BIRZEIT UNIVERSITY

Department of Electrical and Computer Engineering

ENCS3320 – Group Number: 46

Project 1

	Students Name	Students Number
1	Yousef Sharbi	1202057
2	Anas karakra	1200467

Part (1):

1.

Ping

Is a network tool used to test the reachability between the host and the server and measure the time it takes for a small data set to be transmitted from a device to a server on the Internet and back to your device again, and it's time is measured in milliseconds (ms).

Tracert

Is a network tool (command) used to display the path and measure the transit delays of packets across a network and it determines the route to a destination by sending Internet Control Message Protocol (ICMP) echo packets to the destination.

Nslookup

Is a network utility tool (command) used to find (DNS) servers and IP address with domain name using a host name.

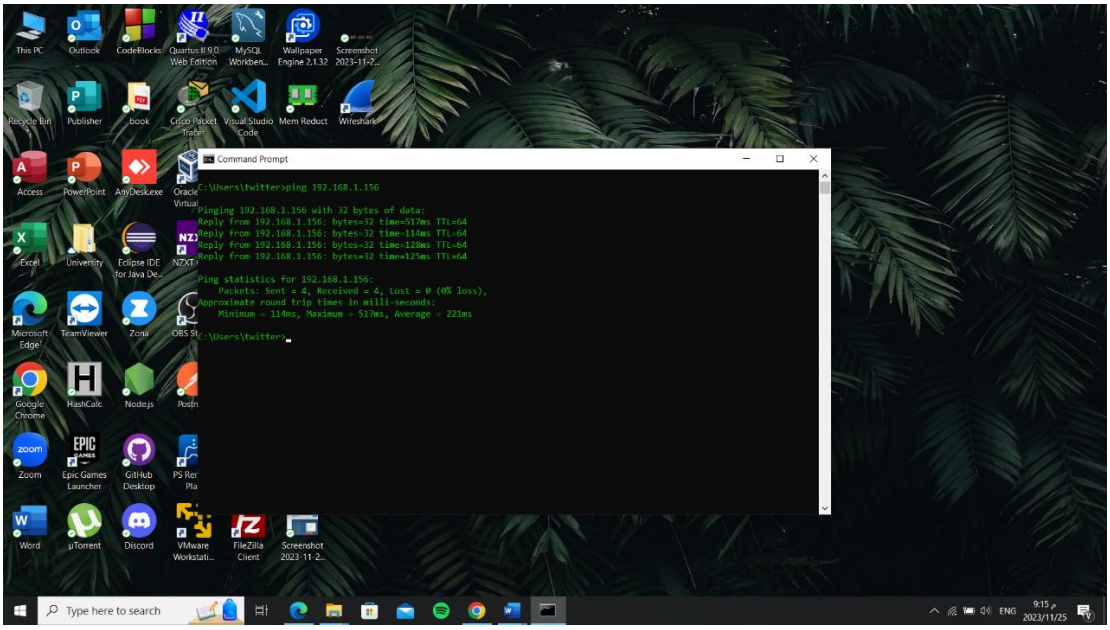
Telnet

It stands for (Teletype Network) and it is a protocol or tool (command) that lets a computer to connect to another host locally and it provides text-oriented communication.

2.

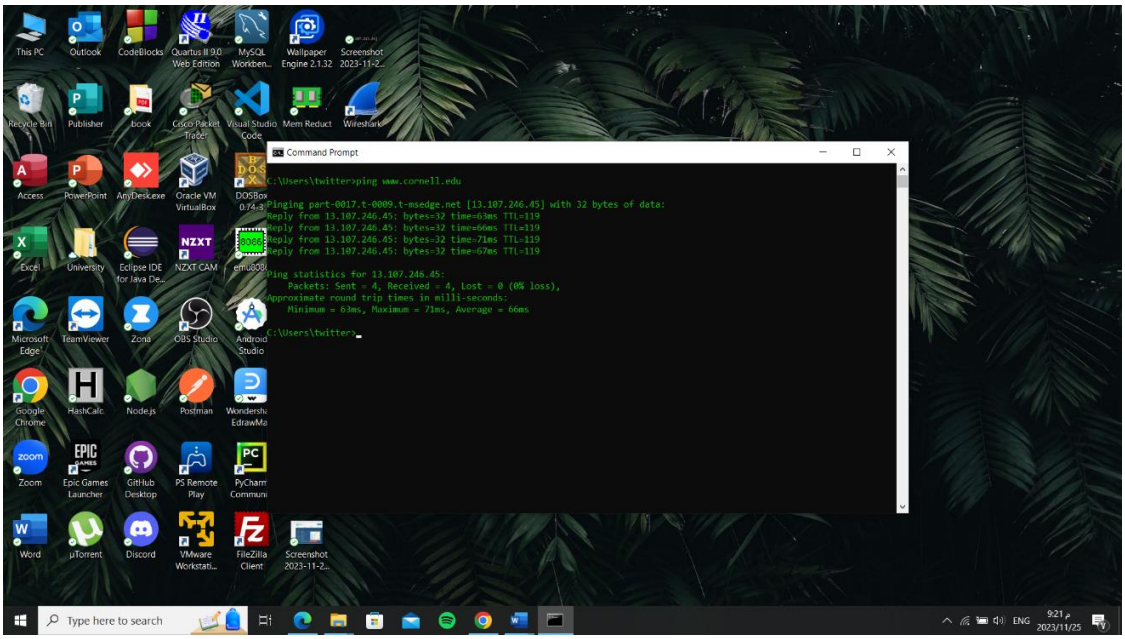
1_ command is: ping my IP_address

This command will send several ICMP echo request packets to the specified IP address (192.168.1.156) and display the round-trip time for each packet.



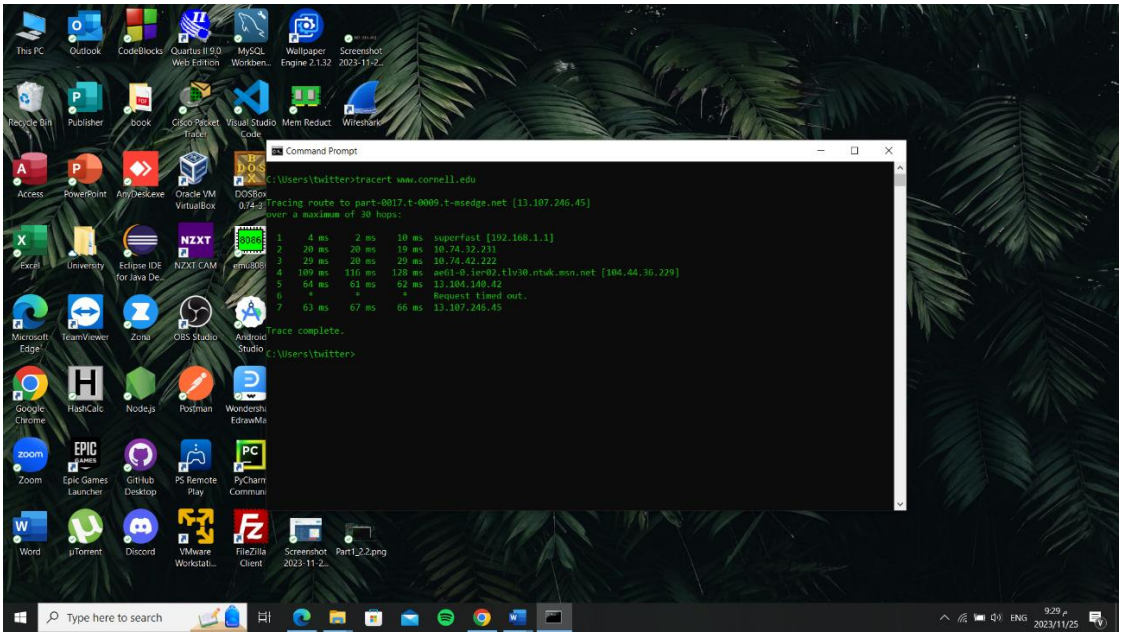
2_ command is: ping www.cornell.edu

This command will send several ICMP Request packets to the domain www.cornell.edu. The server (or another device on the route) will then respond with ICMP Echo Reply packets. The purpose of this is to test the network connection between your machine and the destination, in this case, cornell website.



