

FINAL PROJECT

DATA COMMUNICATIONS NETWORKS

Prof. Dr. Noha ElKorany - Prof. Dr. Karim Banawan
Communication & Electronics Department

Asmaa Gamal Abdel-Halem Mabrouk Nagy

أسماء جمال عبد الحليم مبروك ناجي

15010473 - section 8

“Socket Programming” Assignment

Part 1:

1. Install and compile the Python programs:

Code

1. UDP client:

```
udp_client.py X  udp_server.py  tcp_client.py  tcp_server.py
udp_client.py > ...
1  from socket import *
2  serverName = "127.0.0.1"
3  serverPort = 12000
4  clientSocket = socket(AF_INET, SOCK_DGRAM)
5  message = input("Input lowercase sentence:")
6  clientSocket.sendto(message.encode(),(serverName, serverPort))
7  modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
8  print (modifiedMessage.decode())
9  clientSocket.close()
```

2. UDP server:

```
udp_client.py  udp_server.py X  tcp_client.py  tcp_server.py
udp_server.py > ...
1  from socket import *
2  serverPort = 12000
3  serverSocket = socket(AF_INET, SOCK_DGRAM)
4  serverSocket.bind(('', serverPort))
5  print ("The server is ready to receive")
6  while True:
7      message, clientAddress = serverSocket.recvfrom(2048)
8      modifiedMessage = message.decode().upper()
9      serverSocket.sendto(modifiedMessage.encode(), clientAddress)
```

3. TCP client:

```
udp_client.py  udp_server.py  tcp_client.py X  tcp_server.py

tcp_client.py > ...
1  from socket import *
2  serverName = '127.0.0.1'
3  serverPort = 12000
4  clientSocket = socket(AF_INET, SOCK_STREAM)
5  clientSocket.connect((serverName, serverPort))
6  sentence = input('Input lowercase sentence:')
7  clientSocket.send(sentence.encode())
8  modifiedSentence = clientSocket.recv(1024)
9  print ('From Server:', modifiedSentence.decode())
10 clientSocket.close()
```

