



Faculty of Engineering & Technology
Electrical & Computer Engineering Department
ENCS3320 - Computer Networks
Project #1 Report

Prepared by:

Mohammad Makhamreh	1200227
Ahmad Zubaidea	1200105
Islam Jihad	1191375

Instructor: Dr. Abdalkarim Awad

Section: 3

Date: 23/12/2022

Part1:

1.1 - In your own words, what are ping, tracert, nslookup (write one sentence for each one)

- **Ping :** a command to check if the address exists and if we can request it. And check if we connected to the network by getting response from our router.
- **Tracert:** it gives more details than “ping” , so we use it to see the path of request messages among routers that between the sender and receiver.
- **Nslookup :** a command that we use it to query DNS, it takes the name and return the address of the website.

1.2 - Make sure that your computer is connected to the internet and then run the following commands:

**1.1.2 Ping a device in the same network, e.g. from a laptop to a smartphone.
We found that the phone ip is “ 192.168.1.109”**



```
C:\Users\HP>ping 192.168.1.109

Pinging 192.168.1.109 with 32 bytes of data:
Reply from 192.168.1.109: bytes=32 time=92ms TTL=64
Reply from 192.168.1.109: bytes=32 time=205ms TTL=64
Reply from 192.168.1.109: bytes=32 time=3ms TTL=64
Reply from 192.168.1.109: bytes=32 time=109ms TTL=64

Ping statistics for 192.168.1.109:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 205ms, Average = 102ms

C:\Users\HP>
```

When we disconnect the internet on the phone the result was request timed out.

```
C:\Users\HP>ping 192.168.1.109

Pinging 192.168.1.109 with 32 bytes of data:
Request timed out.
Request timed out.
```

- It sent 4 packets and received them all, and it takes 3ms.

1.2.2 - ping www.yale.edu

```
C:\Users\HP>ping www.yale.edu

Pinging pantheon-systems.map.fastly.net [199.232.82.133] with 32 bytes of data:
Reply from 199.232.82.133: bytes=32 time=98ms TTL=51
Reply from 199.232.82.133: bytes=32 time=88ms TTL=51
Reply from 199.232.82.133: bytes=32 time=88ms TTL=51
Reply from 199.232.82.133: bytes=32 time=88ms TTL=51

Ping statistics for 199.232.82.133:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 88ms, Maximum = 98ms, Average = 90ms

C:\Users\HP>
```

Activate Windows
Go to Settings to activate Windows.

6:49 PM
12/23/2022

The minimum round trip times was 88ms greater than the phone.

1.2.3 - tracert www.yale.edu

```
C:\Users\HP>tracert www.yale.edu

Tracing route to pantheon-systems.map.fastly.net [199.232.82.133]
over a maximum of 30 hops:

  0  1 ms    1 ms    1 ms   192.168.1.1
  1  21 ms   20 ms   21 ms   10.74.96.231
  2  66 ms   64 ms   65 ms   10.74.23.77
  3  64 ms   63 ms   63 ms   10.74.16.49
  4  62 ms   63 ms   62 ms   10.74.19.134
  5  65 ms   66 ms   66 ms   10.74.59.190
  6  *        *        *       Request timed out.
  7  88 ms   89 ms   88 ms   199.232.82.133

Trace complete.

C:\Users\HP>
```

Activate Windows
Go to Settings to activate Windows.

6:52 PM
12/23/2022

1.2.4 - nslookup www.yale.edu

```
C:\Users\HP>nslookup www.yale.edu
Server:  UnKnown
Address:  192.168.1.1
```

```
Non-authoritative answer:
Name:    pantheon-systems.map.fastly.net
Address: 199.232.82.133
Aliases: www.yale.edu
```

Activate Windows

Go to Settings to activate Windows.

```
C:\Users\HP>
```

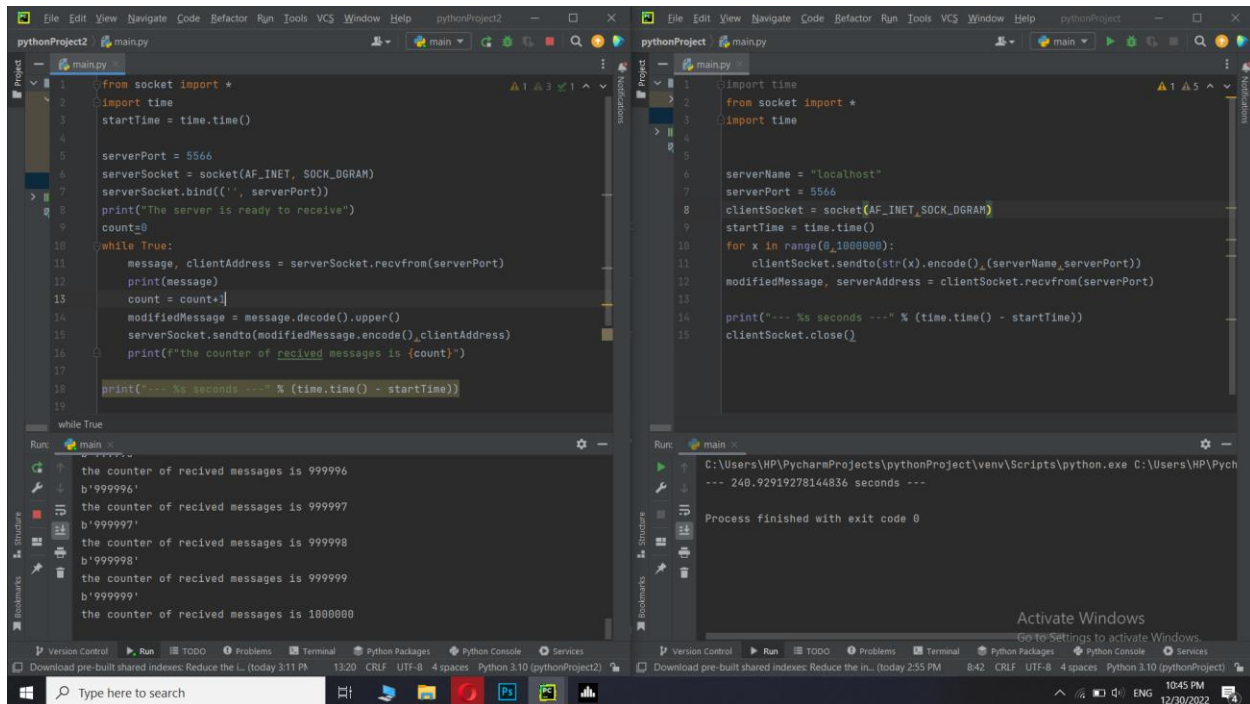
6:54 PM
12/23/2022

Part 2.