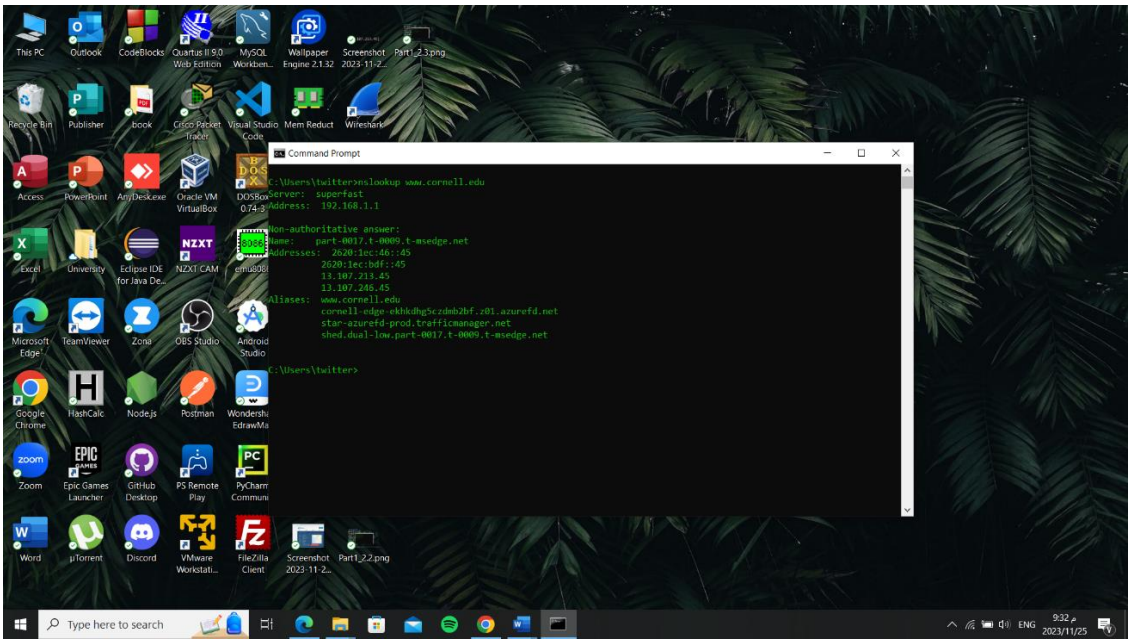
3. command is: tracert www.cornell.edu

This command traces the route that packets take to get from your computer to the destination domain and it will return output indicating the hops discovered and the time for each hop.



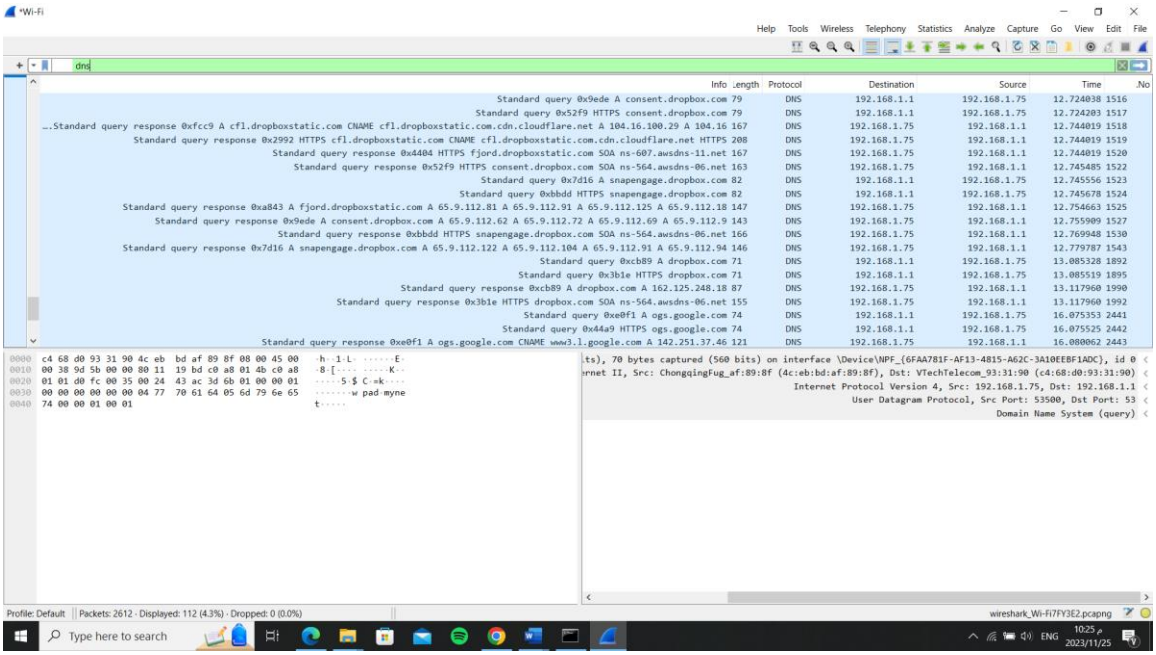
4. command is: nslookup www.cornell.edu

This command will obtain the (DNS) servers and domain name or IP address information.



3.

Some DNS messages from Wireshark:



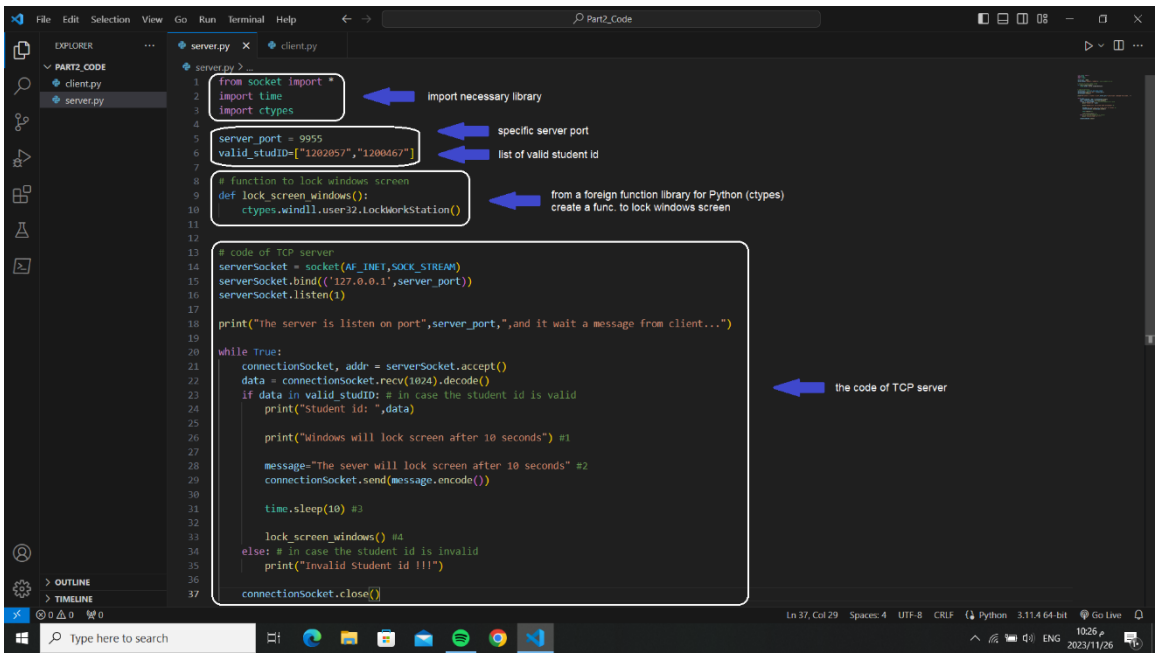
From the ping results, do you think the response you have got is from USA?

No, cause ping result do not provide information about the geographical location of the server which is impossible to determine if the response from USA.

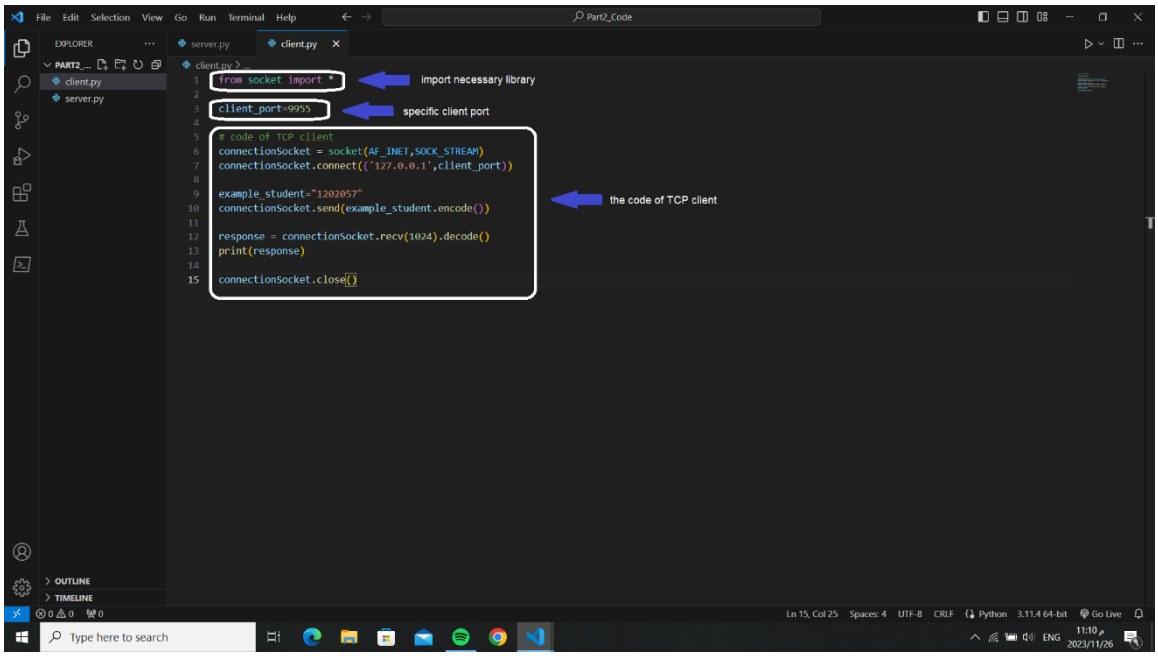
Part (2):

I used socket programming to implement TCP client and server in python language that listening on port 9955 in two separate files .py:

Server.py



client.py



At first in server.py we create a TCP socket using (AF_INET: IPv4 addressing and SOCK_STREAM: TCP socket) and then bind the socket to an (default address: 127.0.0.1 and specific port: 9955) and after that we set it Listen for one incoming connection from port and it wait a message from client then print that.