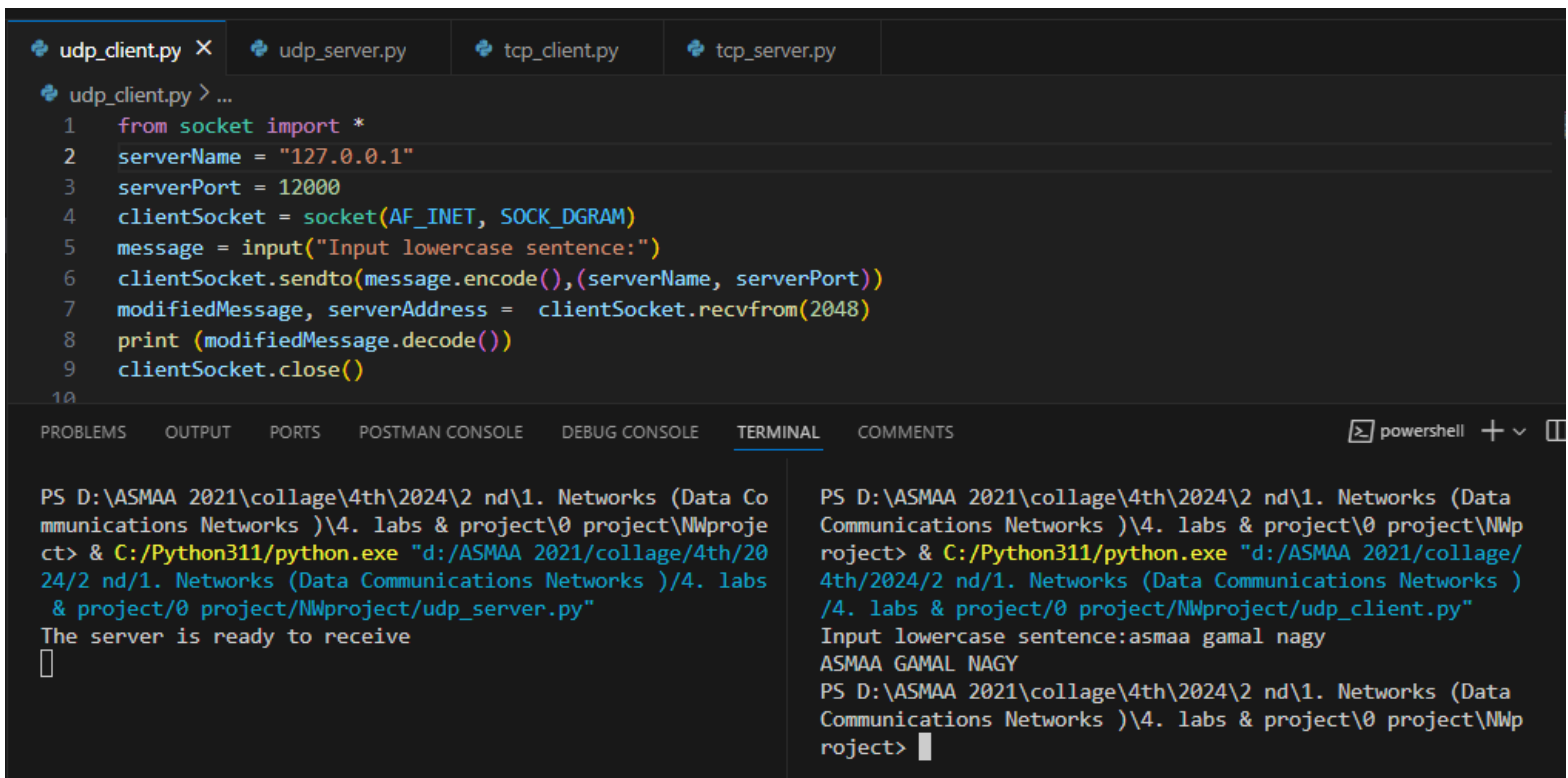
4. TCP server:

```
udp_client.py  udp_server.py  tcp_client.py  tcp_server.py X

tcp_server.py > ...
1  from socket import *
2  serverPort = 12000
3  serverSocket = socket(AF_INET, SOCK_STREAM)
4  serverSocket.bind(('', serverPort))
5  serverSocket.listen(1)
6  print('The server is ready to receive')
7  while True:
8      connectionSocket, addr = serverSocket.accept()
9      sentence = connectionSocket.recv(1024).decode()
10     capitalizedSentence = sentence.upper()
11     connectionSocket.send(capitalizedSentence.encode())
12     connectionSocket.close()
```

Results after “compile and run”

1. UDP:



The screenshot shows a Visual Studio Code editor with four tabs: `udp_client.py`, `udp_server.py`, `tcp_client.py`, and `tcp_server.py`. The `udp_client.py` tab is active, displaying the following Python code:

```
1 from socket import *
2 serverName = "127.0.0.1"
3 serverPort = 12000
4 clientSocket = socket(AF_INET, SOCK_DGRAM)
5 message = input("Input lowercase sentence:")
6 clientSocket.sendto(message.encode(), (serverName, serverPort))
7 modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
8 print (modifiedMessage.decode())
9 clientSocket.close()
10
```