Implement the following server and client application both for TCP and for UDP: A client continuously sends the numbers from 0 to 1000,000 to a server listening on port 5566. The server counts the received messages. Run the programs

2.1- on same computer

First, The UDP by Python Socket Programming and we calculate the time and the number of packets received.



```
1 from socket import *
2 import time
3 startTime = time.time()
4
5 serverPort = 5566
6 serverSocket = socket(AF_INET, SOCK_DGRAM)
7 serverSocket.bind(('', serverPort))
8 print("The server is ready to receive")
9 count=0
10 while True:
11     message, clientAddress = serverSocket.recvfrom(serverPort)
12     print(message)
13     count = count+1
14     modifiedMessage = message.decode().upper()
15     serverSocket.sendto(modifiedMessage.encode(), clientAddress)
16     print(f"the counter of received messages is {count}")
17
18 print("---- %s seconds ----" % (time.time() - startTime))
19
20 while True:
```

```
1 import time
2 from socket import *
3 import time
4
5 serverName = "localhost"
6 serverPort = 5566
7 clientSocket = socket(AF_INET, SOCK_DGRAM)
8 startTime = time.time()
9 for x in range(0,1000000):
10     clientSocket.sendto(str(x).encode(), (serverName, serverPort))
11     modifiedMessage, serverAddress = clientSocket.recvfrom(serverPort)
12
13 print("---- %s seconds ----" % (time.time() - startTime))
14 clientSocket.close()
```

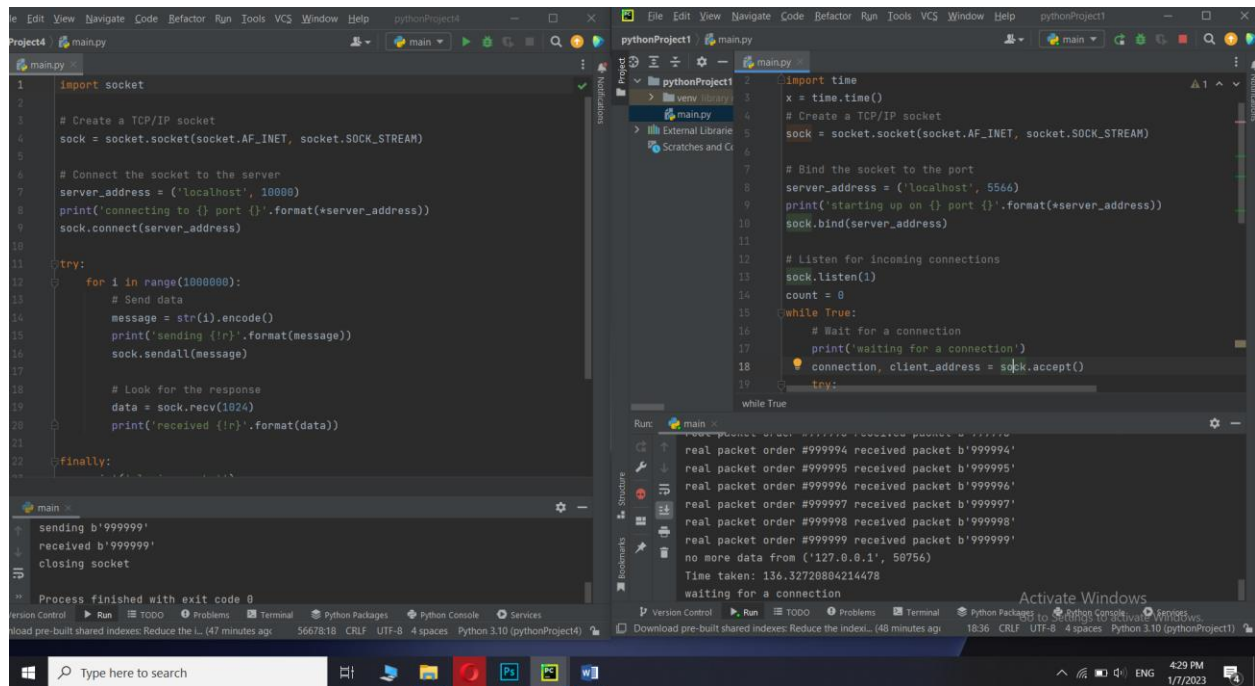
the counter of received messages is 999996
b'999996'
the counter of received messages is 999997
b'999997'
the counter of received messages is 999998
b'999998'
the counter of received messages is 999999
b'999999'
the counter of received messages is 1000000

C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject\main.py
--- 240.92919278144836 seconds ---
Process finished with exit code 0

Figure 1 : Client and server UDP system on the same computer

We have the results UCP time to send from a client to server is 240.92 seconds, and there is no packet loss.

Second, The TCP by Python Socket Programing and we calculate the time and the number of packets received.



The image shows two side-by-side IDE windows. The left window, titled 'Project4', contains a Python script for a TCP client. The right window, titled 'pythonProject1', contains a Python script for a TCP server. Both scripts use the 'socket' module. The client script sends a message 'b'999999' and receives a response 'b'999999'. The server script listens on port 5566 and accepts a connection from the client. The Run console in the right window shows the execution output, including the time taken (136.32720884214478 seconds) and the number of packets received (9).

```
1 import socket
2
3 # Create a TCP/IP socket
4 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5
6 # Connect the socket to the server
7 server_address = ('localhost', 10000)
8 print('connecting to {} port {}'.format(*server_address))
9 sock.connect(server_address)
10
11 try:
12     for i in range(1000000):
13         # Send data
14         message = str(i).encode()
15         print('sending ({}).format(message))
16         sock.sendall(message)
17
18         # Look for the response
19         data = sock.recv(1024)
20         print('received ({}).format(data))
21
22 finally:
23     sock.close()
24
25 main ->
26 sending b'999999'
27 received b'999999'
28 closing socket
29 Process finished with exit code 0
```

```
1 import time
2 x = time.time()
3
4 # Create a TCP/IP socket
5 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6
7 # Bind the socket to the port
8 server_address = ('localhost', 5566)
9 print('starting up on {} port {}'.format(*server_address))
10 sock.bind(server_address)
11
12 # Listen for incoming connections
13 sock.listen(1)
14 count = 0
15
16 while True:
17     # Wait for a connection
18     print('waiting for a connection')
19     connection, client_address = sock.accept()
20     try:
21         # Receive the data from the client
22         data = connection.recv(1024)
23         print('received {} bytes from {}'.format(data.decode(), client_address))
24         count += 1
25         # Send the data back to the client
26         connection.sendall(data)
27     finally:
28         connection.close()
29
30 print('Time taken: {} seconds'.format(time.time() - x))
31 print('Total packets received: {}'.format(count))
32
33 Run:
34 real packet order #999994 received packet b'999994'
35 real packet order #999995 received packet b'999995'
36 real packet order #999996 received packet b'999996'
37 real packet order #999997 received packet b'999997'
38 real packet order #999998 received packet b'999998'
39 real packet order #999999 received packet b'999999'
40 no more data from ('127.0.0.1', 50756)
41 Time taken: 136.32720884214478
42 waiting for a connection
```