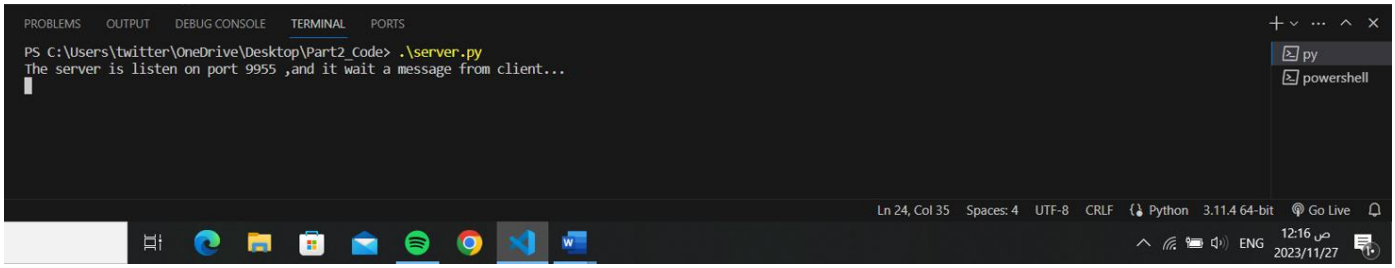
Server Code:

```
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind(('127.0.0.1',server_port))
serverSocket.listen(1)

print("The server is listen on port",server_port,",and it wait a message from client...")
```

And then get inside infinite loop which it accepts client connections and wait for the data that coming from client.

```
while True:
    connectionSocket, addr = serverSocket.accept()
    data = connectionSocket.recv(1024).decode()
```

Server Output:



Now, we go to client.py which we created a socket same of server.py and send an encoded message of example_studentID to the server.

Client Code:

```
from socket import *

client_port=9955

# code of TCP client
connectionSocket = socket(AF_INET,SOCK_STREAM)
connectionSocket.connect(('127.0.0.1',client_port))

example_student="1202057"
connectionSocket.send(example_student.encode())
```

Client Output:

```
PS C:\Users\twitter\OneDrive\Desktop\Part2_Code> .\client.py
```

after it the server decode this message and store it on data and then we have two cases:

Case 1: The student id is valid according to the valid_studID

0) print student id.

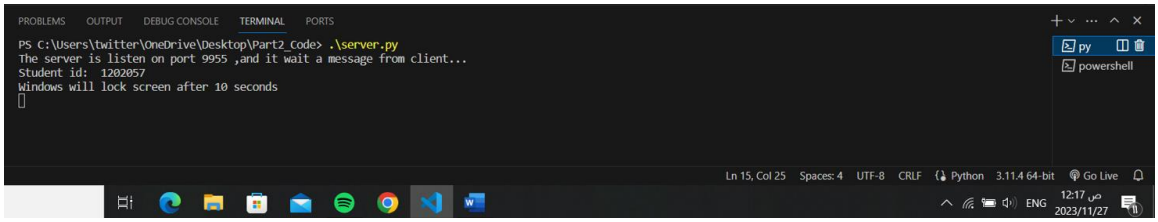
1) print on server side that the windows will lock screen after 10 seconds.

2) send an encoded message from server side to client side that the windows will lock screen after 10 seconds.

Server Code:

```
data = connectionSocket.recv(1024).decode()
if data in valid_studID: # in case the student id is valid
    print("Student id: ",data)

    print("Windows will lock screen after 10 seconds") #1

    message="The sever will lock screen after 10 seconds" #2
    connectionSocket.send(message.encode())
```

Server Output:

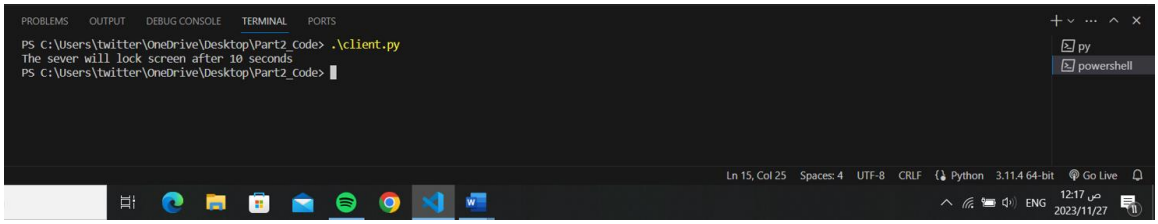


And then the client decodes this message and print it then close the socket connection.

Client Code:

```
response = connectionSocket.recv(1024).decode()
print(response)

connectionSocket.close()
```

Client Output:



3) using time library, we use sleep func. that make windows wait 10 second.

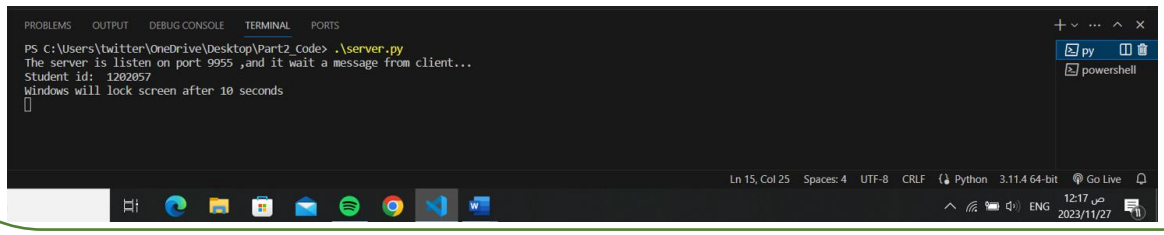
4) using ctypes library I create func. that lock the screen of windows.

Server Code:

```
time.sleep(10) #3

lock_screen_windows() #4
```

```
# function to lock windows screen
def lock_screen_windows():
    ctypes.windll.user32.LockWorkStation()
```

Server Output:



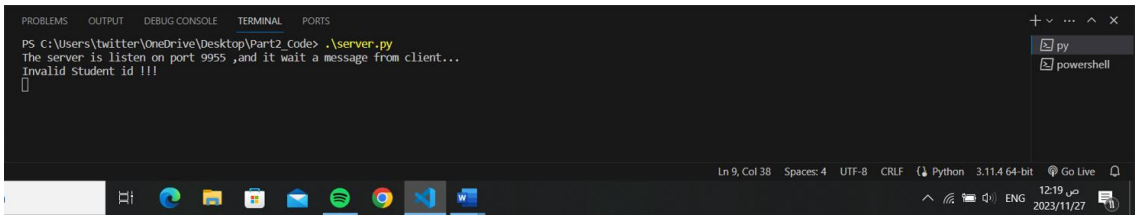
Case 2: The student id is Invalid

print that invalid student id on server side.

Server Code:

```
else: # in case the student id is invalid
    print("Invalid Student id !!!")
```

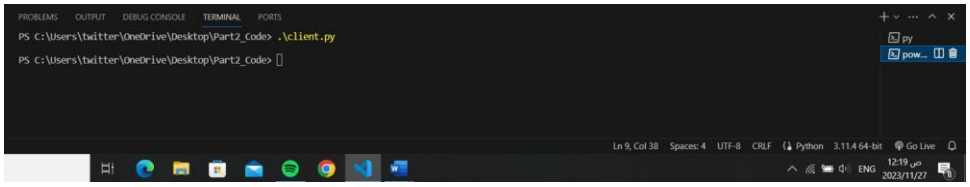
Server Output:



Client Code:

```
example_student="ttttttttttttt5655554"
connectionSocket.send(example_student.encode())
```

Client Output:

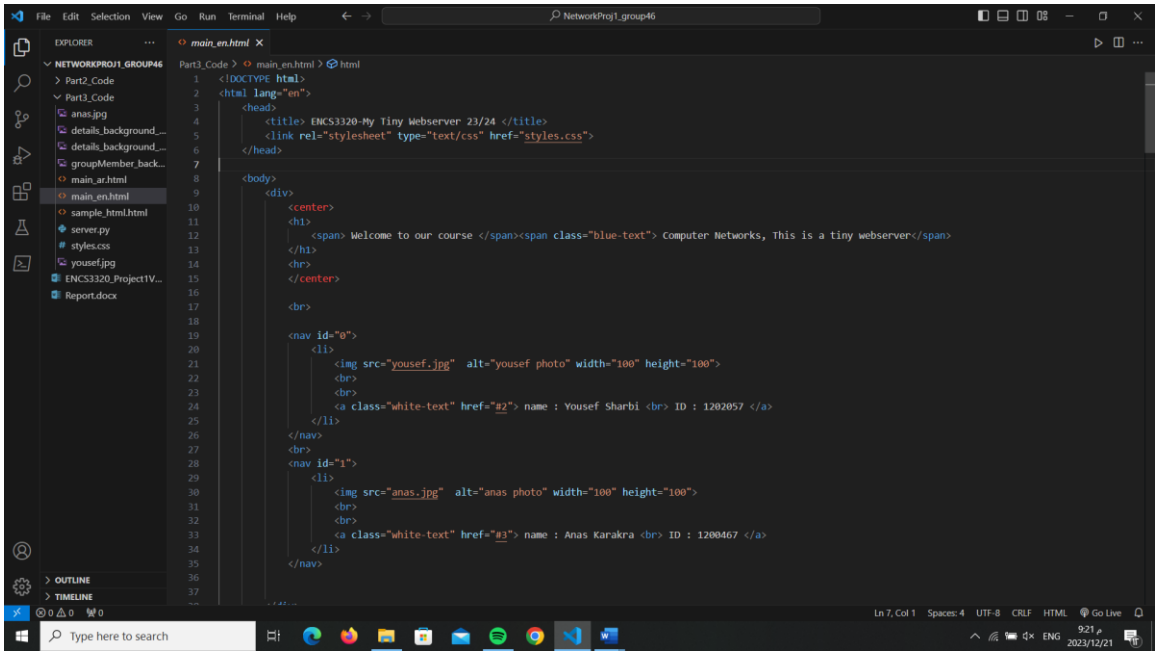


Part (3):

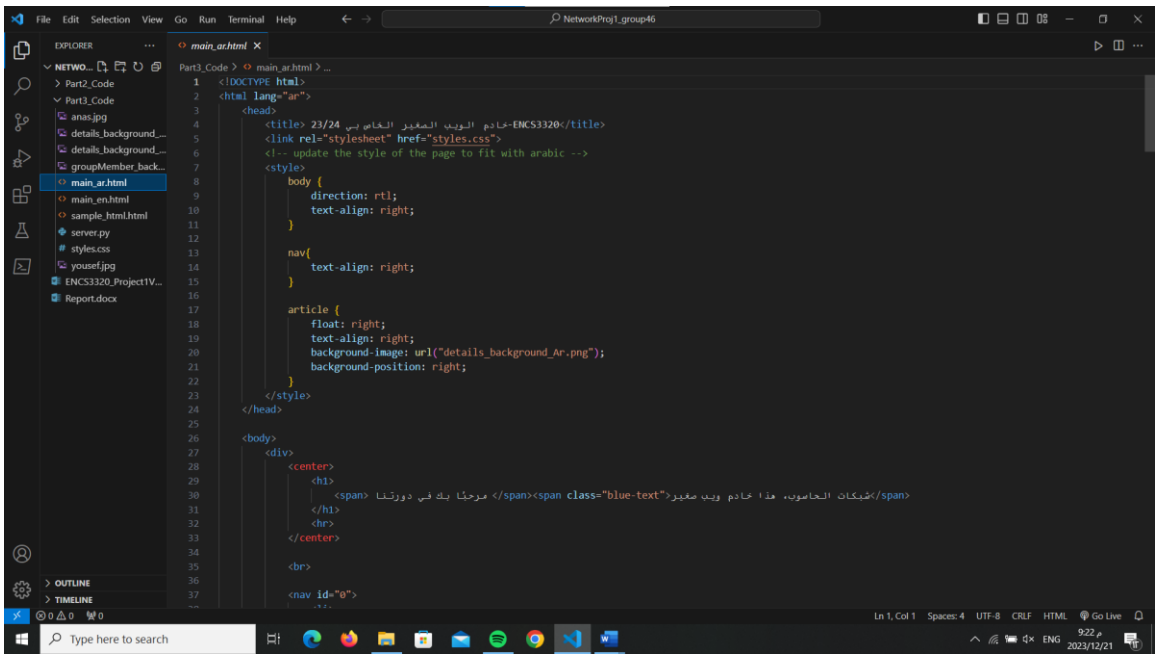
I used socket programming to implement complete web server in python language that listening on port 9966:

I create these file:

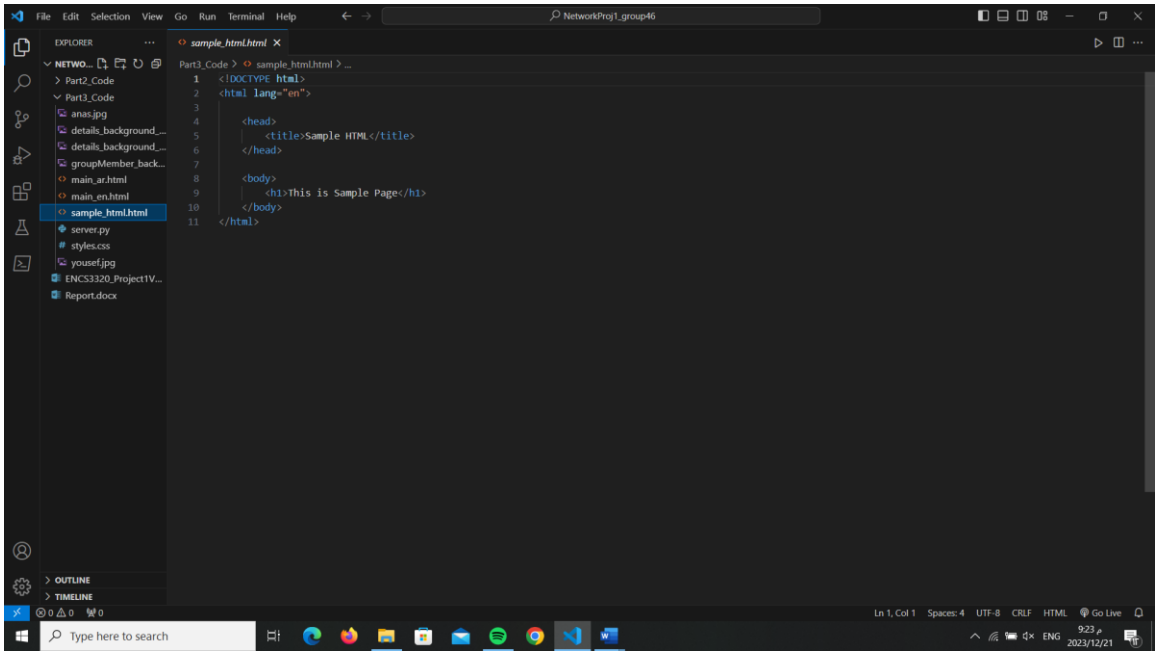
main_en.html:



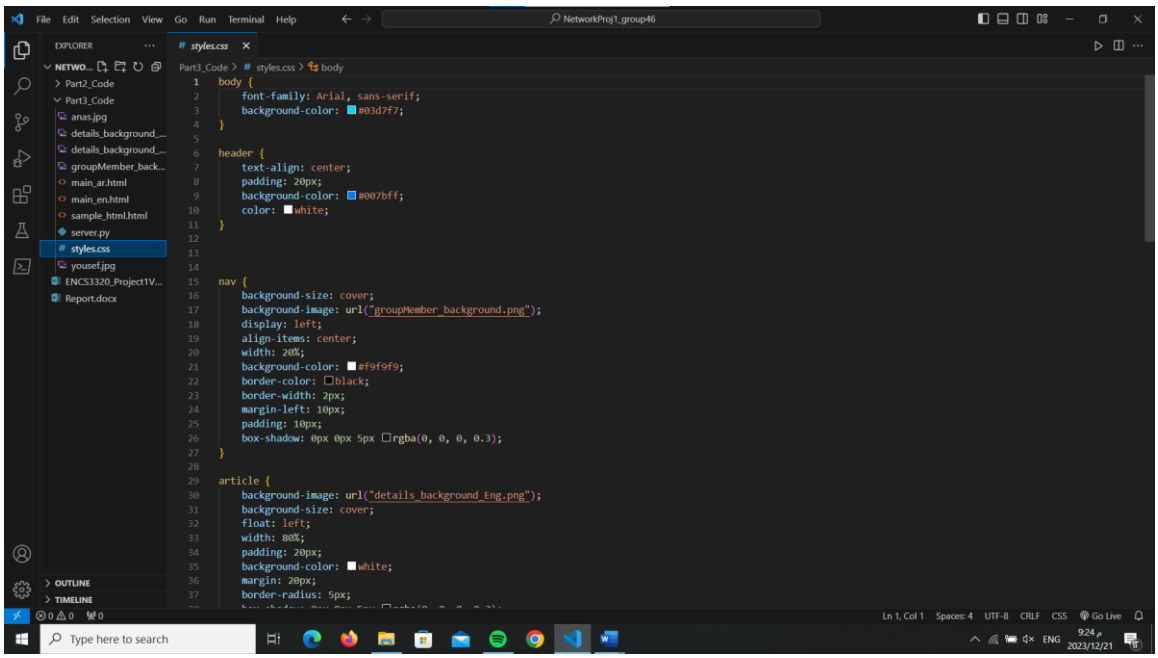
main_ar.html:



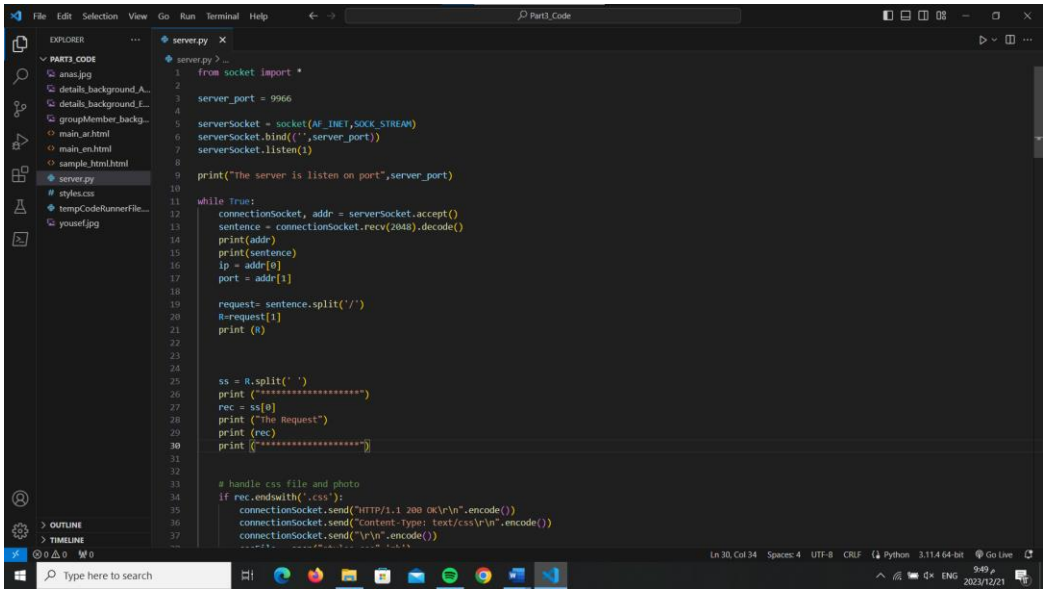
sample_html.html:



styles.css:

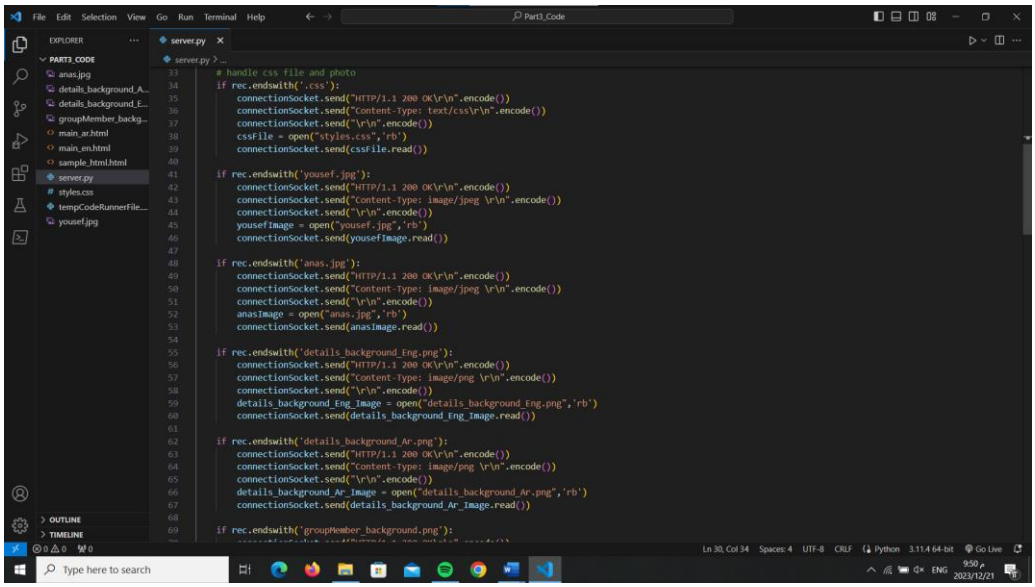


Server.py:



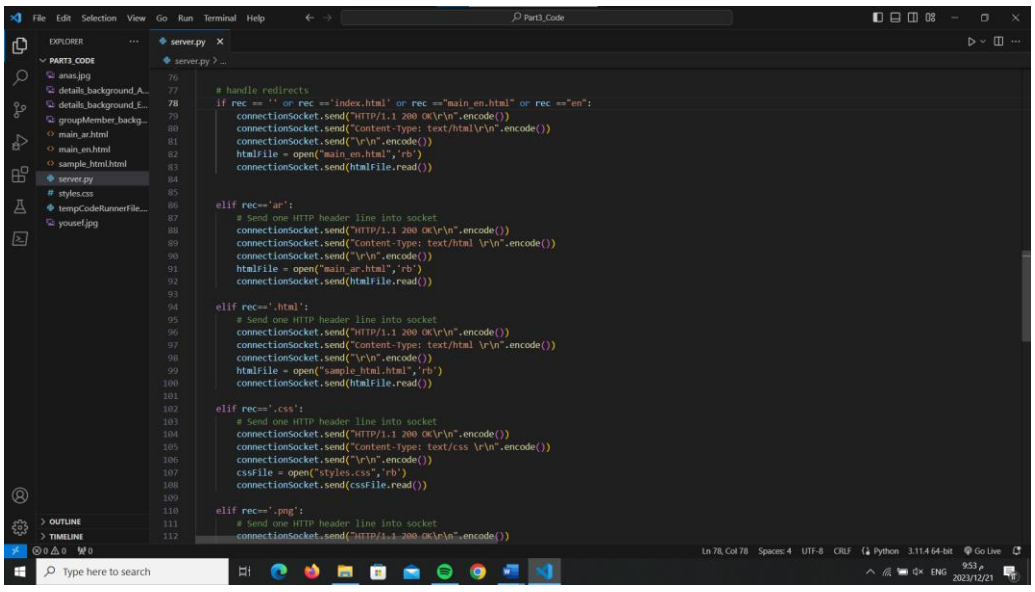
```python
from socket import *

server_port = 9966

serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind(('',server_port))
serverSocket.listen(1)

print("The server is listen on port",server_port)

while True:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(2048).decode()
    print(addr)
    print(sentence)
    ip = addr[0]
    port = addr[1]

    request= sentence.split('/')
    R=request[1]
    print (R)



    ss = R.split(' ')
    print ("********************")
    rec = ss[0]
    print ("The Request")
    print (rec)
    print ("********************")


    # handle css file and photo
    if rec.endswith('.css'):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send("Content-Type: text/css\r\n".encode())
        connectionSocket.send("\r\n".encode())
```

To make server handle css file and photos:



```python
    # handle css file and photo
    if rec.endswith('.css'):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send("Content-Type: text/css\r\n".encode())
        connectionSocket.send("\r\n".encode())
        cssFile = open("styles.css",'rb')
        connectionSocket.send(cssFile.read())

    if rec.endswith('yousef.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())
        yousefImage = open("yousef.jpg",'rb')
        connectionSocket.send(yousefImage.read())

    if rec.endswith('anas.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())
        anasImage = open("anas.jpg",'rb')
        connectionSocket.send(anasImage.read())

    if rec.endswith('details_background_Eng.png'):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send("Content-Type: image/png \r\n".encode())
        connectionSocket.send("\r\n".encode())
        details_background_Eng_Image = open("details_background_Eng.png",'rb')
        connectionSocket.send(details_background_Eng_Image.read())

    if rec.endswith('details_background_Ar.png'):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send("Content-Type: image/png \r\n".encode())
        connectionSocket.send("\r\n".encode())
        details_background_Ar_Image = open("details_background_Ar.png",'rb')
        connectionSocket.send(details_background_Ar_Image.read())

    if rec.endswith('groupMember_background.png'):
```

To handle direction:



```python
    # handle redirects
    if rec == '' or rec =='index.html' or rec =="main_en.html" or rec =="en":
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send("Content-Type: text/html\r\n".encode())
        connectionSocket.send("\r\n".encode())
        htmlFile = open("main_en.html",'rb')
        connectionSocket.send(htmlFile.read())


    elif rec=='ar':
        # Send one HTTP header line into socket
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send("Content-Type: text/html \r\n".encode())
        connectionSocket.send("\r\n".encode())
        htmlFile = open("main_ar.html",'rb')
        connectionSocket.send(htmlFile.read())

    elif rec=='.html':
        # Send one HTTP header line into socket
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send("Content-Type: text/html \r\n".encode())
        connectionSocket.send("\r\n".encode())
        htmlFile = open("sample_html.html",'rb')
        connectionSocket.send(htmlFile.read())

    elif rec=='.css':
        # Send one HTTP header line into socket
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
        connectionSocket.send("Content-Type: text/css \r\n".encode())
        connectionSocket.send("\r\n".encode())
        cssFile = open("styles.css",'rb')
        connectionSocket.send(cssFile.read())

    elif rec=='.png':
        # Send one HTTP header line into socket
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
```

In case: the request is **/ or /index.html or /main_en.html or /en**:

```
# handle redirects
if rec == '' or rec =='index.html' or rec =="main_en.html" or rec =="en":
    connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
    connectionSocket.send("Content-Type: text/html\r\n".encode())
    connectionSocket.send("\r\n".encode())
    htmlFile = open("main_en.html",'rb')
    connectionSocket.send(htmlFile.read())
```

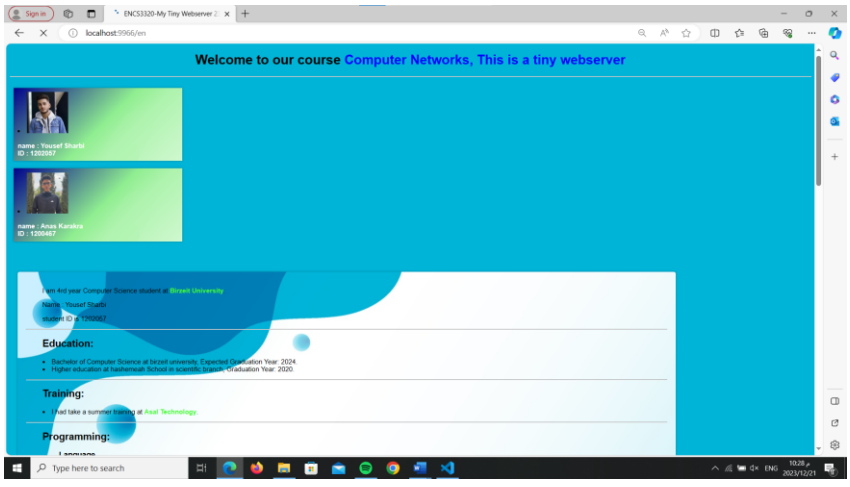
Output:

```
('127.0.0.1', 52826)
GET /favicon.ico HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:9966/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

```
('127.0.0.1', 52841)
GET /styles.css HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:9966/index.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

```
main_en.html HTTP
******************
The Request
main_en.html
******************
('127.0.0.1', 52894)
GET /styles.css HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:9966/main_en.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

```
en HTTP
******************
The Request
en
******************
('127.0.0.1', 52918)
GET /styles.css HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:9966/en
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

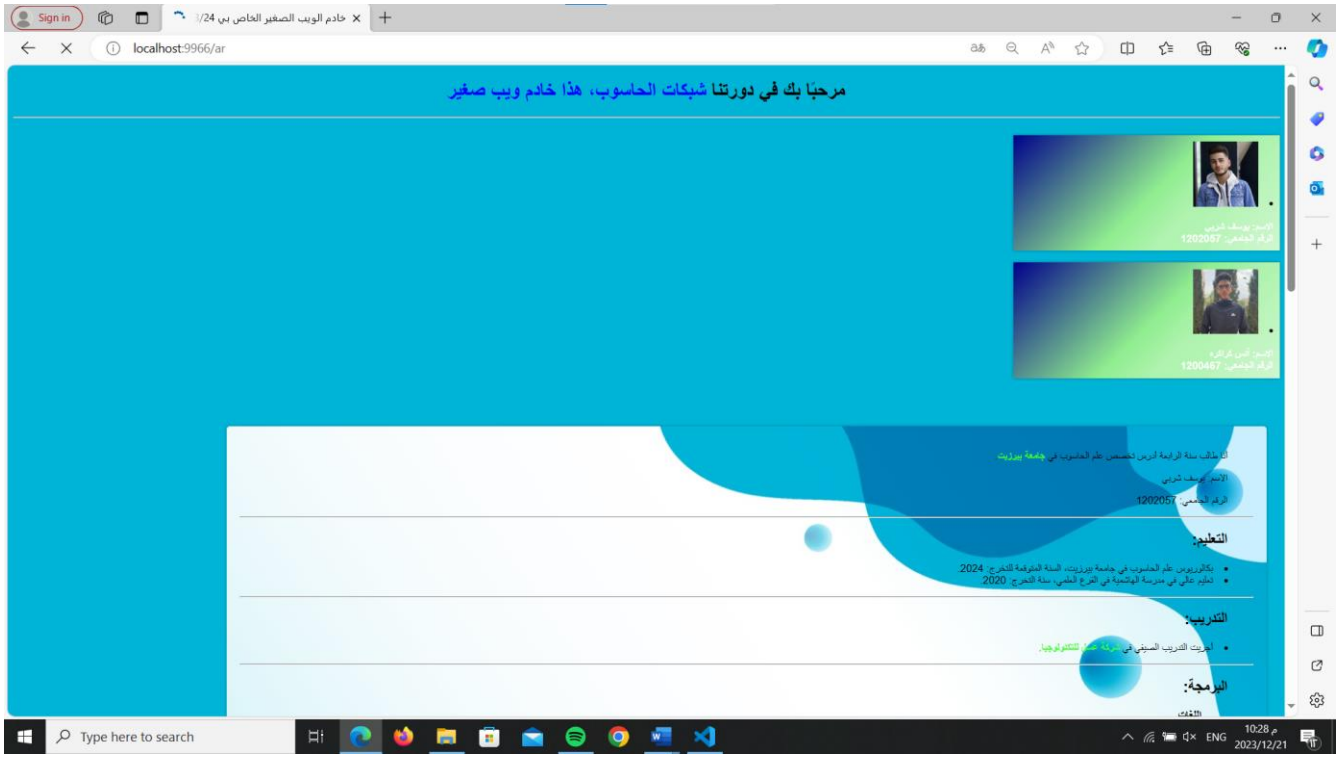


In case: the request is **/ar**

```
elif rec=='ar':
    # Send one HTTP header line into socket
    connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
    connectionSocket.send("Content-Type: text/html \r\n".encode())
    connectionSocket.send("\r\n".encode())
    htmlFile = open("main_ar.html",'rb')
    connectionSocket.send(htmlFile.read())
```

Output:

```
ar HTTP
********************
The Request
ar
********************
('127.0.0.1', 53352)
GET /styles.css HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:9966/ar
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

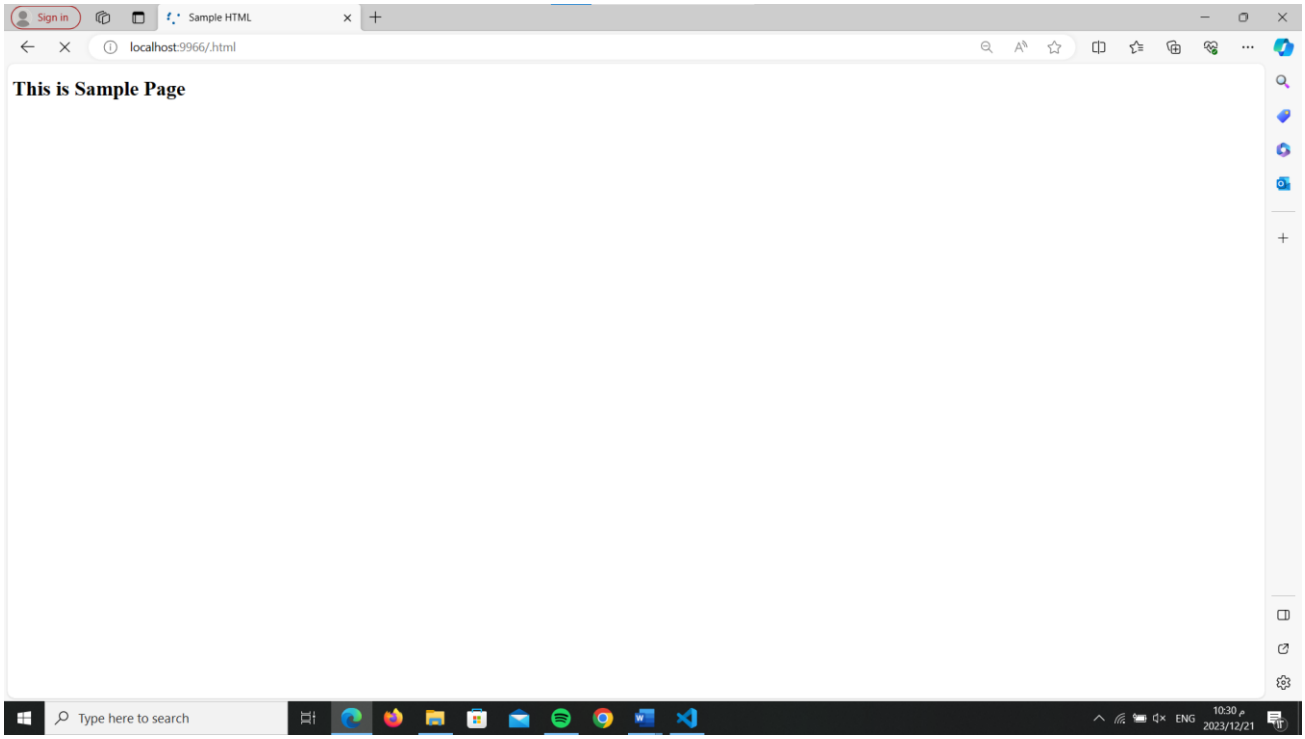


In case: the request is .**html**

```
elif rec=='.html':
    connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
    connectionSocket.send("Content-Type: text/html \r\n".encode())
    connectionSocket.send("\r\n".encode())
    htmlFile = open("sample_html.html",'rb')
    connectionSocket.send(htmlFile.read())
```

Output:

```
.html HTTP
********************
The Request
.html
********************
('127.0.0.1', 54226)
GET /favicon.ico HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:9966/.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

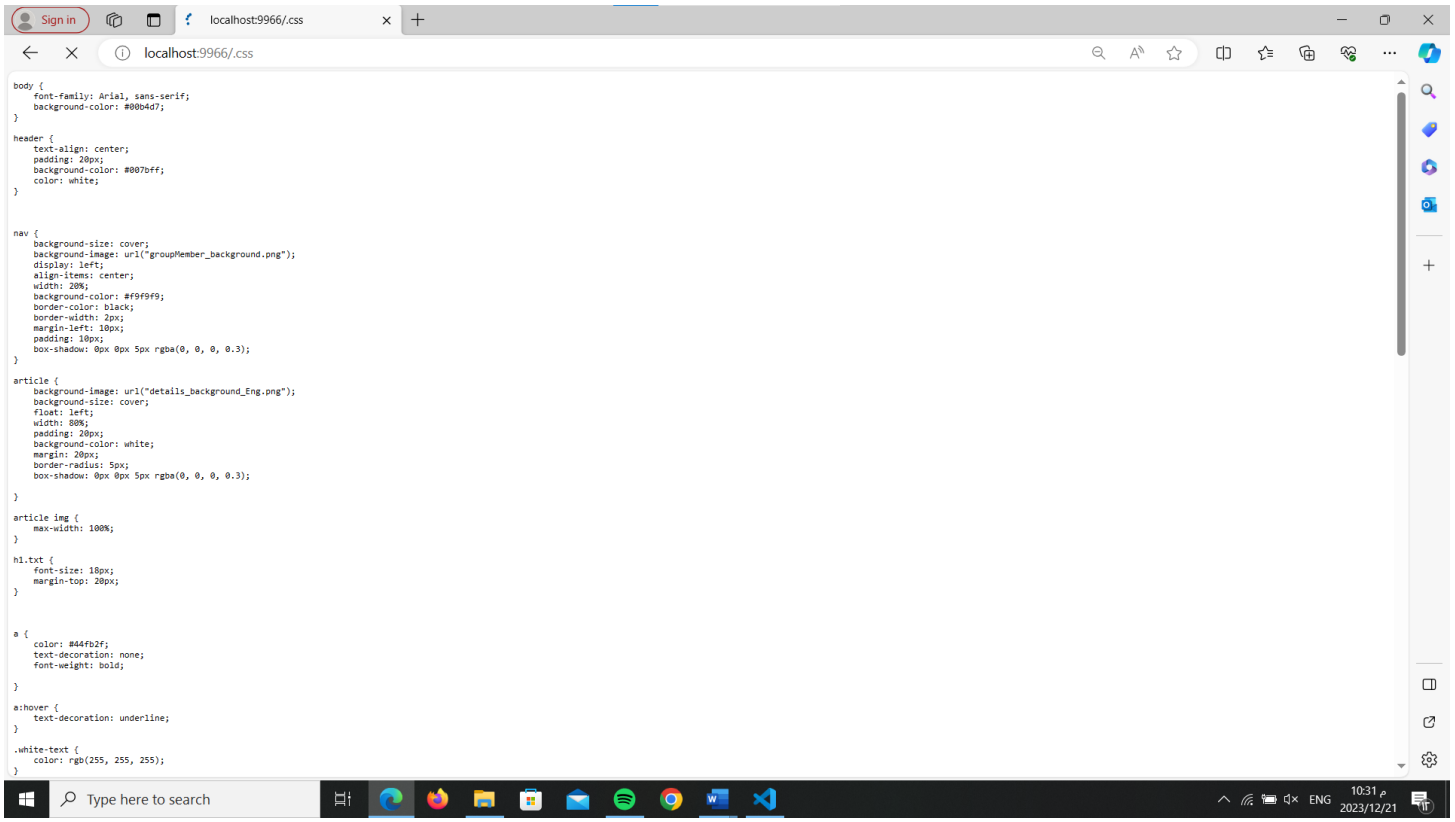


In case: the request is **.css**

```
elif rec=='.css':
    connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
    connectionSocket.send("Content-Type: text/css \r\n".encode())
    connectionSocket.send("\r\n".encode())
    cssFile = open("styles.css",'rb')
    connectionSocket.send(cssFile.read())
```

Output:

```
.css HTTP
********************
The Request
.css
********************
('127.0.0.1', 54010)
GET /favicon.ico HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:9966/.css
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

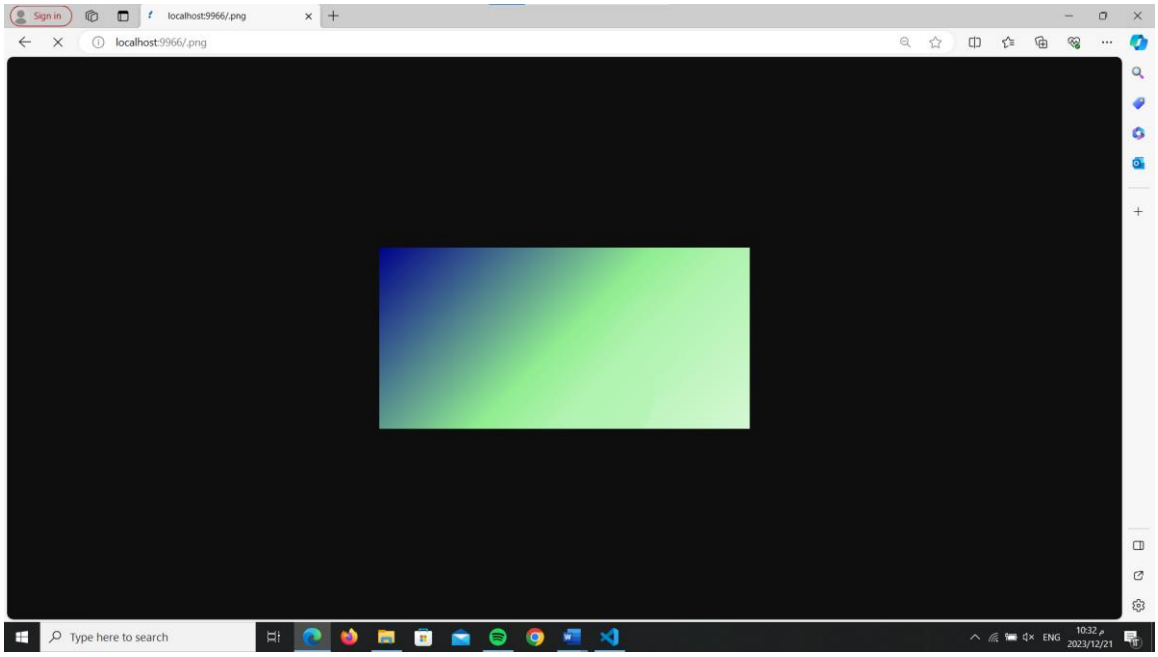


In case: the request is **.png**

```
elif rec=='.png':
    connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
    connectionSocket.send("Content-Type: image/png \r\n".encode())
    connectionSocket.send("\r\n".encode())
    pngImage = open("groupMember_background.png",'rb')
    connectionSocket.send(pngImage.read())
```

Output:

```
.png HTTP
********************
The Request
.png
********************
('127.0.0.1', 54575)
GET /favicon.ico HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:9966/.png
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

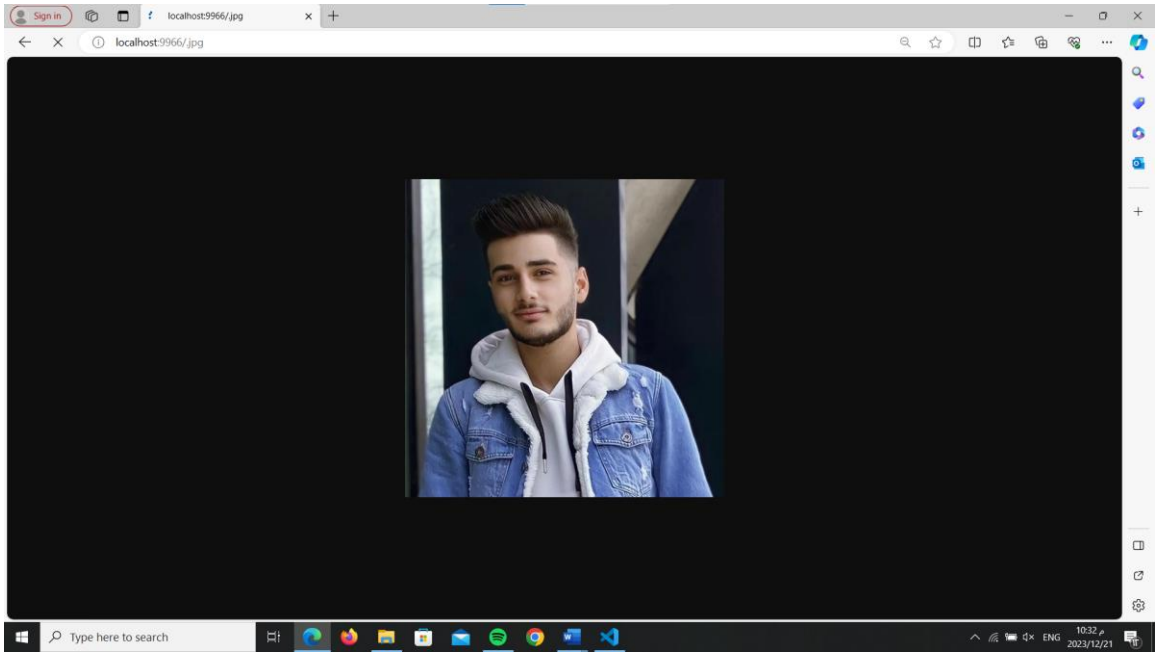


In case: the request is **.jpg**

```
elif rec=='.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())
    jpgImage = open("yousef.jpg",'rb')
    connectionSocket.send(jpgImage.read())
```

Output:

```
.jpg HTTP
********************
The Request
.jpg
********************
('127.0.0.1', 54839)
GET /favicon.ico HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:9966/.jpg
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```



In case: the request is **/cr or /so or /rt** using the status code **307 Temporary Redirect**

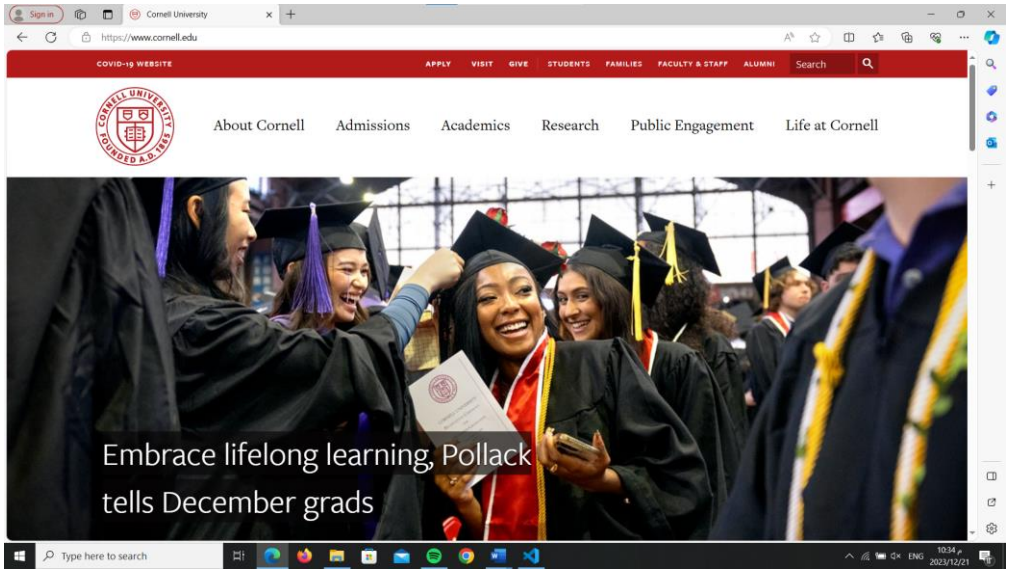
```
elif rec== 'cr':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())
    connectionSocket.send("Location: https://www.cornell.edu \r\n".encode())
    connectionSocket.send("Content-Type: text/html\r\n".encode())
    connectionSocket.send("\r\n".encode())
    connectionSocket.send("<html><body>Redirecting to Cornell...</body></html>".encode())
    connectionSocket.close()

elif rec== 'so':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())
    connectionSocket.send("Location: https://stackoverflow.com \r\n".encode())
    connectionSocket.send("Content-Type: text/html\r\n".encode())
    connectionSocket.send("\r\n".encode())
    connectionSocket.send("<html><body>Redirecting to Stack OverFlow...</body></html>".encode())
    connectionSocket.close()

elif rec== 'rt':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())
    connectionSocket.send("Location: https://ritaj.birzeit.edu \r\n".encode())
    connectionSocket.send("Content-Type: text/html\r\n".encode())
    connectionSocket.send("\r\n".encode())
    connectionSocket.send("<html><body>Redirecting to Ritaj...</body></html>".encode())
    connectionSocket.close()
```

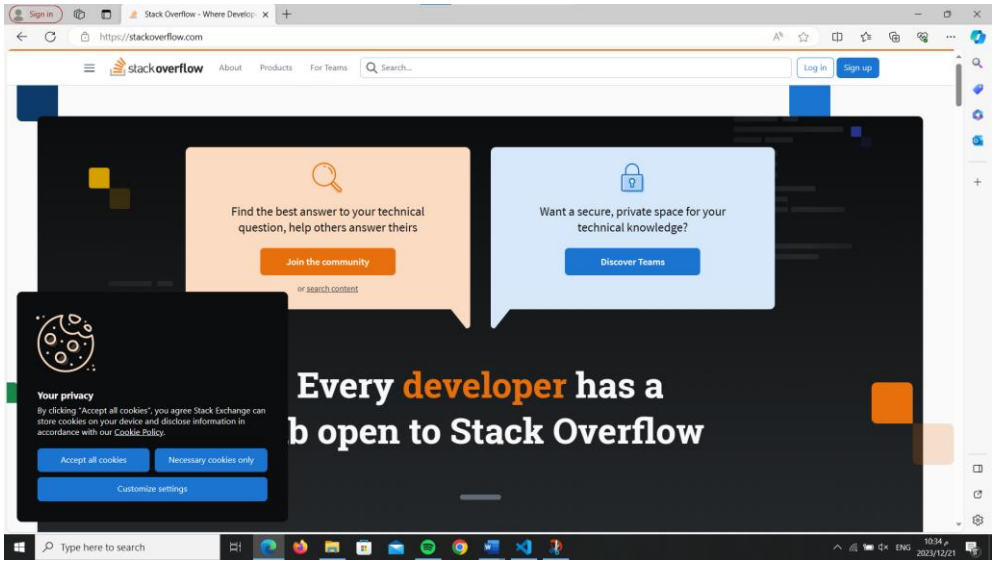
Output:

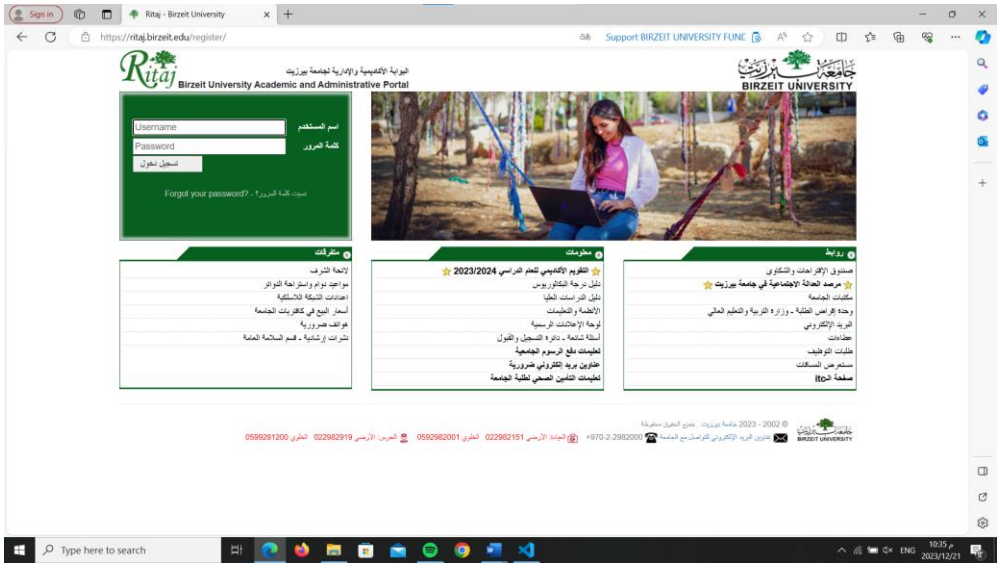
```
cr HTTP
********************
The Request
cr
********************
('127.0.0.1', 55228)
GET /so HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
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
```

```
so HTTP
********************
The Request
so
********************
('127.0.0.1', 55233)
GET /rt HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
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/120.0.0.0 Safari/537.36 Edg/120.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
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
```

```
PS C:\Users\twitter\OneDrive\Desktop\NetworkProj1_group46\Part3_Code> python -u "c:\Users\twitter\OneDrive\Desktop\NetworkProj1_group46\Part3_Code\server.py"
The server is listen on port 9966
('127.0.0.1', 55668)
GET /rt HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
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/120.0.0.0 Safari/537.36 Edg/120.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
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


rt HTTP
********************
The Request
rt
********************
```


Cornell University
https://www.cornell.edu
COVID-19 WEBSITE
APPLY
VISIT
GIVE
STUDENTS
FAMILIES
FACULTY & STAFF
ALUMNI
Search
About Cornell
Admissions
Academics
Research
Public Engagement
Life at Cornell
Embrace lifelong learning, Pollack tells December grads
Type here to search


Stack Overflow - Where Develop
https://stackoverflow.com
stackoverflow
About
Products
For Teams
Search...
Log in
Sign up
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
Your privacy
By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.
Accept all cookies
Necessary cookies only
Customize settings
Every developer has a
b open to Stack Overflow
Type here to search


Ritaj - Birzeit University
https://ritaj.birzeit.edu/register/
Support BIRZEIT UNIVERSITY FUNC
Ritaj
البوابة الأكاديمية والإدارية لجامعة بيرزيت
Birzeit University Academic and Administrative Portal
BIRZEIT UNIVERSITY
Username
اسم المستخدم
Password
كلمة المرور
تسجيل دخول
Forgot your password? - نسيت كلمة المرور؟
روابط
معلومات
متفرقات
Type here to search

In case: the request is wrong or the file doesn’t exist

```
else:
    connectionSocket.send("HTTP/1.1 404 Not Found\r\n".encode())
    connectionSocket.send("Content-Type: text/html \r\n".encode())
    connectionSocket.send("\r\n".encode())
    s="<html> <head> <title>Error 404</title> </head> <body> <h1>Error 404</h1> <p style='color: red;'> The file is not found</p> <br> <p>Name: <strong><bold>Yousef Sharbi</bold></
    connectionSocket.send(s.encode())
```

Output:

```
sssssssssssssssssss HTTP
********************
The Request
sssssssssssssssssss
********************
('127.0.0.1', 55968)
GET /favicon.ico HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Microsoft Edge";v="120"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:9966/sssssssssssssssssss
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```