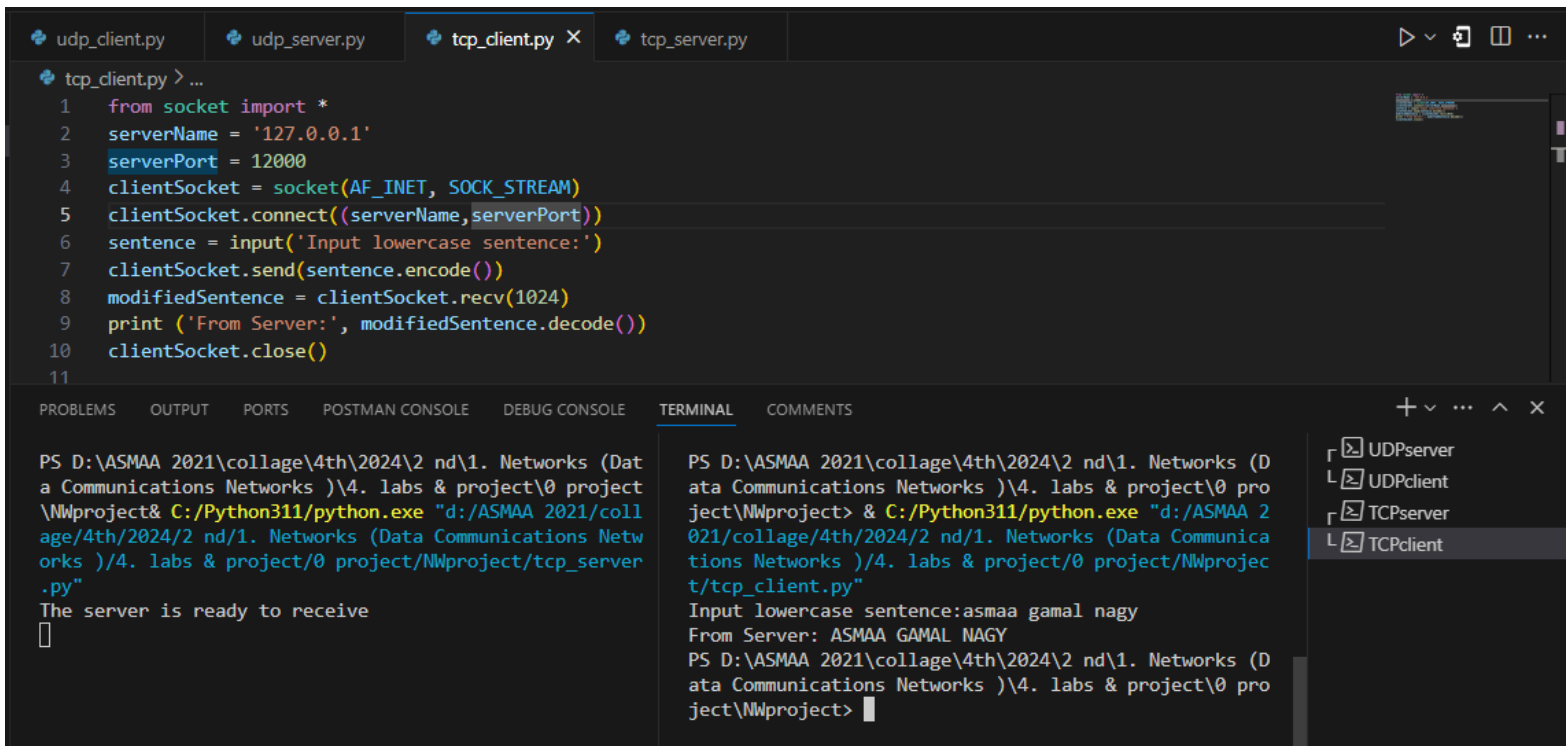
Below the code editor, the **TERMINAL** tab is active, showing the execution of the programs in a PowerShell environment. The left pane shows the server output:

```
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks )\4. labs & project\0 project\NWproject> & C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks )/4. labs & project/0 project/NWproject/udp_server.py"
The server is ready to receive
█
```

The right pane shows the client output:

```
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks )\4. labs & project\0 project\NWproject> & C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks )/4. labs & project/0 project/NWproject/udp_client.py"
Input lowercase sentence:asmaa gamal nagy
ASMAA GAMAL NAGY
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks )\4. labs & project\0 project\NWproject> █
```

2. TCP:



The screenshot shows a Visual Studio Code editor with four tabs: `udp_client.py`, `udp_server.py`, `tcp_client.py`, and `tcp_server.py`. The `tcp_client.py` tab is active, displaying the following Python code:

```
1 from socket import *
2 serverName = '127.0.0.1'
3 serverPort = 12000
4 clientSocket = socket(AF_INET, SOCK_STREAM)
5 clientSocket.connect((serverName, serverPort))
6 sentence = input('Input lowercase sentence:')
7 clientSocket.send(sentence.encode())
8 modifiedSentence = clientSocket.recv(1024)
9 print ('From Server:', modifiedSentence.decode())
10 clientSocket.close()
11
```