figure: TCP Client and Server System at the same computer

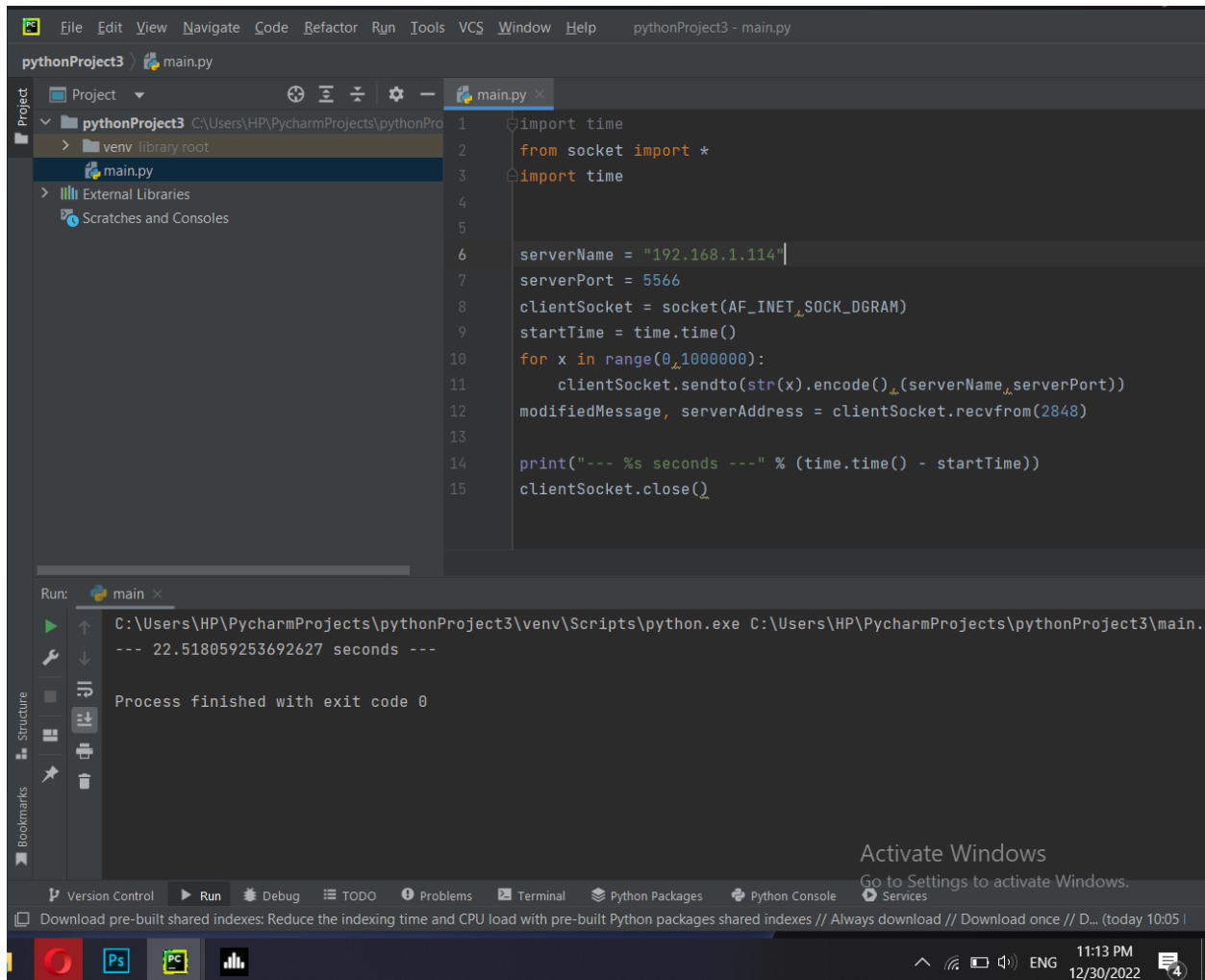
We recognized that there are no loss, and time taken is 136.327 seconds.

2.2) on 2 different computers connected by a cable directly or through a switch

We faced problems in this part, the code didn't run in 3 labtops.

2.3) on 2 different computers connected through WiFi

On the UDP, we use the IP of server address computer that both computers are connected on the same Wi-fi, and put it as a server name, then we have run both codes.



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script named `main.py` with the following code:

```
1 import time
2 from socket import *
3 import time
4
5
6 serverName = "192.168.1.114"
7 serverPort = 5566
8 clientSocket = socket(AF_INET, SOCK_DGRAM)
9 startTime = time.time()
10 for x in range(0, 1000000):
11     clientSocket.sendto(str(x).encode(), (serverName, serverPort))
12     modifiedMessage, serverAddress = clientSocket.recvfrom(2848)
13
14 print("--- %s seconds ---" % (time.time() - startTime))
15 clientSocket.close()
```

The Run tool window at the bottom shows the execution output:

```
Run: main
C:\Users\HP\PycharmProjects\pythonProject3\venv\Scripts\python.exe C:\Users\HP\PycharmProjects\pythonProject3\main.py
--- 22.518059253692627 seconds ---
Process finished with exit code 0
```

The status bar at the bottom indicates the system time as 11:13 PM on 12/30/2022.

Figure n : The UDP Client on different computers with same Wi-fi

We noticed that the time that the process takes is 22.5 seconds that is less than the system on the same computer, because of packet loss from the server side.

