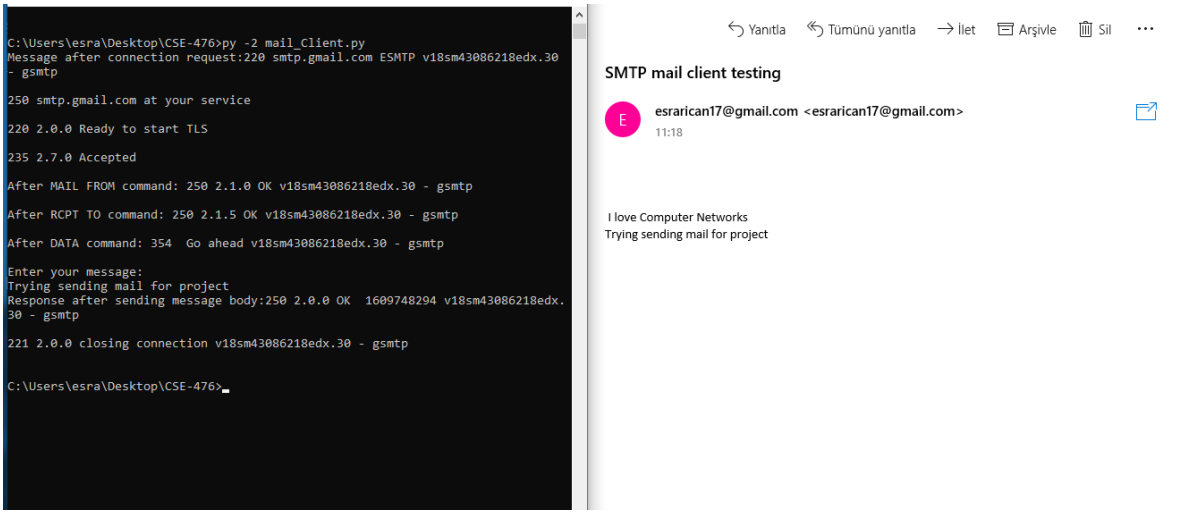
NOTE: Before submit my code, I deleted my password from the file

```
# Info for username and password
username =  "esrarican17@gmail.com"                    #the username for your server
password = "***somepassword***"                        #1 changed the password after tests, for submission
```

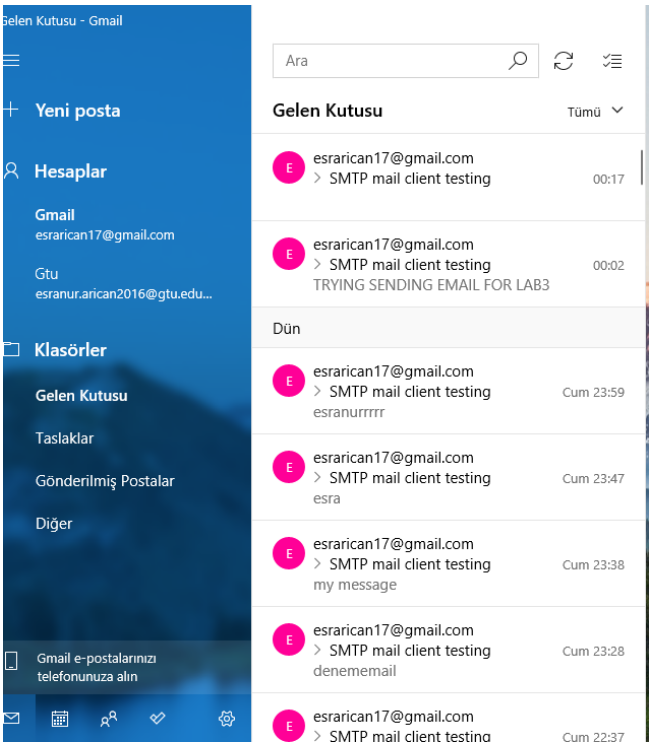
TEST RESULTS – LAB 3



Sending mail to my gtu address from my gmail address



I also tried to send both given message and another message entered by user from terminal



I send the message entered from terminal as an e-mail for trial purposes.

I took the part of receiving input from terminal to comment for submission