Below the code editor, the **TERMINAL** tab is active, showing the execution of the programs. The left pane shows the server output:

```
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks )\4. labs & project\0 project\NWproject> & C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks )/4. labs & project/0 project/NWproject/tcp_server.py"
The server is ready to receive
█
```

The right pane shows the client output:

```
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks )\4. labs & project\0 project\NWproject> & C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks )/4. labs & project/0 project/NWproject/tcp_client.py"
Input lowercase sentence:asmaa gamal nagy
From Server: ASMAA GAMAL NAGY
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks )\4. labs & project\0 project\NWproject> █
```

On the right side of the terminal, a file explorer shows the project structure:

- UDPServer
- UDPclient
- TCPserver
- TCPclient

a. Suppose you run TCPClient before you run TCPServer. What happens? Why?

It gives an error:

ConnectionRefusedError: [WinError 10061] No connection could be made because the target machine actively refused it

Because:

No TCP connection is established yet, because the server is not running so that the client can not start the connection setup.

Screenshot for what happens:

The screenshot shows a Python IDE with four tabs: `udp_client.py`, `udp_server.py`, `tcp_client.py` (active), and `tcp_server.py`. The `tcp_client.py` tab displays the following code:

```
1 from socket import *
2 serverName = '127.0.0.1'
3 serverPort = 12000
4 clientSocket = socket(AF_INET, SOCK_STREAM)
5 clientSocket.connect((serverName, serverPort))
6 sentence = input('Input lowercase sentence:')
7 clientSocket.send(sentence.encode())
8 modifiedSentence = clientSocket.recv(1024)
9 print('From Server:', modifiedSentence.decode())
10 clientSocket.close()
```

Below the code editor, the `TERMINAL` tab shows the command prompt output:

```
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject>
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject> C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject/tcp_client.py"
Traceback (most recent call last):
  File "d:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject\tcp_client.py", line 5, in <module>
    clientSocket.connect((serverName, serverPort))
ConnectionRefusedError: [WinError 10061] No connection could be made because the target machine actively refused it
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject>
```

On the right side of the terminal, a file explorer shows the project structure with files: `UDPServer`, `UDPClient`, `TCPserver`, and `TCPclient`.

b. Suppose you run UDPClient before you run UDPServer. What happens? Why?

The UDP client runs, and asks you to enter your input lowercase message, but it gives the user a connection error when the client try to send your message to the remote closed unactive server

Because:

UDP doesn't need to establish a connection unlike the TCP, but when it comes to receive the message reply, it will brust an error, after this code line:

```
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
```

Screenshot for what happens:

The screenshot shows a Python IDE with four tabs: `udp_client.py`, `udp_server.py`, `tcp_client.py`, and `tcp_server.py`. The `udp_client.py` tab is active, showing the following code:

```
1 from socket import *
2 serverName = "127.0.0.1"
3 serverPort = 12000
4 clientSocket = socket(AF_INET, SOCK_DGRAM)
5 message = input("Input lowercase sentence:")
6 clientSocket.sendto(message.encode(), (serverName, serverPort))
7 modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
8 print (modifiedMessage.decode())
9 clientSocket.close()
```

The terminal window shows the execution of the program. It displays the command prompt, the input "asmaa gamal nagy", and a traceback error message:

```
ata Communications Networks )\4. labs & project\0 pro
ject\NWpro& C:/Python311/python.exe "d:/ASMAA 2021/co
llage/4th/2024/2 nd/1. Networks (Data Communications
Networks )/4. labs & project\0 project\NWproject/udp_
client.py"
Input lowercase sentence:asmaa gamal nagy
Traceback (most recent call last):
  File "d:\ASMAA 2021\collage\4th\2024\2 nd\1. Networ
ks (Data Communications Networks )\4. labs & project\
0 project\NWproject\udp_client.py", line 7, in <modul
e>
    modifiedMessage, serverAddress = clientSocket.re
cvfrom(2048)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ConnectionResetError: [WinError 10054] An existing co
nnection was forcibly closed by the remote host
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (D
ata Communications Networks )\4. labs & project\0 pro
ject\NWproject>
```

c. What happens if you use different port numbers for the client and server sides?