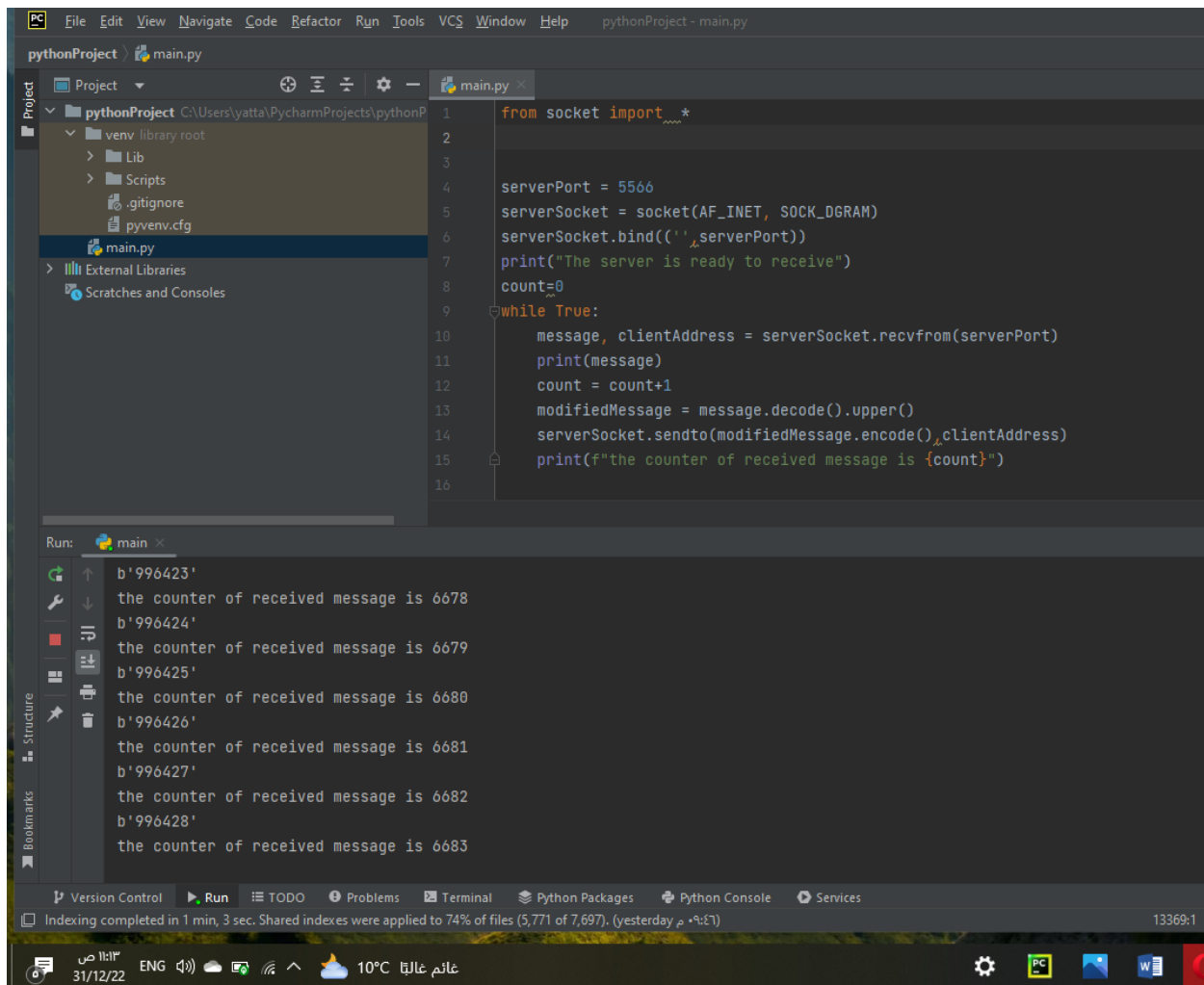


Figure n : The UDP Server on different computers with same Wi-fi

We noticed that there is a loss of packets by comparing the counter of the received messages and the number of the packet that came and they are not the same, and the UDP keep his work without resend it (that's how the UDP works!).

- TCP Client and Server in different Computers connected at the Same Wifi
-

The screenshot shows an IDE window titled 'pythonProject1' with a file named 'main.py' open. The code in 'main.py' is a Python TCP server script. It starts by printing 'waiting for a connection', then uses 'socket.accept()' to accept a connection. It enters a 'try' block where it prints the client address, then enters a 'while True' loop to receive and process data. Inside the loop, it receives data in 1024-byte chunks. If data is received, it prints the packet order and content, increments a counter, and sends the data back. If no more data is received, it prints a message and breaks the loop. Finally, it calculates and prints the time taken. The script ends with a 'finally' block.

```
17 print('waiting for a connection')
18 connection, client_address = socket.accept()
19 try:
20     print('connection from', client_address)
21
22     # Receive and process each number individually
23     while True:
24         data = connection.recv(1024)
25         if data:
26             print("real packet order #" + str(count) + ' received packet b' + str(data))
27             count += 1
28             connection.sendall(data)
29         else:
30             print('no more data from', client_address)
31             x = time.time() - x
32             print("Time taken: " + str(x))
33             break
34     finally:
```

The Run window shows the output of the script. It displays the received packets and the time taken for the server to process them. The output is as follows:

```
real packet order #999995 received packet b'999995'
real packet order #999996 received packet b'999996'
real packet order #999997 received packet b'999997'
real packet order #999998 received packet b'999998'
real packet order #999999 received packet b'999999'
no more data from ('192.168.1.6', 51451)
Time taken: 3749.7678430080414
waiting for a connection
```

The bottom of the image shows the Windows taskbar with the system clock at 10:18 PM on 1/7/2023.