UDP:

After changing the UDP server port number to a different port number (4000) and keep the UDP client working at the same old port num(12000), the UDP client asks the user to enter an input message then gives us the same **connection error** us before, so your packets are lost.

Screenshot for what happens:

```
udp_client.py | udp_server.py X | tcp_client.py | tcp_server.py
udp_server.py > ...
1 from socket import *
2 serverPort = 4000
3 serverSocket = socket(AF_INET, SOCK_DGRAM)
4 serverSocket.bind(('', serverPort))
5 print("The server is ready to receive")
6 while True:
7     message, clientAddress = serverSocket.recvfrom(2048)
8     modifiedMessage = message.decode().upper()
9     serverSocket.sendto(modifiedMessage.encode(), clientAddress)
10
```

PROBLEMS OUTPUT PORTS POSTMAN CONSOLE DEBUG CONSOLE TERMINAL COMMENTS

PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject> & C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject/udp_server.py"

The server is ready to receive

PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject> & C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject/udp_client.py"

Input lowercase sentence:asmaa

Traceback (most recent call last):

File "d:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject\udp_client.py", line 7, in <module>

modifiedMessage, serverAddress = clientSocket.recvfrom(2048)

ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host

Ln 2, Col 15 Spaces: 4 UTF-8 CRLF Python 3.11.4.6

TCP:

After changing the TCP server port number to a different port number (5000) and keep the UDP client working at the same old port num(12000), the TCP client will give an error directly because **No Connection** has established as the server port num is now different from the port num that the client is now trying to reach out, so they can't talk to each other now.

The error statement:

```
clientSocket.connect((serverName,serverPort))
```

ConnectionRefusedError: [WinError 10061] No connection could be made because the target machine actively refused it

Screenshot for what happens:

```
udp_client.py | udp_server.py | tcp_client.py | tcp_server.py X
tcp_server.py > ...
1 from socket import *
2 serverPort = 5000
3 serverSocket = socket(AF_INET, SOCK_STREAM)
4 serverSocket.bind(('', serverPort))
5 serverSocket.listen(1)
6 print('The server is ready to receive')
7 while True:
8     connectionSocket, addr = serverSocket.accept()
9     sentence = connectionSocket.recv(1024).decode()
10    capitalizedSentence = sentence.upper()
11    connectionSocket.send(capitalizedSentence.encode())
```

PROBLEMS OUTPUT PORTS POSTMAN CONSOLE DEBUG CONSOLE TERMINAL COMMENTS

PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject> & C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject/tcp_server.py"

The server is ready to receive

PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject> & C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject/tcp_client.py"

Traceback (most recent call last):

File "d:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks)\4. labs & project\0 project\NWproject\tcp_client.py", line 5, in <module>

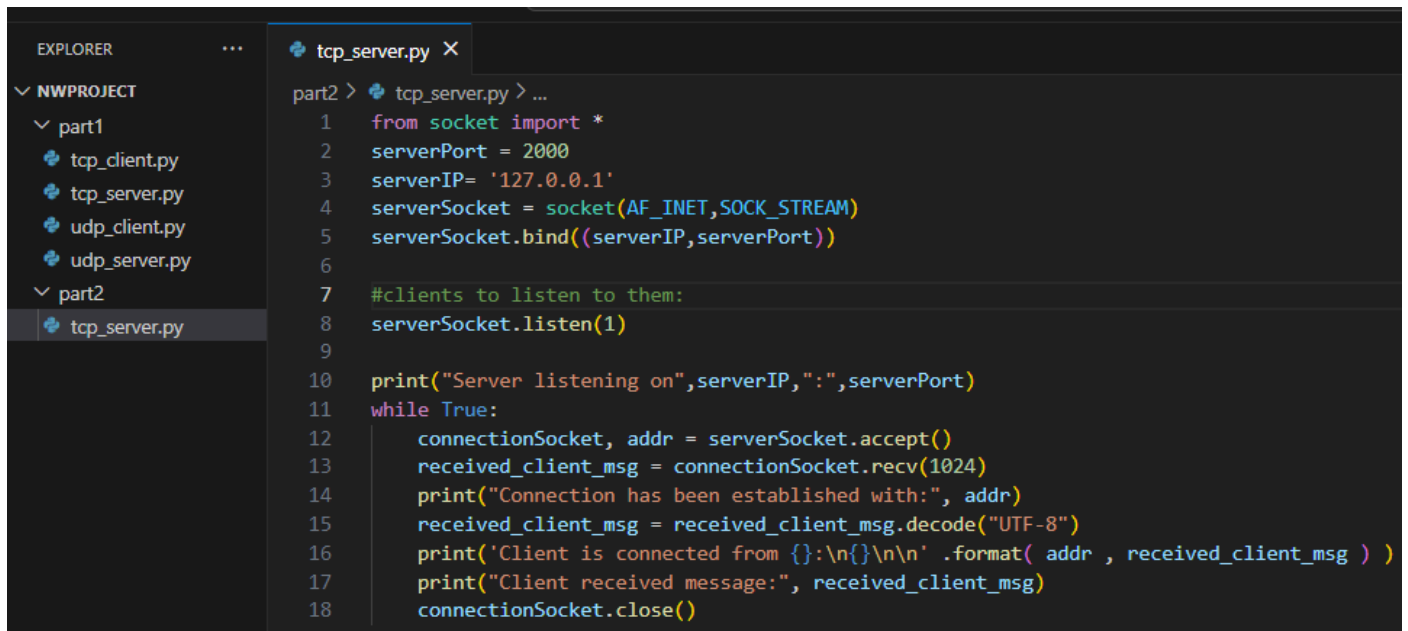
clientSocket.connect((serverName,serverPort))

ConnectionRefusedError: [WinError 10061] No connection could be made because the target machine actively refused it

Part 2:

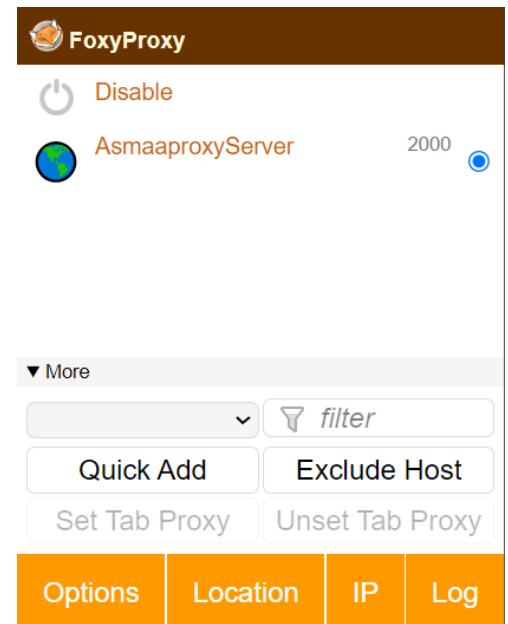
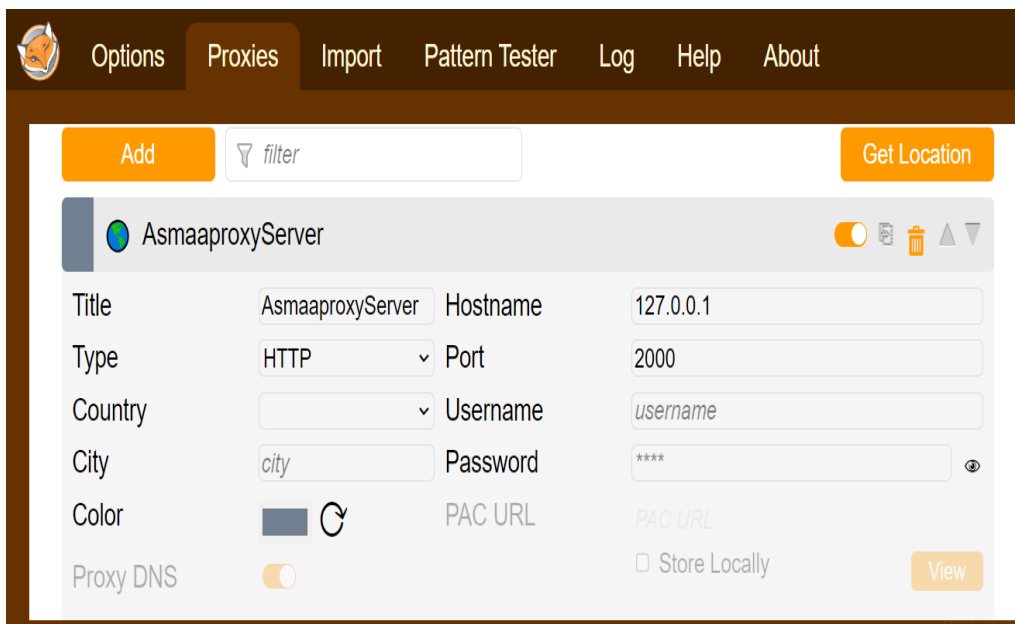
1. Write a simple TCP program for a server that accepts lines of input from a client and prints the lines onto the server's standard output.