The screenshot shows an IDE window titled 'pythonProject2' with a file explorer on the left and a run console at the bottom. The main editor displays the code for 'client.py'. The code creates a TCP/IP socket, connects to a server at '192.168.1.8' on port '5566', and sends a series of messages from '999997' to '999999'. A 'try' block contains a 'for' loop that iterates from 1000000 down to 999997. The output console shows the execution of this code, displaying the messages being sent and received, and confirming that the process finished with exit code 0.

```
pythonProject2 > client.py
# Create a TCP/IP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Connect the socket to the server
server_address = ('192.168.1.8', 5566)
print('connecting to {} port {}'.format(*server_address))
sock.connect(server_address)

try:
    for i in range(1000000, 999997, -1):
        # Send data
        message = str(i).encode()
        print('sending {!r}'.format(message))
        sock.sendall(message)

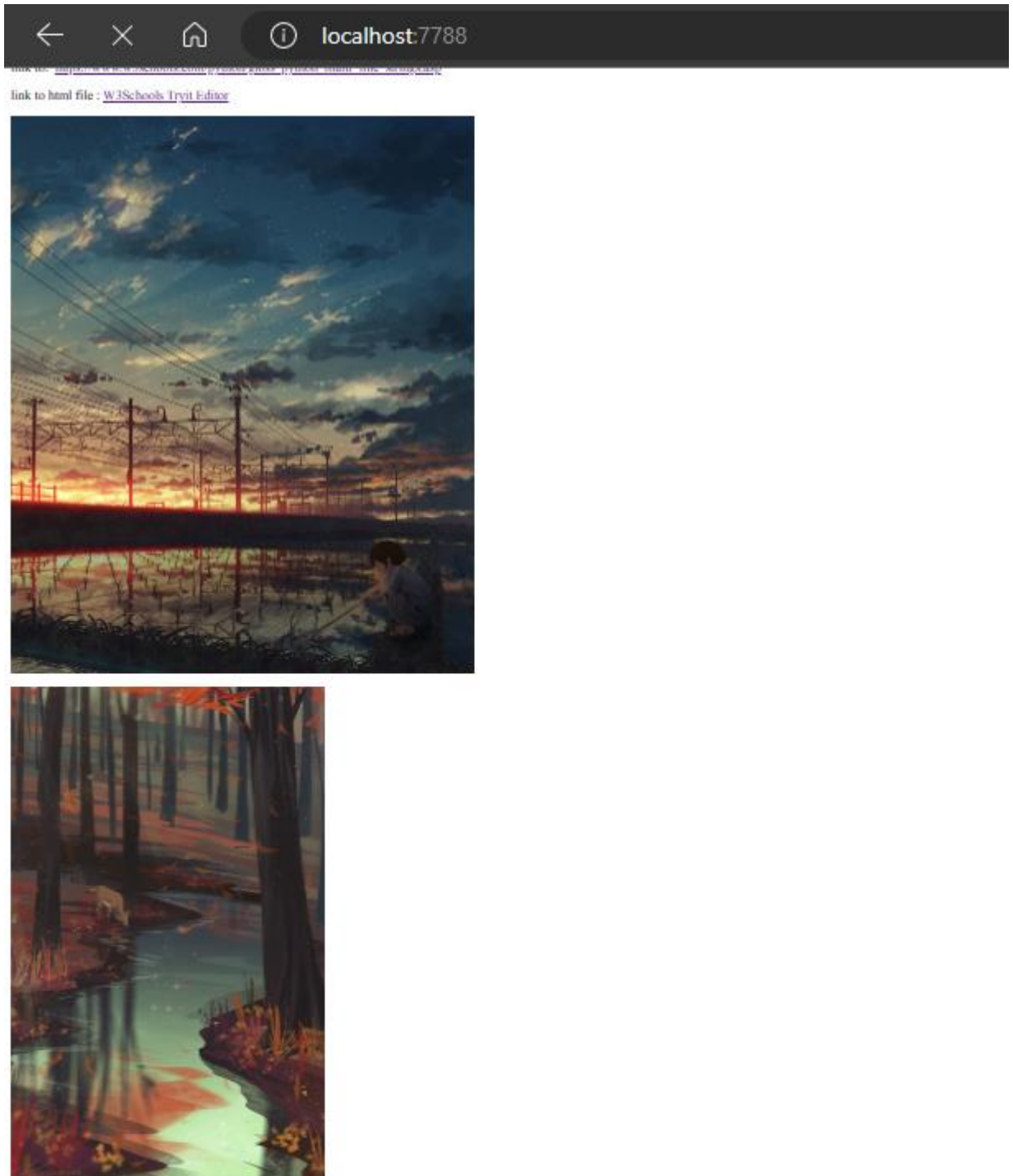
        # Look for the response
        data = sock.recv(1024)

except:
    pass

Process finished with exit code 0
```

We recognize that all packets have been received(no packet loss), not Like the udp case, but the time 3450 second.

- 1- if the request is / or /index.html or /main_en.html or /en (for example localhost:7788/ or localhost:7788/en) then the server should send main_en.html file with Content-Type: text/html.



- 2-

ENCS3320-My First Webserver

Welcome to our course [Computer Networks](#)

Group members

Ahmad zubaidia , id 1200105

Mohammad Makamri , id 1200227

Islam jihad , id 1191375

Some informaiton about the group

Ahmad is good, and love c languge

mohammad is good , and love java language

islam is the leader of the club and he is good , and love python

link to: https://www.w3schools.com/python/gloss_python_multi_line_strings.asp

link to html file : [W3Schools Tryit Editor](#)



```
The server is ready to receive
Got connection from IP: 127.0.0.1, Port: 49941
GET / HTTP/1.1
Host: localhost:7788
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```


ENCS3320-My First Webserver

Welcome to our course [Computer Networks](#)

Group members

Ahmad zubaidia , id 1200105

Mohammad Makamri , id 1200227

Islam jihad , id 1191375

Some informaiton about the group

Ahmad is good, and love c langage

mohammad is good , and love java language

islam is the leader of the club and he is good , and love python

link to: https://www.w3schools.com/python/gloss_python_multi_line_strings.asp

link to html file : [W3Schools Tryit Editor](#)



```
Got connection from IP: 127.0.0.1, Port: 49971
GET /main_en.html HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```

ENCS3320-My First Webserver

Welcome to our course [Computer Networks](#)

Group members

Ahmad zubaidia , id 1200105

Mohammad Makamri , id 1200227

Islam jihad , id 1191375

Some informaiton about the group

Ahmad is good, and love c language

mohammad is good , and love java language

islam is the leader of the club and he is good , and love python

link to: https://www.w3schools.com/python/gloss_python_multi_line_strings.asp

link to html file : [W3Schools Tryit Editor](#)



```
Got connection from IP: 127.0.0.1, Port: 50000
GET /index.html HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```

ENCS3320-My First Webserver

Welcome to our course [Computer Networks](#)

Group members

Ahmad zubaidia , id 1200105

Mohammad Makamri , id 1200227

Islam jihad , id 1191375

Some informaiton about the group

Ahmad is good, and love c language

mohammad is good , and love java language

islam is the leader of the club and he is good , and love python

link to: https://www.w3schools.com/python/gloss_python_multi_line_strings.asp

link to html file : [W3Schools Tryit Editor](#)



```
Got connection from IP: 127.0.0.1, Port: 50014
GET /en HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Id=a-8286ae51=ae27e1f8-8821-4da6-b0c7-770e16e366a2
```

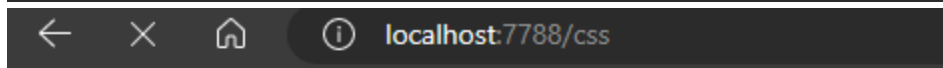
3&if the request is an .html file then the server should send the requested html file with Content-Type: text/html. You can use any html file. 4- 5- if the request is a .css file then the server should send the requested css file with Content-Type: text/css. You can use any CSS file



My Html file that the 3- part required

My first html.

```
Got connection from IP: 127.0.0.1, Port: 50018
GET /htmlfile HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```



A Blue Heading

A red paragraph.

```
Got connection from IP: 127.0.0.1, Port: 50023
GET /css HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```

2- If the request is /ar then the server response with main_ar.html which is an Arabic version of main_en.html

أول سيرفر لي-ENCS3320

مرحبا بكم في دوراتنا التدريبية حول شبكات الكمبيوتر

أعضاء المجموعة

احمد زيبيديا 1200105

محمد مكامري 1200227

جهاد الإسلام ، 1191375

يعض المعلومات عن المجموعة

```
Got connection from IP: 127.0.0.1, Port: 50336
GET /main_ar.html HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```

أول سيرفر لي-ENCS3320

مرحبا بكم في دوراتنا التدريبية حول [شبكات الكمبيوتر](#)

أعضاء المجموعة

احمد زينبيا 1200105

محمد مكاسري 1200227

جهاد الإسلام ، 1191375

بعض المعلومات عن المجموعة

احمد طيب ويحب لغة سي

محمد طيب ويحب لغة جافا

الإسلام زعيم النادي وهو طيب ويحب البايثون

رابط الى https://www.w3schools.com/python/gloss_python_multi_line_strings.php

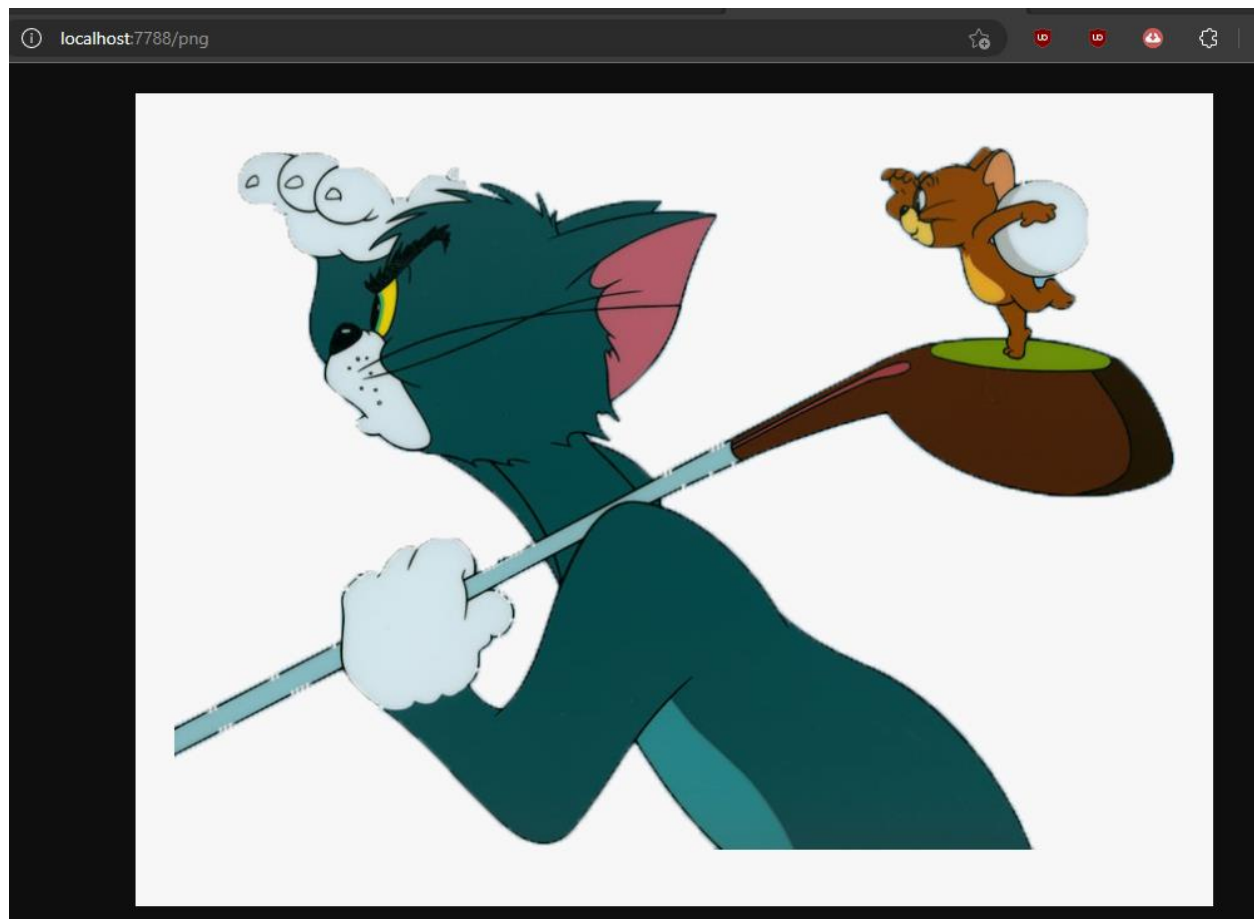
ملف اتيهمل : [W3Schools Tryit Editor](#)





```
Got connection from IP: 127.0.0.1, Port: 50027
GET /ar HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```

7- if the request is a .png then the server should send the png image with Content-Type: image/png.
You can use any image. 8- if the request is a .jpg then the server should send the jpg image with Content-Type: image/jpeg. You can use any image.




```
Got connection from IP: 127.0.0.1, Port: 50049
GET /png HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```



```
Got connection from IP: 127.0.0.1, Port: 50056
GET /jpg HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```

10- If the request is wrong or the file doesn't exist the server should return a simple HTML webpage that contains (Content-Type: text/html) 1- "HTTP/1.1 404 Not Found" in the response status 2- "Error" in the title 3- "The file is not found" in the body in red 4- Your names and IDs in Bold 5- The IP and port number of the client

← ↻ 🏠 ⓘ localhost:7788/anything

Error

The file is not found.

Group members names and IDs

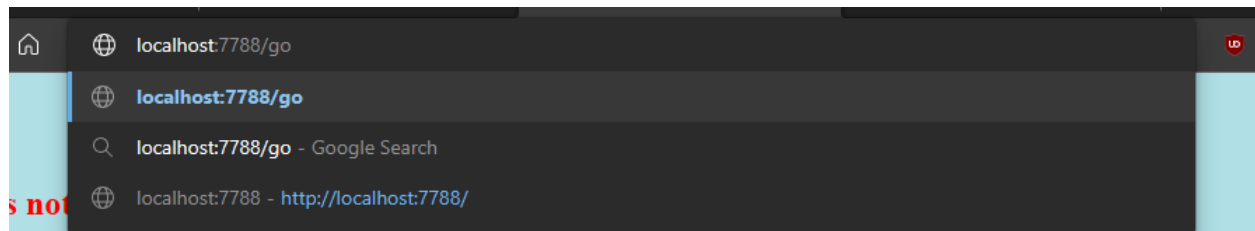
Ahmed Zubaidia , ID = 1200105

Mohammad Makhamri, ID = 1200227

Islam jihad , ID = 119758

the ip is :127.0.0.1 the port number is :50095

```
Got connection from IP: 127.0.0.1, Port: 50095
GET /anything HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```



bers names and IDs

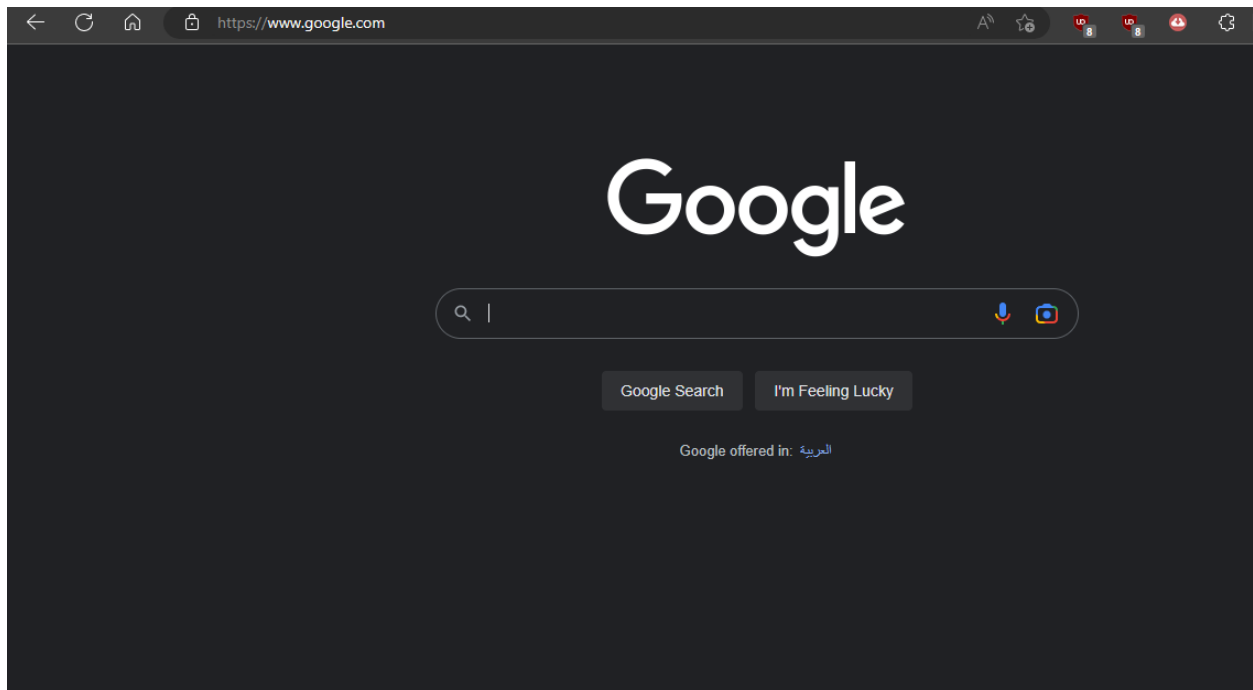
dia , ID = 1200105

Makhamri, ID = 1200227

ID = 119758

.0.1 the port number is :50095

9- Use the status code 307 Temporary Redirect to redirect the following a. If the request is /go then redirect to google website b. If the request is /so then redirect to stackoverflow.com website c. If the request is /bzu then redirect to birzeit university website



```
The server is ready to receive
Got connection from IP: 127.0.0.1, Port: 50151
GET /go HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Mi
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) A
Accept: text/html,application/xhtml+xml,application/xml
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

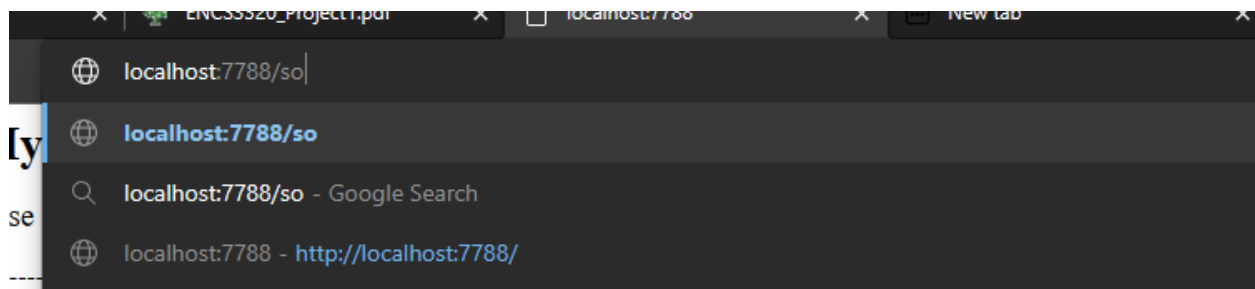
```

elif s[0] == 'go':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect \r\n".encode())
    connectionSocket.send('Content-Type: text/html \r\n'.encode())
    connectionSocket.send("Location: https://www.google.com \r\n".encode())
    connectionSocket.send('\r\n'.encode())

elif s[0] == 'so':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect \r\n".encode())
    connectionSocket.send('Content-Type: text/html \r\n'.encode())
    connectionSocket.send("Location: https://stackoverflow.com/ \r\n".encode())
    connectionSocket.send('\r\n'.encode())


elif s[0] == 'bzu':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect \r\n".encode())
    connectionSocket.send('Content-Type: text/html \r\n'.encode())
    connectionSocket.send("Location: https://www.birzeit.edu/ \r\n".encode())
    connectionSocket.send('\r\n'.encode())

```



0105


1 1200227



Find the best answer to your technical question, help others answer theirs

[Join the community](#)

or [search content](#)

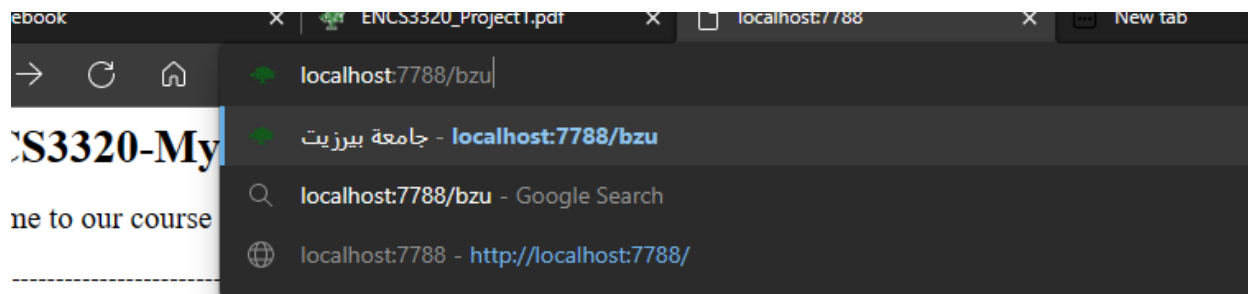


Want a secure, private space for your technical knowledge?

[Discover Teams](#)

Every **developer** has a tab open to Stack Overflow

```
The server is ready to receive
Got connection from IP: 127.0.0.1, Port: 50184
GET /so HTTP/1.1
Host: localhost:7788
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```

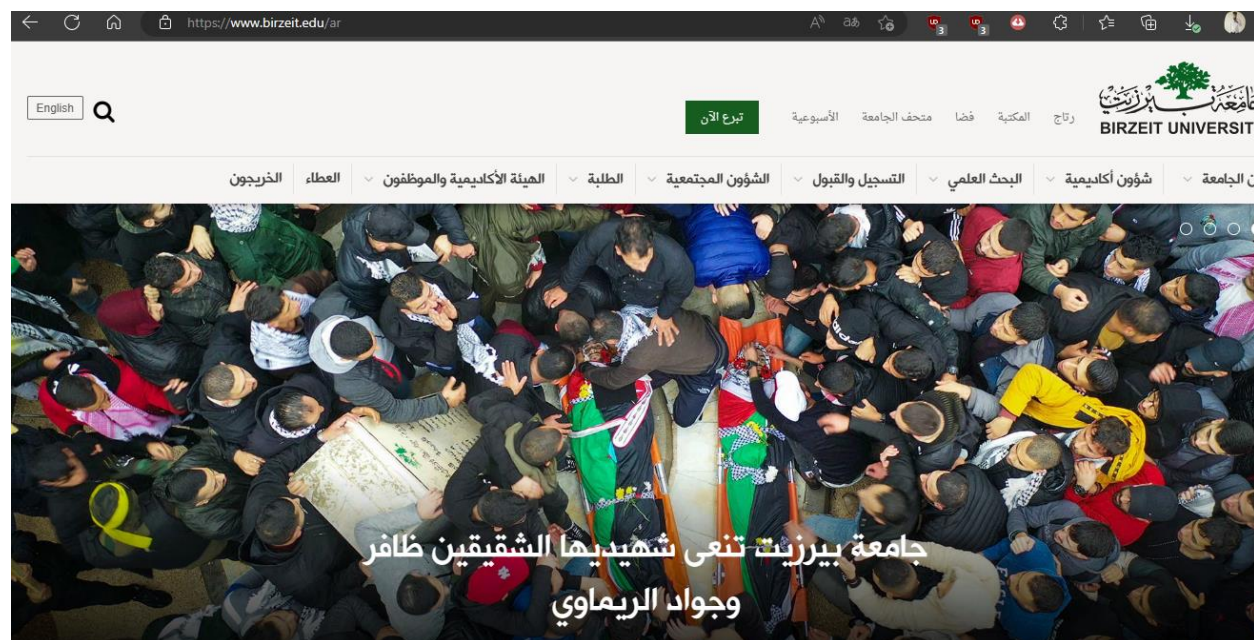


members

zubaidia , id 1200105

nad Makamri , id 1200227

nad , id 1191375



```
Got connection from IP: 127.0.0.1, Port: 50207
GET /bzu HTTP/1.1
Host: localhost:7788
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Idea-8296eef1=ec27e1f8-8821-4da4-b0c7-770c146c346a
```

Code :

```
import re
from socket import *

serverPort = 7788
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(("localhost", serverPort))
serverSocket.listen(1)
print('The server is ready to receive')
while True:
    connectionSocket, addr = serverSocket.accept()
    ip = addr[0]
    port = addr[1]
    print('Got connection from', "IP: " + ip + ", Port: " + str(port))
    sentence = connectionSocket.recv(1024).decode()
    print(sentence)

    x = re.split('/', sentence)
    s = x[1].split()
    a = 'HTTP'

    if s[0] == 'en' or s[0] == 'index.html' or s[0] == 'main_en.html' or
s[0] == a:

        f1 = open("mainEn.html", "rb")
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send('Content-Type: text/html \r\n'.encode())
        connectionSocket.send('\r\n'.encode())
```



```

        connectionSocket.send(f1.read())
    elif s[0] == 'ar' or 'main_ar.html':
        f2 = open("arabic.html", "rb")
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send('Content-Type: text/html \r\n'.encode())
        connectionSocket.send('\r\n'.encode())
        connectionSocket.send(f2.read())

    elif s[0] == 'htmlfile':
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send('Content-Type: text/html \r\n'.encode())
        f2 = open("3_htmlfile.html", "rb")
        connectionSocket.send(f2.read())

    elif s[0] == 'css':
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send('Content-Type: text/html \r\n'.encode())
        f2 = open("CSS.css", "rb")
        connectionSocket.send(f2.read())

    elif s[0] == 'png':
        ip = addr[0]
        port = addr[1]
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send('Content-Type: image/png \r\n'.encode())
        connectionSocket.send('\r\n'.encode())

        f2 = open("tom-and-jerry.png", "rb")
        connectionSocket.send(f2.read())

    elif s[0] == 'jpg':
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send('Content-Type: image/jpeg\r\n'.encode())
        connectionSocket.send('\r\n'.encode())

        f2 = open("el.jpg", "rb")
        connectionSocket.send(f2.read())

    elif s[0] == 'go':
        connectionSocket.send("HTTP/1.1 307 Temporary Redirect
\r\n".encode())
        connectionSocket.send('Content-Type: text/html \r\n'.encode())
        connectionSocket.send("Location: https://www.google.com
\r\n".encode())
        connectionSocket.send('\r\n'.encode())

    elif s[0] == 'so':
        connectionSocket.send("HTTP/1.1 307 Temporary Redirect
\r\n".encode())
        connectionSocket.send('Content-Type: text/html \r\n'.encode())
        connectionSocket.send("Location: https://stackoverflow.com/
\r\n".encode())
        connectionSocket.send('\r\n'.encode())

    elif s[0] == 'bzu':
        connectionSocket.send("HTTP/1.1 307 Temporary Redirect
\r\n".encode())
        connectionSocket.send('Content-Type: text/html \r\n'.encode())
        connectionSocket.send("Location: https://www.birzeit.edu/
\r\n".encode())
        connectionSocket.send('\r\n'.encode())

```

```
elif s[0] == 'jpg':
    connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
    connectionSocket.send('Content-Type: image/jpeg\r\n'.encode())
    connectionSocket.send('\r\n'.encode())
else :
    f2 = open("erorr.html", "rb")
    connectionSocket.send("\r\nHTTP/1.1 404 Not Found \r\n".encode())
    connectionSocket.send('Content-Type: text/html \r\n'.encode())
    connectionSocket.send('\r\n'.encode())
    connectionSocket.send(f2.read())
    s= ip
    z=str(port)

    r= "ip is"

    connectionSocket.send("the ip is :".encode() )
    connectionSocket.send(s.encode() )
    connectionSocket.send("\n".encode() )
    connectionSocket.send("the port number is :".encode())
    connectionSocket.send(z.encode())

    # connectionSocket.send("the port number is".encode() z.encode())
# connectionSocket.close()
```