This is the TCP program code:



```
part2 > tcp_server.py > ...
1  from socket import *
2  serverPort = 2000
3  serverIP= '127.0.0.1'
4  serverSocket = socket(AF_INET,SOCK_STREAM)
5  serverSocket.bind((serverIP,serverPort))
6
7  #clients to listen to them:
8  serverSocket.listen(1)
9
10 print("Server listening on",serverIP,":",serverPort)
11 while True:
12     connectionSocket, addr = serverSocket.accept()
13     received_client_msg = connectionSocket.recv(1024)
14     print("Connection has been established with:", addr)
15     received_client_msg = received_client_msg.decode("UTF-8")
16     print('Client is connected from {}:{}\n\n'.format( addr , received_client_msg ))
17     print("Client received message:", received_client_msg)
18     connectionSocket.close()
```

2. Compile and execute your program. On Web browser, set the proxy server in the browser to the host that is running your server program; also configure the port number appropriately.



3. Your browser should now send its GET request messages to your server, and your server should display the messages on its standard output. Use this platform to determine whether your browser generates conditional GET messages for objects that are locally cached.

Yes, I got some conditional GET messages as shown bellow:

```
PS D:\ASMAA 2021\collage\4th\2024\2 nd\1. Networks (Data Communications Networks )\4. labs & project\0 project\NWproject> & C:/Python311/python.exe "d:/ASMAA 2021/collage/4th/2024/2 nd/1. Networks (Data Communications Networks )/4. labs & project/0 project/NWproject/part2/tcp_server.py"
Server listening on 127.0.0.1 : 2000
Connection has been established with: ('127.0.0.1', 3337)
Client is connected from ('127.0.0.1', 3337):
GET http://vulnweb.com/ HTTP/1.1
Host: vulnweb.com
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: SL_G_WPT_T0=ar; SL_GWPT_Show_Hide_tmp=1; SL_wptGlobTipTmp=1
If-None-Match: W/"5f1fedf1-fb2"
If-Modified-Since: Tue, 28 Jul 2020 09:20:49 GMT
```

```
Client received message: GET http://vulnweb.com/ HTTP/1.1
Host: vulnweb.com
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: SL_G_WPT_T0=ar; SL_GWPT_Show_Hide_tmp=1; SL_wptGlobTipTmp=1
If-None-Match: W/"5f1fedf1-fb2"
If-Modified-Since: Tue, 28 Jul 2020 09:20:49 GMT
```

Idea Summary of Part2:

