

BIRZEIT UNIVERSITY

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

---

**Prepared by:**

Saja Asfour            1210737  
Rouand Bawatneh 1211403  
Shahd Shreteh        1210444

---

**Instructor:**

Dr. Abd Alkarim Awad , Dr. Mohammad Jubran

**Section:**

Saja &rouand sec 3  
Shahd sec2

**Abstract:**

This project consists of three parts. The first part is to run some commands on the terminal; ping a device in the same network, ping www.cornell.edu website, tracert and nslookup it. The second part is implementing server and client application both for TCP : The server waits for a message from a client and if the messeges is with one of our IDs , the server do some instruction .The last part is to implement a simple but a complete web server in python that is listening on port 9966. Both html and css are used for the design of our website.

## Table of Contents

<b>Abstract.....</b>	1
<b>Table Of Figurs .....</b>	3
<b>Part 1.....</b>	5
<b>What are ping , tracert,nslookup.....</b>	5
<b>Ping a device in the same network .....</b>	5
<b>Ping www.cornell.edu .....</b>	6
<b>tracert www.cornell.edu .....</b>	6
<b>nslookup www.cornell.edu .....</b>	7
<b>Use wireshark to capture some DNS message.....</b>	8
<b>Part 2: Implement a server and client application.....</b>	9
<b>Codes.....</b>	9
<b>Server code .....</b>	9
<b>Client code .....</b>	11
<b>Part 3 : .....</b>	14
<b>content in HTTP request for rfce2616 and Why do we need Content_Type .....</b>	14
<b>If the request is / or /index.html or /main_en.html or /en .....</b>	15
<b>Screenshot from the same computer.....</b>	16
<b>Screenshot from IPHONE.....</b>	28
<b>The Html file.....</b>	35
<b>The html code(for local html ) .....</b>	38
<b>If the request is /ar.....</b>	39
<b>Test from the same computer.....</b>	40
<b>The screenshot from iphone.....</b>	46
<b>The HTML file.....</b>	52
<b>If the request is an .html file.....</b>	55
<b>If the request is an .css file.....</b>	58
<b>If the request is an .png file.....</b>	59
<b>If the request is an .jpg file.....</b>	60
<b>If the request is an /cr file.....</b>	61
<b>If the request is an ./so file.....</b>	63
<b>If the request is an /rt file.....</b>	64
<b>The request is wrong or the file doesn't exist.....</b>	65
<b>Python code.....</b>	67
<b>Conclusion.....</b>	70

## Table of Figures

Figure 1: command for ping a device .....	5
Figure 2: ping www.cornell.edu .....	6
Figure 3: tracert www.cornell.edu .....	6
Figure 4: nslookup www.cornell.edu .....	7
Figure 5: some DNS messages from wireshark .....	8
Figure 6: IP address 13.107.246.45 .....	8
Figure 7: server code.....	9
Figure 8: client code.....	11
Figure 9: client output for first id.....	11
Figure 10: server output for first id.....	11
Figure 11: client output for second id .....	<b>Error! Bookmark not defined.</b>
Figure 12: server output for second id .....	<b>Error! Bookmark not defined.</b>
Figure 13: client output for last id.....	12
Figure 14: server output for last id.....	13
Figure 15: client output for wrong id.....	13
Figure 16: server output for wrong id .....	13
Figure 17: content_type for rfce2616 .....	14
Figure 18: code if request file / or /en or /index.html or /main_en.html.....	16
Figure 19: The result of request for / .....	16
Figure 20: The result of request for /en .....	18
Figure 21: The result of request for /indix.html.....	20
Figure 22: The result of request for /main_en.html .....	22
Figure 23: the html page of link.....	24
Figure 24: the output on the command .....	25
Figure 25: the output when request message is /main_en.html .....	26
Figure 26:page in IPHONE if request is / or /index.html or /main_en.html or /en .....	28
Figure 27: output in terminal when use IPHONE if request is / or /index.html or /main_en.html or /en.....	29
Figure 28: page in android if request is / or /index.html or /main_en.html or /en.....	31
Figure 29: output in terminal when use android if request is / or /index.html or /main_en.html or /en .....	33
Figure 30: main_en.html code .....	35
Figure 31: styles .css code .....	37
Figure 32: page .html code.....	38
Figure 33: code when request is /ar .....	39
Figure 34: the result of the request /ar from the same computer .....	40
Figure 35: the output result of the request /ar from the same computer .....	42
Figure 36: page when click the local html link.....	44
Figure 37: page when click the external link .....	45
Figure 38: output when click the link .....	45
Figure 39: the page when request /ar from IPHONE.....	46
Figure 40: the output when request /ar from IPHONE .....	47
Figure 41: the page when click local html link from IPHONE .....	49
Figure 42: the output when click local html link from IPHONE .....	50

Figure 43: the output when click external link from IPHONE.....	51
Figure 44: main_ar.html code.....	52
Figure 45: page1.html code.....	54
Figure 46: the code when request in an .html file.....	55
Figure 47:the page when request is page.html.....	55
Figure 48:the output when request is an .html file.....	56
Figure 49:the code when the request in an .css file.....	58
Figure 50:the page when request is styels.css file.....	58
Figure 51:the output when request is an styels.css file.....	59
Figure 52: the code when request is an png file.....	59
Figure 53: the page when request is sh.png.....	59
Figure 54: the output when request is an .png file .....	60
Figure 55: the code when request is an .jpg file.....	60
Figure 56: the page when request is s.jpg.....	60
Figure 57: the output when request is an .jpg file.....	61
Figure 58: the code when request is /cr.....	61
Figure 59: the page when request is /cr.....	62
Figure 60: the output when request is /cr.....	62
Figure 61: the code when request is /so.....	63
Figure 62: the page when request is /so.....	63
Figure 63: the output when request is /so.....	63
Figure 64: the code when request is /rt.....	64
Figure 65: the page when request is /rt.....	64
Figure 66: the output when request is /rt.....	64
Figure 67: the code when request is wrong or request does not exist.....	65
Figure 68: the page when request is wrong or request does not exist.....	65
Figure 69: the output when request is wrong or request does not exist.....	66
Figure 70: the python code for part three.....	67

## Part one:

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

**Ping**: is a command used to check the connectivity of the network and it is used to measure the round-trip time(RTT) between devices by sending a small data packet.

**Tracert**: command used to trace the network path and measure latency at each hop (routers and intermediate devices in the path).

**Nslookup**: is a program that allows the user to enter the hostname and find the IP address or DNS, also by using a specific command can do reverse DNS lookups (find the hostname of the IP address).

**telnet**: a protocol that allows the computer to establish a command-line connection to a remote device over the network.

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

1-Ping a device in the same network, e.g. from a laptop to a smartphone

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>ping 192.168.1.7

Pinging 192.168.1.7 with 32 bytes of data:
Reply from 192.168.1.7: bytes=32 time=114ms TTL=64
Reply from 192.168.1.7: bytes=32 time=12ms TTL=64
Reply from 192.168.1.7: bytes=32 time=31ms TTL=64
Reply from 192.168.1.7: bytes=32 time=56ms TTL=64

Ping statistics for 192.168.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 12ms, Maximum = 114ms, Average = 51ms

C:\Users\DELL>
```

Figure 1:command for ping a device

## 2-Ping [www.cornell.edu](http://www.cornell.edu)

```
C:\Users\DELL>ping 192.168.1.7

Pinging 192.168.1.7 with 32 bytes of data:
Reply from 192.168.1.7: bytes=32 time=114ms TTL=64
Reply from 192.168.1.7: bytes=32 time=12ms TTL=64
Reply from 192.168.1.7: bytes=32 time=31ms TTL=64
Reply from 192.168.1.7: bytes=32 time=50ms TTL=64

Ping statistics for 192.168.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 12ms, Maximum = 114ms, Average = 51ms

C:\Users\DELL>ping www.cornell.edu

Pinging part-0017.t-0009.t-msedge.net [13.107.246.45] with 32 bytes of data:
Reply from 13.107.246.45: bytes=32 time=46ms TTL=118
Reply from 13.107.246.45: bytes=32 time=62ms TTL=118
Reply from 13.107.246.45: bytes=32 time=46ms TTL=118
Reply from 13.107.246.45: bytes=32 time=47ms TTL=118

Ping statistics for 13.107.246.45:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 46ms, Maximum = 62ms, Average = 50ms

C:\Users\DELL>
```

Figure2: ping www.cornell.edu

This is sent to check if the device is existed or not , so the reply message is to inform that the device exists and connects to the network .Also , it show the round trip time(RTT) of the packets and there ant losses along the way.

TTL: time to live (numbers of hops the packet passes along the path)

## 3-traceroute [www.cornell.edu](http://www.cornell.edu)

```
C:\Users\DELL>ping www.cornell.edu

Pinging part-0017.t-0009.t-msedge.net [13.107.246.45] with 32 bytes of data:
Reply from 13.107.246.45: bytes=32 time=46ms TTL=118
Reply from 13.107.246.45: bytes=32 time=62ms TTL=118
Reply from 13.107.246.45: bytes=32 time=46ms TTL=118
Reply from 13.107.246.45: bytes=32 time=47ms TTL=118

Ping statistics for 13.107.246.45:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 46ms, Maximum = 62ms, Average = 50ms

C:\Users\DELL>tracert www.cornell.edu

Tracing route to part-0017.t-0009.t-msedge.net [13.107.213.45]
over a maximum of 30 hops:

 1   5 ms    14 ms    7 ms  192.168.1.1
 2   6 ms    4 ms    26 ms  10.74.32.246
 3   5 ms    5 ms    *      10.74.19.113
 4   49 ms   49 ms   48 ms  ae61-0.ieer02.tlv30.ntwk.msn.net [104.44.36.229]
 5   49 ms   46 ms   46 ms  13.104.140.42
 6   *       *       *       Request timed out.
 7   47 ms   49 ms   70 ms  13.107.213.45

Trace complete.

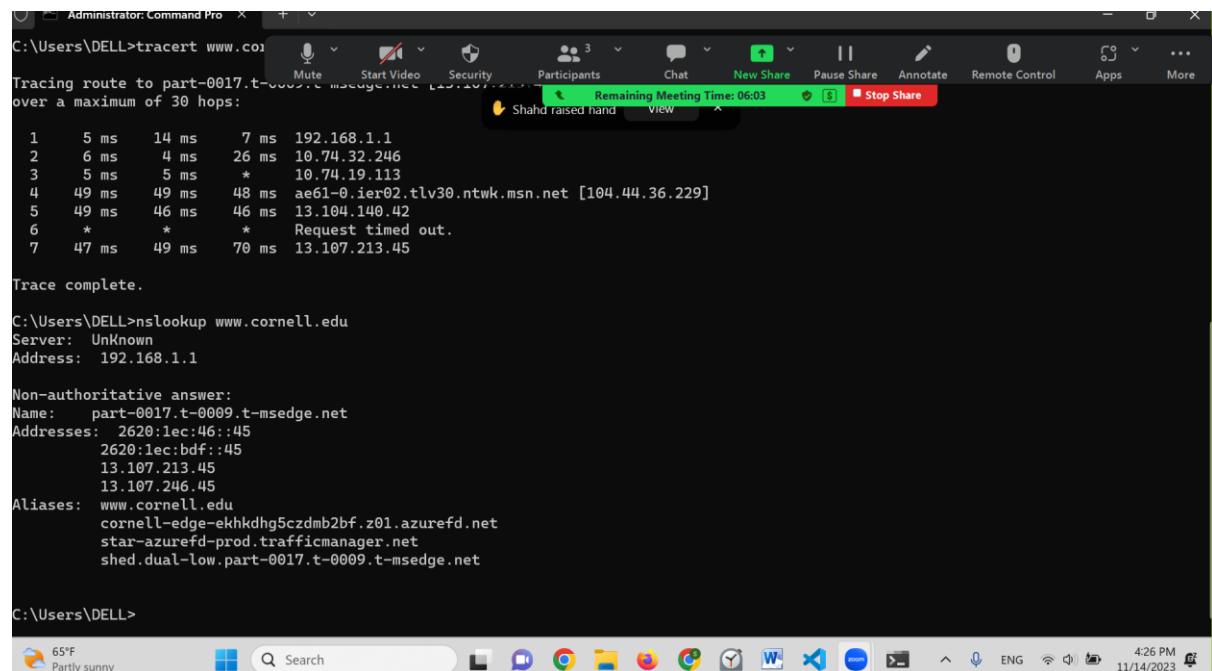
C:\Users\DELL>
```

Figure3: tracert www.cornell.edu

Tracert shows the IP address of the destination (13.107.213.45) and lists all hops along the path , the first one is 192.168.1.1 then 10.74.32.246 then 10.74.19.113 then 104.44.36.229 then 13.104.140.42 then we have request timed out with measurements

are \*, then the packet is prevented from reaching the destination because there is a problem at the location or the router is incorrect , and the last IP address is the host. When we go down in the lines , we will see the 3 measurements increases because the router go further.

4-nslookup [www.cornell.edu](http://www.cornell.edu)



```
C:\Users\DELL>tracert www.co
Tracing route to part-0017.t-0009.t-msedge.net [192.168.1.1]
over a maximum of 30 hops:
 1  5 ms    14 ms    7 ms  192.168.1.1
 2  6 ms    4 ms    26 ms  10.74.32.246
 3  5 ms    5 ms    *     10.74.19.113
 4  49 ms   49 ms   48 ms  ae61-0.iern02.tlv30.ntwk.msn.net [104.44.36.229]
 5  49 ms   46 ms   46 ms  13.104.140.42
 6  *       *       *     Request timed out.
 7  47 ms   49 ms   70 ms  13.107.213.45

Trace complete.

C:\Users\DELL>nslookup www.cornell.edu
Server:  UnKnown
Address: 192.168.1.1

Non-authoritative answer:
Name:  part-0017.t-0009.t-msedge.net
Addresses:  2620:lec:46::45
           2620:lec:bdf::45
           13.107.213.45
           13.107.246.45
Aliases:  www.cornell.edu
          cornell-edge-ekhkdhg5czdmb2bf.z01.azurefd.net
          star-azurefd-prod.trafficmanager.net
          shed.dual-low.part-0017.t-0009.t-msedge.net

C:\Users\DELL>
```

Figure4: nslookup www.cornell.edu

The IP address of the user (192.168.1.1)

The addresses that come back are for cornell.edu.

So the user sent the hostname and nslookup response has the IP addresses of that host.

Aliases : Additional names associated with the resolved IP addresses. In this case, "[www.cornell.edu](http://www.cornell.edu)" is an alias for the resolved IP addresses, and there are other aliases such as "cornell-edge-ekhkdhg5czdmb2bf.z01.azurefd.net" and "shed.dual-low.part-0017.t-0009.t-msedge.net."

### 3- use wireshark to capture some DNS messages.

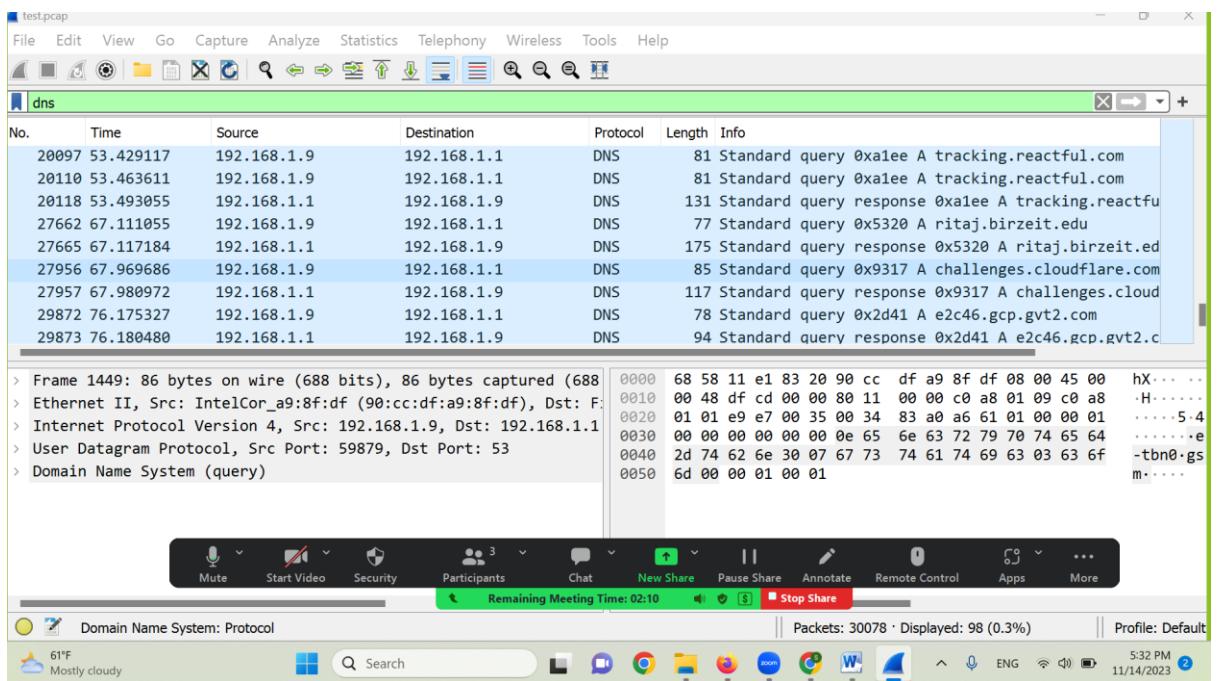


Figure 5: some DNS messages from wireshark

From the ping results, do you think the response you have got is from USA? Explain your answer briefly.

### IP information 13.107.246.45

IP address	13.107.246.45
Location	United States (US)
Registry	arin

### Hosting information

This extension shows detailed info about your IP, your domain and your provider. Get an IP change alert and see your IP history.

Your public IP Address  
Information about your public IP  
Address like reverse dns, hosting info and whois data.

Chrome Extension  
Our data and tools directly inside your Chrome browser.

My IP Address for Chrome  
This extension shows detailed info about your IP, your domain and your provider. Get an IP change alert and see your IP history.

IP Domain Flag for Chrome  
This extension shows detailed info about your IP, your domain and your provider. Get an IP change alert and see your IP history.

Figure 6: IP address 13.107.246.45

Yes, based on the information , it appears that the responses are from the United States. The IP address 13.107.246.45, which is associated with the domain "www.cornell.edu" in my ping command, is owned by Microsoft (msedge.net). Microsoft's content delivery network (CDN) servers, including those used by Edge services, are distributed globally. The IP address 13.107.246.45 is one of Microsoft's servers, and the low round-trip times (ranging from 46ms to 62ms) in the ping responses suggest a relatively low latency, which is typical for servers located in the United State.

## Part two:

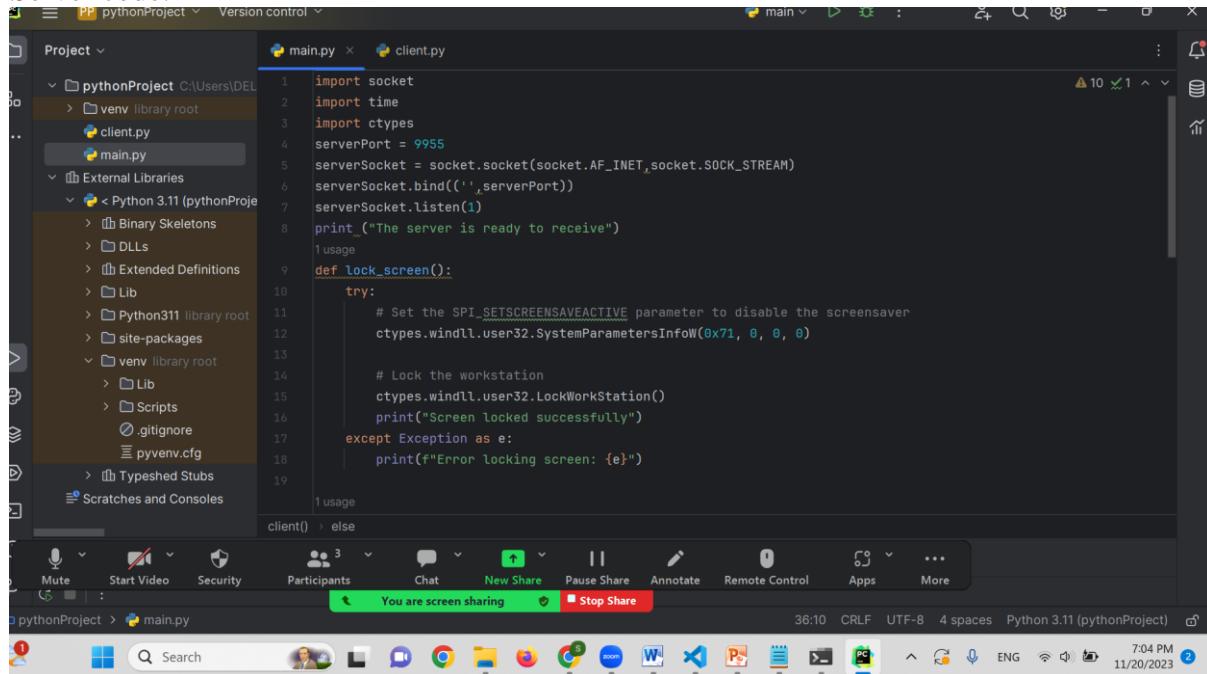
Using socket programming, implement TCP client and server applications in go, python, java or C. The server should listen on port 9955. The server waits for a message from a client. If the message is with one of the students ID, the sever should do the following:

1. display a message on the server side that the OS will lock screen after 10 seconds
2. send a message to the client that the sever will lock screen after 10 seconds
3. then wait 10 seconds
4. then call a function lock the screen of the operating system (windows or Linux or MAC).

Any student ID of the group member should work. Any other student number or any text should display an error message on the server side and no lock screen should be done.

## Codes:

### Server code:



The screenshot shows a Python IDE interface with the following details:

- Project View:** Shows a project named "pythonProject" containing files "client.py" and "main.py".
- Code Editor:** The "main.py" file is open, displaying the following code:

```
import socket
import time
import ctypes
serverPort = 9955
serverSocket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
serverSocket.bind(('',serverPort))
serverSocket.listen(1)
print("The server is ready to receive")
usage
def lock_screen():
    try:
        # Set the SPI_SETSCREENSAVEACTIVE parameter to disable the screensaver
        ctypes.windll.user32.SystemParametersInfoW(0x71, 0, 0, 0)

        # Lock the workstation
        ctypes.windll.user32.LockWorkStation()
        print("Screen locked successfully")
    except Exception as e:
        print(f"Error locking screen: {e}")
client() > else
```
- Toolbar:** Includes icons for Mute, Start Video, Security, Participants, Chat, New Share, Pause Share, Annotate, Remote Control, Apps, and More.
- Bottom Status Bar:** Shows "You are screen sharing" and "Stop Share" buttons, along with system information like "36:10 CRLF UTF-8 4 spaces Python 3.11 (pythonProject)" and a timestamp "7:04 PM 11/20/2023".

The screenshot shows a Python IDE interface with the project 'pythonProject' open. The left sidebar displays the project structure, including a 'venv' folder and 'External Libraries'. The main editor window shows the 'client.py' file with the following code:

```
main.py x client.py
18     print(f"Error locking screen: {e}")
19
20     def client(client_socket, data):
21         usage
22         ids = ["1210737", "1210444", "1211403"]
23         if data in ids:
24             print("It's valid student ID which is", data)
25
26             # Display message on the server side
27             print("The OS will lock screen after 10 seconds...")
28
29             # Send a message to the client
30             client_socket.send("Server will lock screen after 10 seconds".encode())
31
32             # Wait for 10 seconds
33             time.sleep(10)
34
35             # Call the function to lock the screen
36             lock_screen()
37
38         else:
39             print("Error: Invalid student ID ")
40
41     client() > else
```

The status bar at the bottom indicates 'You are screen sharing'.

The screenshot shows the same Python IDE interface with the project 'pythonProject' open. The left sidebar shows the project structure. The main editor window shows the 'main.py' file with the following code:

```
main.py x client.py
32     time.sleep(10)
33
34     # Call the function to lock the screen
35     lock_screen()
36
37     else:
38         print("Error: Invalid student ID ")
39
40     while True:
41         client_socket, addr = serverSocket.accept()
42         print("Accepted connection from", addr)
43
44         data = client_socket.recv(1024).decode()
45         client(client_socket, data)
46
47         client_socket.close()
```

The status bar at the bottom indicates 'You are screen sharing'.

Figure 7: server code

### Client code:

```
Project main.py client.py
pythonProject C:\Users\DELL\pythonProject\venv\lib\site-packages\client.py
1 import socket
2
3 message = input("Enter Valid ID")
4 serverName = socket.gethostname()
5
6 # Set up the client
7 client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8 client_socket.connect((serverName, 9955))
9
10 # Send the student ID to the server
11 client_socket.send(message.encode())
12
13 # Receive the response from the server
14 response = client_socket.recv(1024).decode()
15 # Close the connection
16 client_socket.close()
```

Figure 8:client code

We check our program with our id number and another id number :

This for our id number:

→first id 1210737

```
Run main client
C:\Users\DELL\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\DELL\PycharmProjects\pythonProject\client.py
Enter Valid ID1210737
Process finished with exit code 0
```

Figure 9:client output for first id

And the screen is look after 10 sec

```
Run main client
C:\Users\DELL\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\DELL\PycharmProjects\pythonProject\main.py
The server is ready to receive
Accepted connection from ('192.168.1.11', 59670)
It's valid student ID which is 1210737
The OS will lock screen after 10 seconds...
Screen locked successfully
```

Figure 10:server output for first id

→the second id is 1210444

The screenshot shows the PyCharm terminal window with two tabs: 'Run' and 'client'. The 'client' tab is active, displaying the command 'python.exe C:/Users/DELL/PycharmProjects/pythonProject/venv/Scripts/python.exe C:/Users/DELL/PycharmProjects/pythonProject/client.py' and the response 'Enter Valid ID1210444'. Below the command, it says 'Process finished with exit code 0'. The status bar at the bottom indicates Python 3.11 (pythonProject) and the date/time 11/20/2023 7:13 PM.

Figure 11:client output for second id

And the screen is look after 10 sec

The screenshot shows the PyCharm terminal window with two tabs: 'Run' and 'client'. The 'client' tab is active, displaying the command 'python.exe C:/Users/DELL/PycharmProjects/pythonProject/venv/Scripts/python.exe C:/Users/DELL/PycharmProjects/pythonProject/main.py'. The server's response includes: 'The server is ready to receive', 'Accepted connection from ('192.168.1.11', 59670)', 'It's valid student ID which is 1210737', 'The OS will lock screen after 10 seconds...', 'Screen locked successfully', 'Accepted connection from ('192.168.1.11', 59693)', 'It's valid student ID which is 1210444', 'The OS will lock screen after 10 seconds...', and 'Screen locked successfully'. The status bar at the bottom indicates Python 3.11 (pythonProject) and the date/time 11/20/2023 7:13 PM.

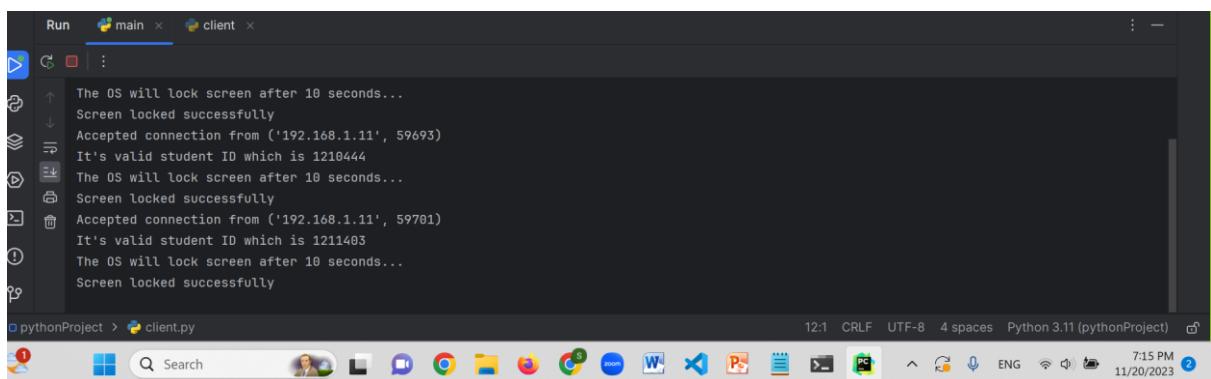
Figure 12:server output for second id

→the last id is 1211403

The screenshot shows the PyCharm terminal window with two tabs: 'Run' and 'client'. The 'client' tab is active, displaying the command 'python.exe C:/Users/DELL/PycharmProjects/pythonProject/venv/Scripts/python.exe C:/Users/DELL/PycharmProjects/pythonProject/client.py' and the response 'Enter Valid ID1211403'. Below the command, it says 'Process finished with exit code 0'. The status bar at the bottom indicates Python 3.11 (pythonProject) and the date/time 11/20/2023 7:14 PM.

Figure 13:client output for last id

And the screen is look after 10 sec



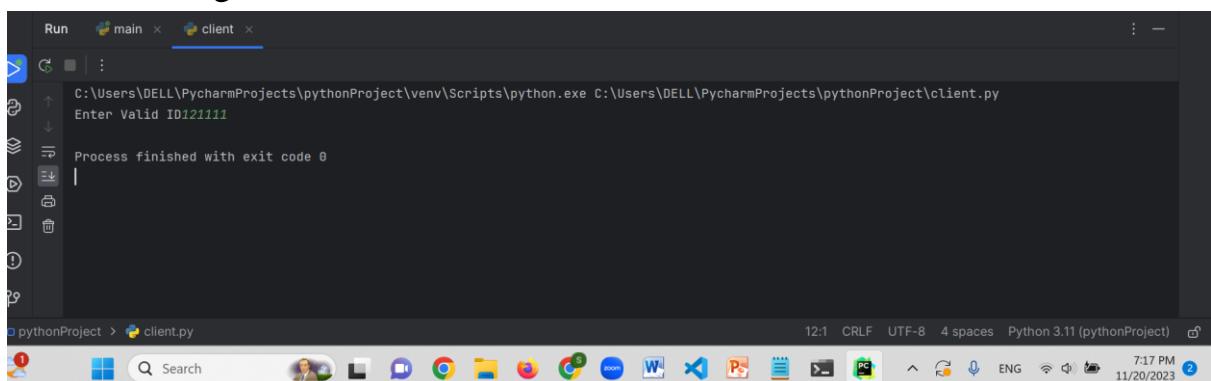
The screenshot shows the PyCharm IDE interface with two tabs open: 'Run' and 'client'. The 'client' tab is active, displaying the following text in the terminal window:

```
The OS will lock screen after 10 seconds...
Screen locked successfully
Accepted connection from ('192.168.1.11', 59693)
It's valid student ID which is 1210444
The OS will lock screen after 10 seconds...
Screen locked successfully
Accepted connection from ('192.168.1.11', 59701)
It's valid student ID which is 1211403
The OS will lock screen after 10 seconds...
Screen locked successfully
```

The status bar at the bottom indicates Python 3.11 (pythonProject) and the current date and time as 11/20/2023 7:15 PM.

Figure 14:server output for last id

This for wrong ids:



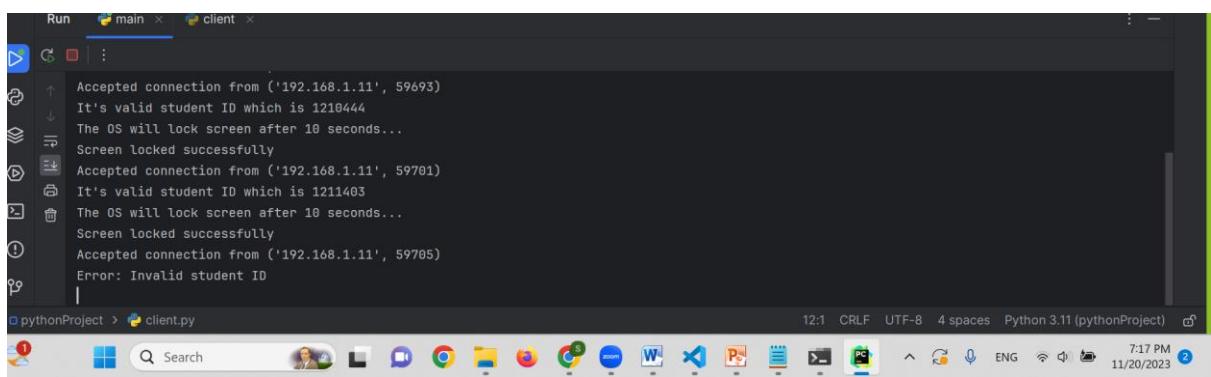
The screenshot shows the PyCharm IDE interface with two tabs open: 'Run' and 'client'. The 'client' tab is active, displaying the following text in the terminal window:

```
C:\Users\DELL\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\DELL\PycharmProjects\pythonProject\client.py
Enter Valid ID121111
Process finished with exit code 0
```

The status bar at the bottom indicates Python 3.11 (pythonProject) and the current date and time as 11/20/2023 7:17 PM.

Figure 15:client output for wrong id

And the screen not looked



The screenshot shows the PyCharm IDE interface with two tabs open: 'Run' and 'client'. The 'client' tab is active, displaying the following text in the terminal window:

```
Accepted connection from ('192.168.1.11', 59693)
It's valid student ID which is 1210444
The OS will lock screen after 10 seconds...
Screen locked successfully
Accepted connection from ('192.168.1.11', 59701)
It's valid student ID which is 1211403
The OS will lock screen after 10 seconds...
Screen locked successfully
Accepted connection from ('192.168.1.11', 59705)
Error: Invalid student ID
```

The status bar at the bottom indicates Python 3.11 (pythonProject) and the current date and time as 11/20/2023 7:17 PM.

Figure 16:server output for wrong id

### Part three:

0- from rfce2616, what is Content-Type in the HTTP request and why do we need it?

Content-Type:

text/html; charset=utf-8

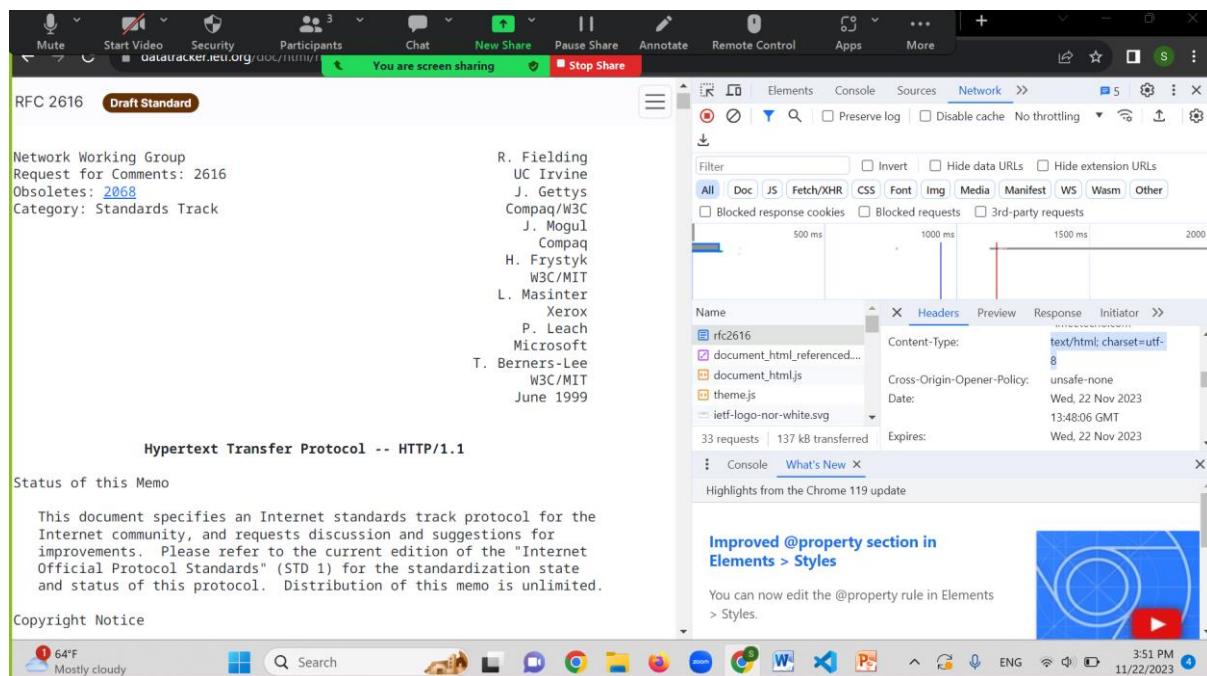


Figure 17:content\_type for rfce2616

Why do we need Content\_Type?

The Content-Type header in HTTP requests and responses serves a crucial role in facilitating proper communication between clients and servers by indicating the media type (MIME type) of the content being sent or received.

the Content-Type header is a fundamental part of the HTTP protocol, playing a key role in data interpretation, negotiation, error handling, security, and proper rendering of content. It helps maintain interoperability and ensures that data is processed correctly between different systems on the web.

Starting the process, a server socket is created, specifying a particular port number (9966). The server socket actively waits for a TCP connection from a singular client.

When a client requests a connection with the server, the server responds by creating a connection socket to facilitate communication. It also captures the address of the client. Subsequently server receives the client's request, enabling further interaction between the two entities.

After examining the URL path, the server proceeds to provide the appropriate content to the client based on the following conditions:

1- if the request is / or /index.html or /main\_en.html or /en (for example localhost:9966/ or localhost:9966/en) then the server should send main\_en.html file with Content-Type: text/html.

The main\_en.html file should contain

HTML webpage that contains

- a. "ENCS3320-My Tiny Webserver 23/24" in the title
- b. "Welcome to our course **Computer Networks, This is a tiny webserver**" (part of the phrase is in **Blue**)
- c. Use CSS to make the page looks nice
- d. Divide the page in different boxes and put student's information in the different boxes
- e. Include CSS as a separate file
- f. Summarize point 0 above in a box
- g. Group members names and IDs (each one in a box)
- h. Some information about the group members. For instance, projects you have done during different course (programming, electrical, math, etc), skills, hobbies, etc.
- i. The page should contain at least an image with extention.jpg and an image with extension .png
- j. A link to a local html file (an html file)
- k. a link to [https://www.w3schools.com/python/python\\_strings.asp](https://www.w3schools.com/python/python_strings.asp)

The connection socket sends a response with "HTTP/1.1 200 OK" to inform the client that the requested object has been found and everything is proceeding smoothly. Subsequently, the socket transmits an HTML file with the content type set as text/html, concluding the response with "\r\n" to signify its completion. The HTML file is then opened and read.

```

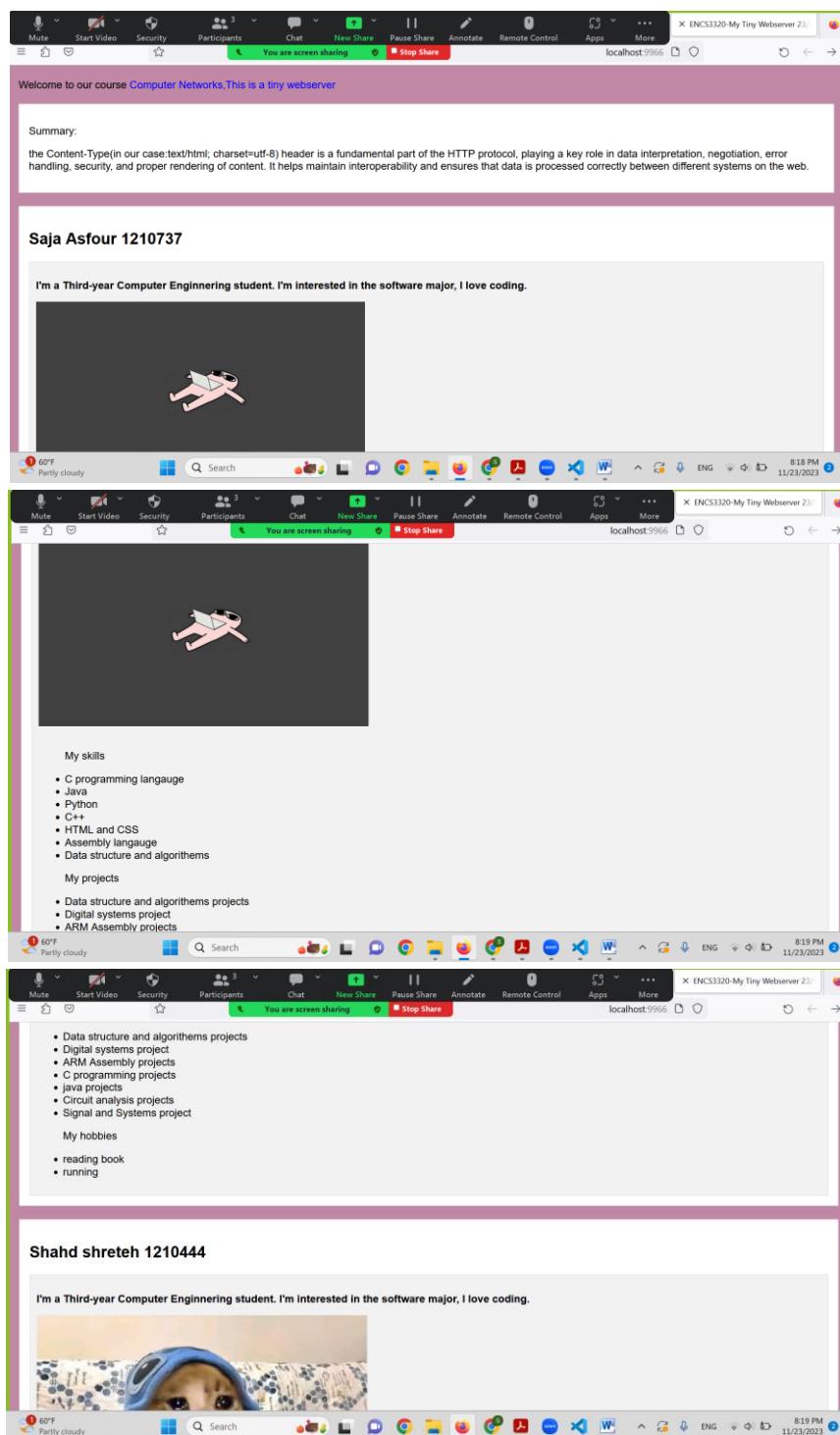
38     if request_File == '' or request_File=='main_en.html' or request_File=='index.html' or request_File =='en':
39         connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
40         connectionSocket.send(f"Content-Type: text/html \r\n".encode())
41         connectionSocket.send(f"\r\n".encode())
42         mhtml=open('main_en.html','rb')
43         connectionSocket.send(mhtml.read())
44         mhtml.close()

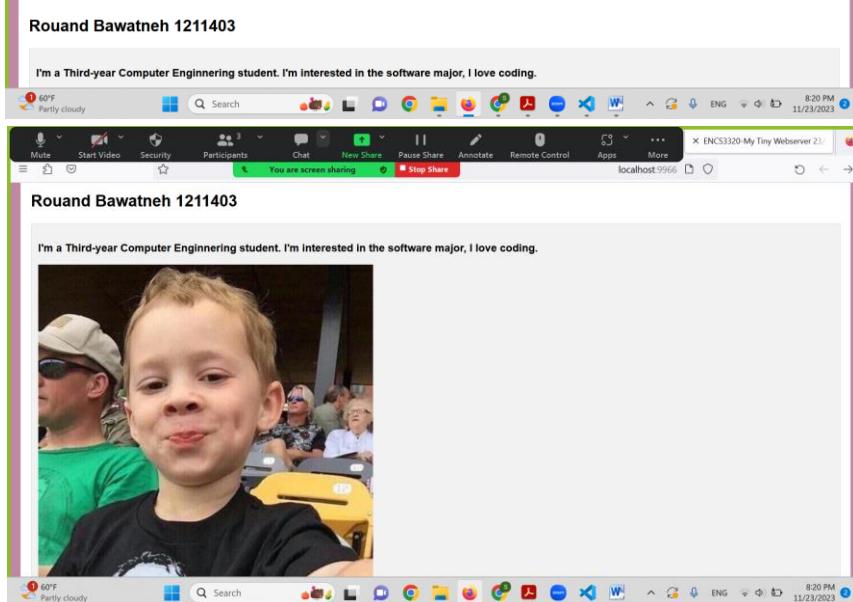
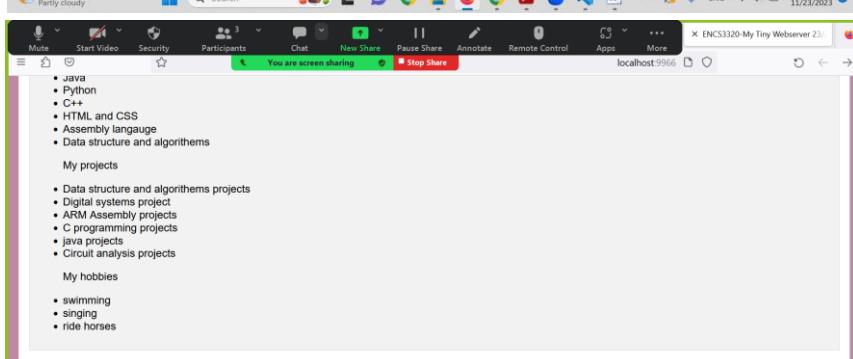
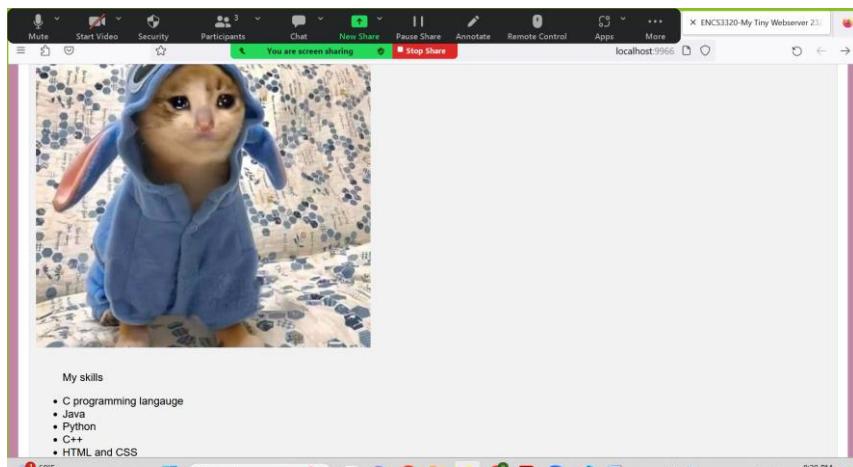
```

Figure 18: code if request file / or /en or /index.html or /main\_en.html

## Screenshot from the same computer:

"/":





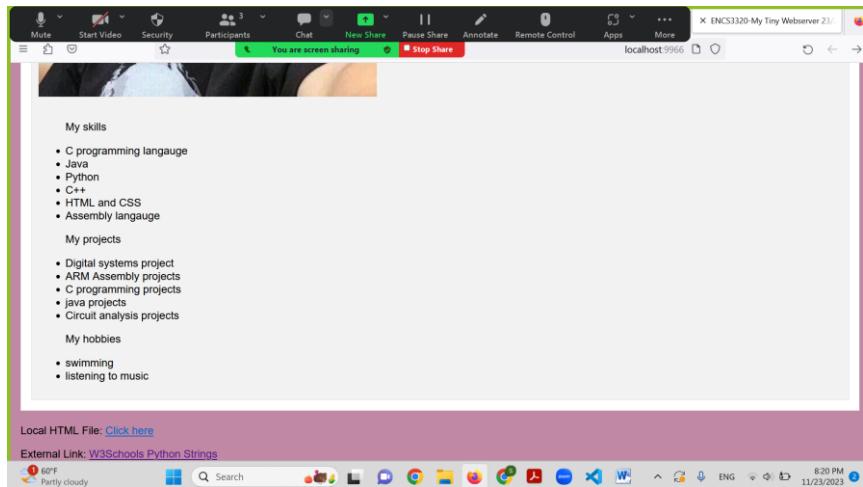
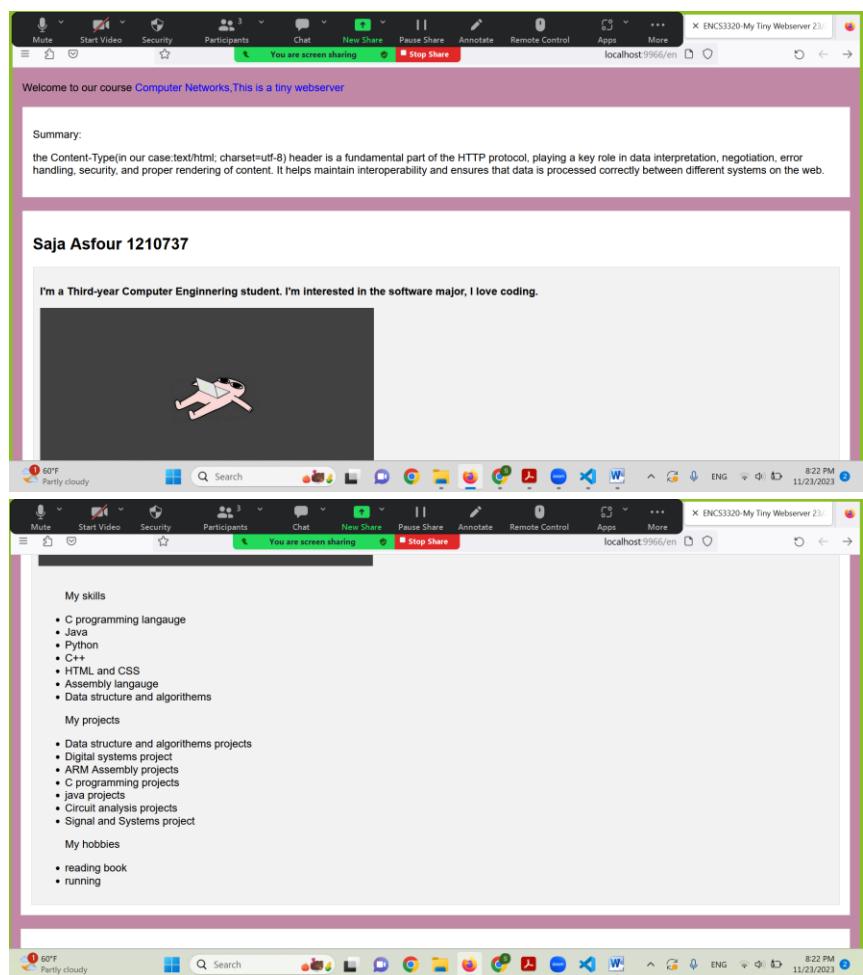


Figure 19: The result of request for /

/en



**Shahd shreth 1210444**

I'm a Third-year Computer Engineering student. I'm interested in the software major, I love coding.



60°F Partly cloudy 8:23 PM 11/23/2023

Mute Start Video Security Participants Chat New Share Pause Share Annotate Remote Control Apps More X ENCS3320-My Tiny Webserver 23... localhost:9966/en

You are screen sharing Stop Share

My skills

- C programming language
- Java
- Python
- C++
- HTML and CSS
- Assembly language
- Data structure and algorithms

My projects

- Data structure and algorithms projects
- Digital systems project
- ARM Assembly projects
- C programming projects
- java projects
- Circuit analysis projects

My hobbies

- swimming
- singing
- ride horses

60°F Partly cloudy 8:23 PM 11/23/2023

Mute Start Video Security Participants Chat New Share Pause Share Annotate Remote Control Apps More X ENCS3320-My Tiny Webserver 23... localhost:9966/en

You are screen sharing Stop Share

**Rouand Bawatneh 1211403**

I'm a Third-year Computer Engineering student. I'm interested in the software major, I love coding.



60°F Partly cloudy 8:23 PM 11/23/2023

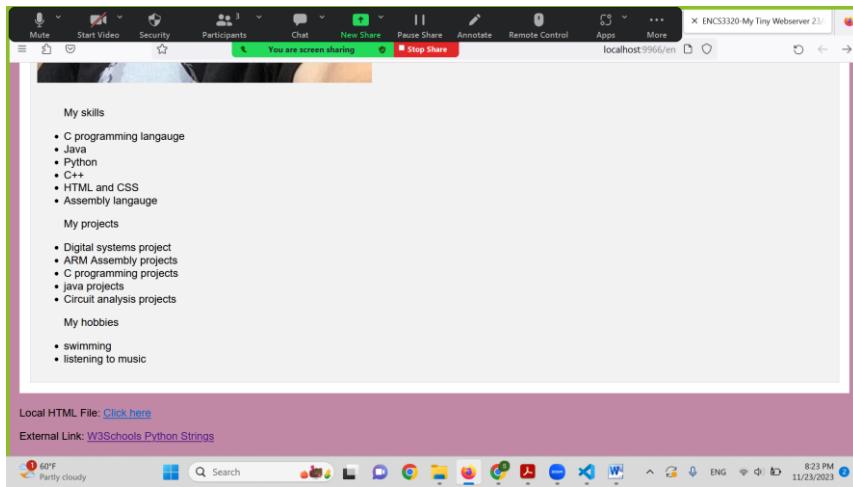
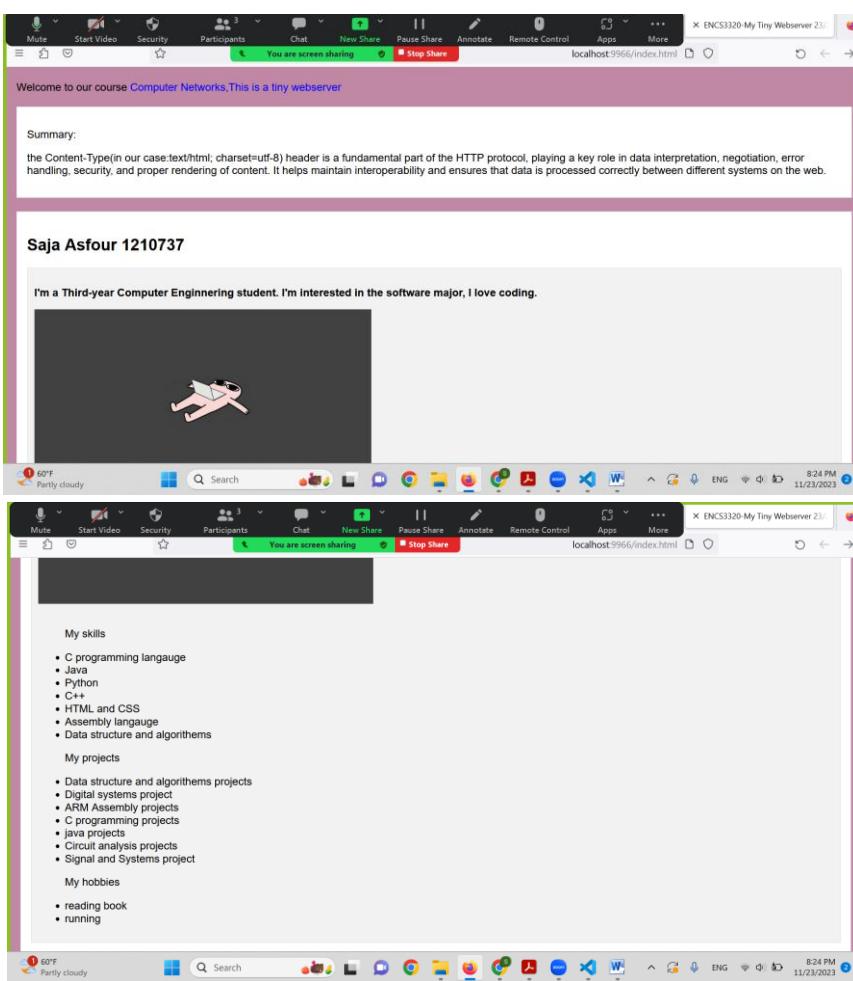


Figure 20: The result of request for /en

/index.html:



**Shahd shreteh 1210444**

I'm a Third-year Computer Engineering student. I'm interested in the software major, I love coding.



My skills

- C programming language
- Java
- Python
- C++
- HTML and CSS
- Assembly language
- Data structure and algorithms

My projects

- Data structure and algorithms projects
- Digital systems project
- ARM Assembly projects
- C programming projects
- Java projects
- Circuit analysis projects

My hobbies

- swimming
- singing
- ride horses

**Rouand Bawatneh 1211403**

I'm a Third-year Computer Engineering student. I'm interested in the software major, I love coding.



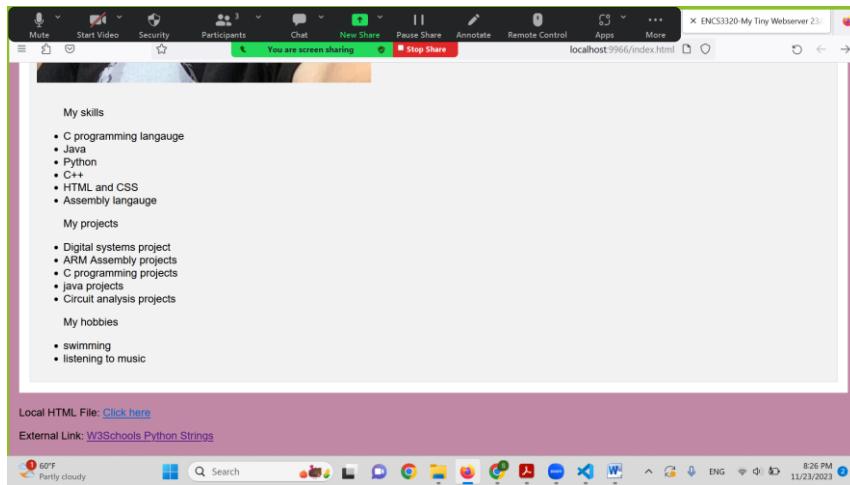
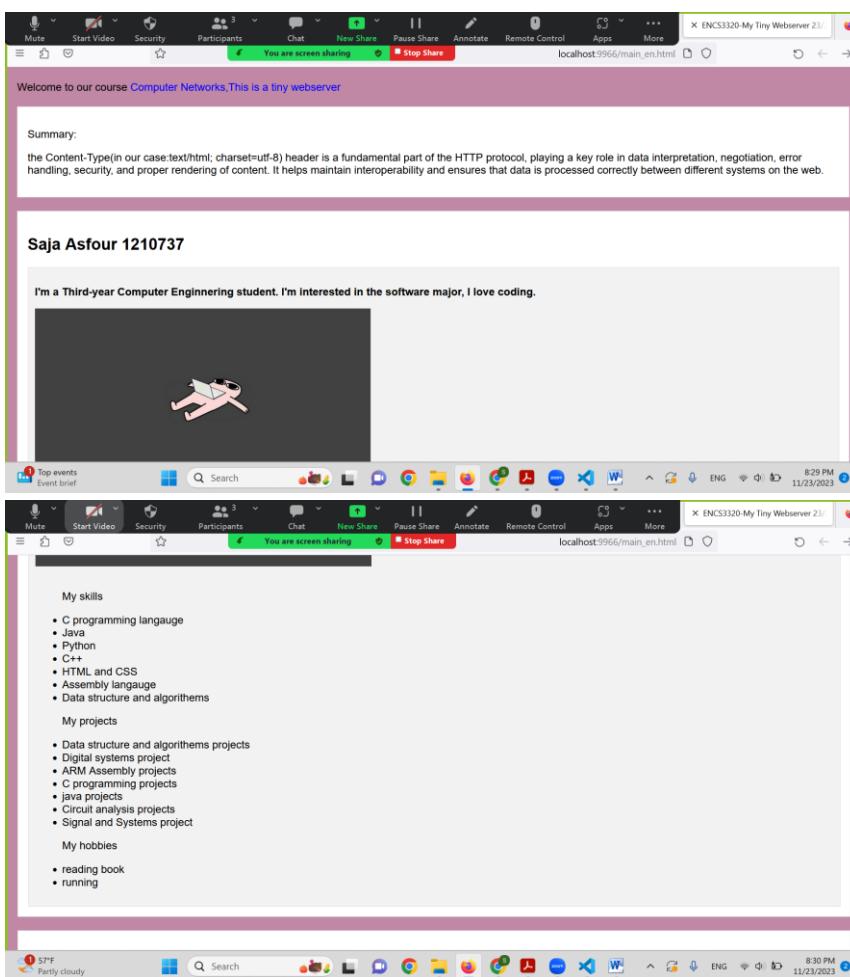


Figure 21: The result of request for /index.html

## 'main\_en.html'



**Shahd shreteh 1210444**

I'm a Third-year Computer Engineering student. I'm interested in the software major, I love coding.



My skills

- C programming language
- Java
- Python
- C++
- HTML and CSS
- Assembly language
- Data structure and algorithms

My projects

- Data structure and algorithms projects
- Digital systems project
- ARM Assembly projects
- C programming projects
- java projects
- Circuit analysis projects

My hobbies

- swimming
- singing
- ride horses

**Rouand Bawatneh 1211403**

I'm a Third-year Computer Engineering student. I'm interested in the software major, I love coding.



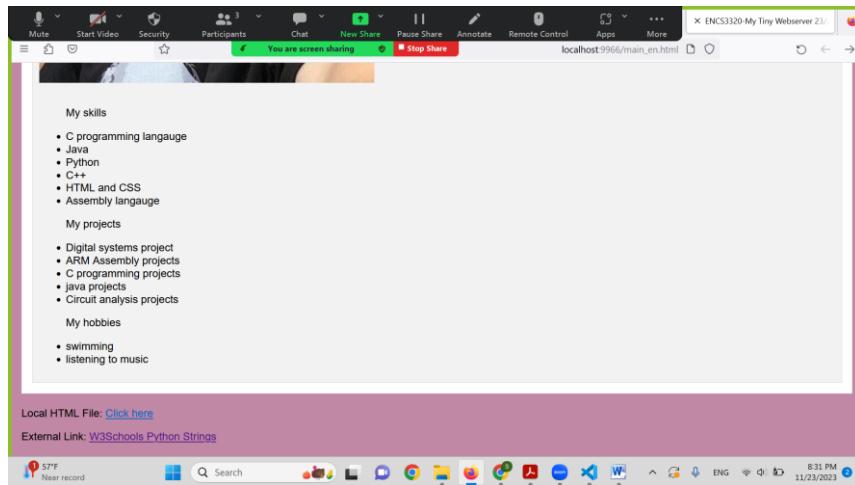


Figure 22: The result of request for /main\_en.html

Each of the pages contains two links.

A link that opens another html page:

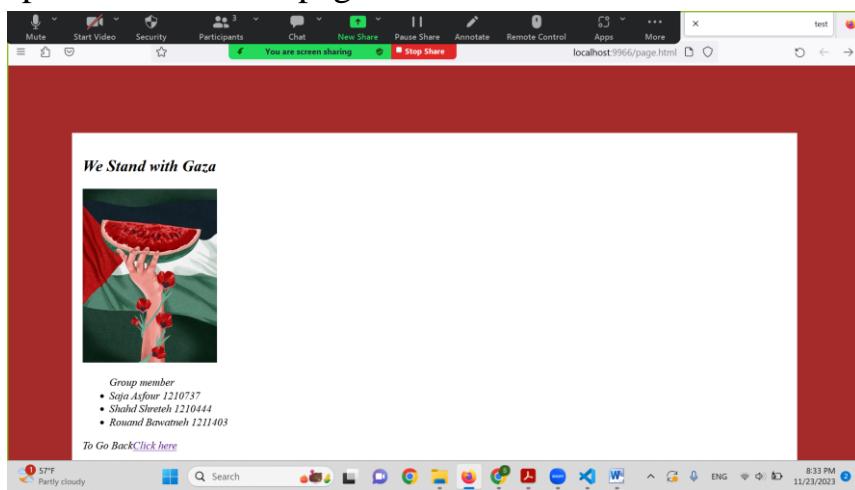


Figure 23: the html page of link

With the above page , it has a link (Go back!) that returns to

[http://localhost:9966/main\\_en.html](http://localhost:9966/main_en.html)

```

IP: 127.0.0.1,Port: 53452
*****
GET /page.html HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/main_en.html
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

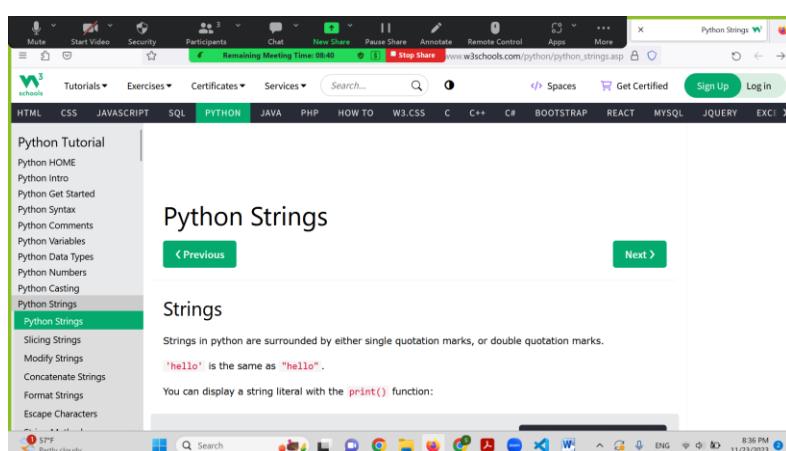
*****
The request File is: page.html
response status: 200 OK

('127.0.0.1', 53453)
IP: 127.0.0.1,Port: 53453
*****
GET /p.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,*/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/page.html
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin

```

Figure 24:the output on the command

And link that opens W3School website:



The output on the terminal from above is(The screen above is when we search main\_en.html ):

```

PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('127.0.0.1', 53368)
IP: 127.0.0.1,Port: 53368
*****
GET /main_en.html HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

*****
The request File is: main_en.html
('127.0.0.1', 53369)
IP: 127.0.0.1,Port: 53369
*****
GET /styles.css HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/css,*/*;q=0.1
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/main_en.html
Sec-Fetch-Dest: style
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin

```

```

*****
The request File is: styles.css
response status: 200 OK

('127.0.0.1', 53370)
IP: 127.0.0.1,Port: 53370
*****
GET /s.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,*/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/main_en.html
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin

*****
The request File is: s.jpg
response status: 200 OK

('127.0.0.1', 53373)
IP: 127.0.0.1,Port: 53373
*****
GET /r.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,*/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3

```

The terminal window displays the following output:

```

IP: 127.0.0.1,Port: 53373
*****
GET /r.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/main_en.html
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin

*****
The request File is: r.jpg
response status: 200 OK

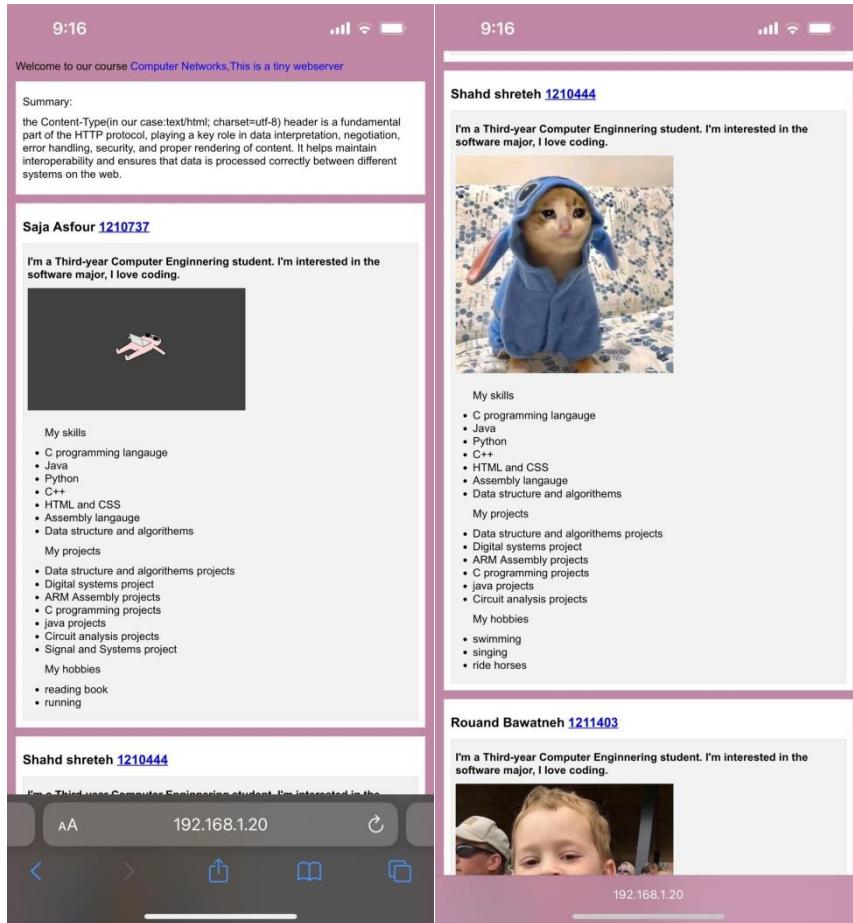
('127.0.0.1', 53374)
IP: 127.0.0.1,Port: 53374
*****
GET /sh.png HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/main_en.html
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin

```

The terminal window also shows the system tray and taskbar at the bottom.

Figure 25: the output when request message is /main\_en.html

## Screenshot from IPHONE



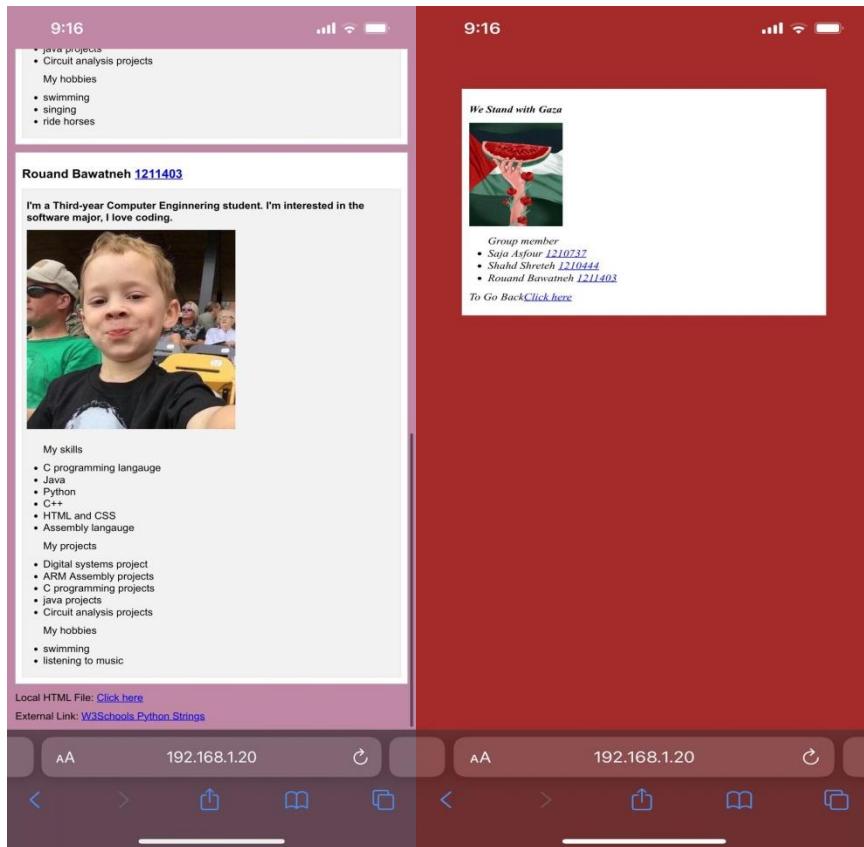


Figure 26:page in IPHONE if request is / or /en or /index.html or /main\_en.html

```

File Edit Selection View Go F PROBLEMS Mute Start Video Security Participants FURIS Chat New Share Pause Share Annotate Remote Control Apps More
EXPLORER OPEN EDITORS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL You are screen sharing Stop Share
IP: 192.168.1.3,Port: 53607 *****
GET / HTTP/1.1
Host: 192.168.1.20:9966
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Accept-Encoding: gzip, deflate
Connection: keep-alive

*****
The request File is:
('192.168.1.3', 53608)
IP: 192.168.1.3,Port: 53608 *****
GET /styles.css HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
Accept: text/css,*/*;q=0.1
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate

*****
The request File is: styles.css
response status: 200 OK
('192.168.1.3', 53609)
IP: 192.168.1.3,Port: 53609

```

```

('192.168.1.3', 53609)
IP: 192.168.1.3,Port: 53609
*****
GET /s.jpg HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
Accept: image/webp,image/avif,image/video/*;q=0.8,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate

*****
The request File is: s.jpg
response status: 200 OK

('192.168.1.3', 53610)
IP: 192.168.1.3,Port: 53610
*****
GET /sh.png HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
Accept: image/webp,image/avif,image/video/*;q=0.8,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate

*****
The request File is: sh.png
response status: 200 OK

```

This screenshot shows a terminal window within a development environment. The terminal output displays log messages from a Python web server running on port 53609. It logs requests for files 's.jpg' and 'sh.png' from an iPhone device at IP address 192.168.1.3. The logs include the host header, connection type, accept headers (mentioning image/webp and image/avif), user agent (Mozilla/5.0 for iPhone), accept-language, referer, and accept-encoding.

```

('192.168.1.3', 53611)
IP: 192.168.1.3,Port: 53611
*****
GET /r.jpg HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
Accept: image/webp,image/avif,image/video/*;q=0.8,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate

*****
The request File is: r.jpg
response status: 200 OK

```

This screenshot shows a terminal window within a development environment. The terminal output displays log messages from a Python web server running on port 53611. It logs a request for file 'r.jpg' from an iPhone device at IP address 192.168.1.3. The logs include the host header, connection type, accept headers (mentioning image/webp and image/avif), user agent (Mozilla/5.0 for iPhone), accept-language, referer, and accept-encoding.

Figure 27:output in terminal when use IPHONE if request is / or /en or /index.html or /main\_en.html

68 9:18

- HTML and CSS
- Assembly language
- Data structure and algorithms

My projects

- Data structure and algorithms projects
- Digital systems project
- ARM Assembly projects
- C programming projects
- Java projects
- Circuit analysis projects
- Signal and Systems project

My hobbies

- reading book
- running

68 9:18

ENCS3320-My Tiny W... 192.168.1.20:9966 X

Welcome to our course Computer Networks, This is a tiny webserver

Summary:

the Content-Type(in our case:text/html; charset=utf-8) header is a fundamental part of the HTTP protocol, playing a key role in data interpretation, negotiation, error handling, security, and proper rendering of content. It helps maintain interoperability and ensures that data is processed correctly between different systems on the web.

### Shahd shreteh 1210444

I'm a Third-year Computer Engineering student. I'm interested in the software major, I love coding.



#### My skills

- C programming language
- Java
- Python

### Saja Asfour 1210737

I'm a Third-year Computer Engineering student. I'm interested in the software major, I love coding.



#### My skills

- C programming language
- Java
- Python
- C++
- HTML and CSS
- Assembly language
- Data structure and algorithms



The image displays two side-by-side screenshots of a mobile application interface, likely from an Android device. Both screenshots show a similar layout for a student profile.

**Screenshot 1 (Left):**

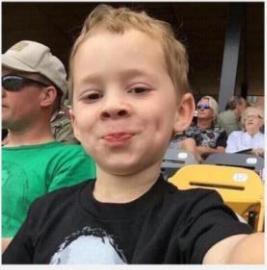
- Header:** Shows signal strength, battery level (68%), and time (9:18).
- Section 1:** **MY SKILLS**
  - C programming langauge
  - Java
  - Python
  - C++
  - HTML and CSS
  - Assembly langauge
  - Data structure and algorithems
- Section 2:** **My projects**
  - Data structure and algorithems projects
  - Digital systems project
  - ARM Assembly projects
  - C programming projects
  - java projects
  - Circuit analysis projects
- Section 3:** **My hobbies**
  - swimming
  - singing
  - ride horses

**Rouand Bawatneh 1211403**

I'm a Third-year Computer Enginnering student. I'm interested in the software major, I love coding.



**Screenshot 2 (Right):**

- Header:** Shows signal strength, battery level (68%), and time (9:18).
- Section 1:** **Rouand Bawatneh 1211403**
- Section 2:** **I'm a Third-year Computer Enginnering student. I'm interested in the software major, I love coding.**
- 
- Section 3:** **My skills**
  - C programming langauge
  - Java
  - Python
  - C++
  - HTML and CSS
  - Assemby langauge
- Section 4:** **My projects**
  - Digital systems project
  - ARM Assembly projects
  - C programming projects
  - java projects
  - Circuit analysis projects
- Section 5:** **My hobbies**
  - swimming
  - listening to music

Local HTML File: [Click here](#)

External Link: [W3Schools Python Strings](#)

Figure 28:page when use android if request is / or /en or /index.html or /main\_en.html

The request File is: r.jpg  
response status: 200 OK

```
(*'192.168.1.10', 51918)
IP: 192.168.1.10,Port: 51918
*****
```

(\*'192.168.1.10', 51920)
IP: 192.168.1.10,Port: 51920
\*\*\*\*\*

(\*'192.168.1.10', 51924)
IP: 192.168.1.10,Port: 51924
\*\*\*\*\*

GET / HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
Accept-Language: ar-EG
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Mobile 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
Referer: android-app://com.google.android.googlequicksearchbox/
Accept-Encoding: gzip, deflate

The request File is:
('192.168.1.10', 51926)
IP: 192.168.1.10,Port: 51926
\*\*\*\*\*

GET /styles.css HTTP/1.1

Ln 87, Col 44 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live

The request File is:
('192.168.1.10', 51926)
IP: 192.168.1.10,Port: 51926
\*\*\*\*\*

GET /styles.css HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Mobile Safari/537.36
Accept: text/css,\*/\*;q=0.1
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate
Accept-Language: ar-EG,ar;q=0.9,en-US;q=0.8,en;q=0.7

The request File is: styles.css
response status: 200 OK

(\*'192.168.1.10', 51928)
IP: 192.168.1.10,Port: 51928
\*\*\*\*\*

GET /s.jpg HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Mobile Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/\*,\*/\*;q=0.8
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate
Accept-Language: ar-EG,ar;q=0.9,en-US;q=0.8,en;q=0.7

The request File is: s.jpg

Ln 87, Col 44 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live

```

*****
The request File is: s.jpg
response status: 200 OK

('192.168.1.10', 51930)
IP: 192.168.1.10,Port: 51930
*****
GET /sh.png HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Mobile Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate
Accept-Language: ar-EG,ar;q=0.9,en-US;q=0.8,en;q=0.7

*****
The request File is: sh.png
response status: 200 OK

('192.168.1.10', 51932)
IP: 192.168.1.10,Port: 51932
*****
GET /r.jpg HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Mobile Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate
Accept-Language: ar-EG,ar;q=0.9,en-US;q=0.8,en;q=0.7

```

```

*****
IP: 192.168.1.10,Port: 51934
*****
GET /r.jpg HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Mobile Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate
Accept-Language: ar-EG,ar;q=0.9,en-US;q=0.8,en;q=0.7

*****
The request File is: r.jpg
response status: 200 OK

('192.168.1.10', 51934)
IP: 192.168.1.10,Port: 51934
*****
GET /favicon.ico HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Mobile Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://192.168.1.20:9966/
Accept-Encoding: gzip, deflate
Accept-Language: ar-EG,ar;q=0.9,en-US;q=0.8,en;q=0.7

```

Figure 29:output in terminal when use android if request is / or /en or /index.html or /main\_en.html

## The Html file:

The image shows a video conferencing interface with two code editors side-by-side. Both editors have the same file open: `main_en.html`. The left editor shows the initial state of the HTML code, while the right editor shows the code after modifications.

**Initial State (Left Editor):**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="styles.css">
<title>ENCS3320-My Tiny Webserver 23/24</title>
</head>
<body>
<section class="course-info">
<p>Welcome to our course <span class="blue-text">Computer Networks, This is a tiny webserver</span></p>
</section>
```

**Completed State (Right Editor):**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="styles.css">
<title>ENCS3320-My Tiny Webserver 23/24</title>
</head>
<body>
<section class="course-info">
<p>Welcome to our course <span class="blue-text">Computer Networks, This is a tiny webserver</span></p>
</section>
<section class="summary-box">
<div class="box">
<p><b>Summary:</b></p>
<p>The Content-Type(in our case:text/html; charset=utf-8) header is a fundamental part of the HTTP protocol</p>
</div>
</section>
<section class="student-info">
<div class="box">
<h2>Saja Asfour 1210737</h2>
<div class="member-box">
<p><b>I'm a Third-year Computer Engineering student. I'm interested in the software major, I love cod</b></p>

<ul>
<li>My skills</li>
<li>C programming language</li>
<li>Java</li>

```

... (lines 28-37)

```

<li>C programming language</li>
<li>Java</li>
<li>Python</li>
<li>C++</li>
<li>HTML and CSS</li>
<li>Assembly language</li>
<li>Data structure and algorithms</li>
</ul>
<ul>
<p>My projects</p>
<li>Data structure and algorithms projects</li>
<li>Digital systems project</li>
<li>ARM Assembly projects</li>
<li>C programming projects</li>
<li>Java projects</li>
<li>Circuit analysis projects</li>
<li>Signal and Systems project</li>
</ul>
<ul>
<p>My hobbies</p>
<li>reading book</li>
<li>running</li>
</ul>
</div>
</div>
</section>
<section class="student-info">
<div class="box">
<h2>Shahd shreneh 1210444</h2>

```

... (line 56)

A screenshot of a Microsoft Teams meeting interface. The top bar shows 'Remaining Meeting Time: 05:54' and various meeting controls like Mute, Start Video, Security, Participants (3), Chat, New Share, Pause Share, Annotate, Remote Control, Apps, and More. The main area displays a code editor for 'main\_en.html'. The code includes sections for 'student-info' and 'member-box', containing details about a student named Shabd shretheh (1210444) and their interests in programming languages like C, Java, Python, C++, HTML, CSS, Assembly, Data structures, and algorithms. It also lists projects such as digital systems, ARM Assembly, C programming, Java, and circuit analysis. The code editor shows lines 54 to 82.

```
<section class="student-info">
  <div class="box">
    <h2>Shabd shretheh 1210444</h2>
    <div class="member-box">
      <p><b>I'm a Third-year Computer Engineering student. I'm interested in the software major, I love cod</b></p>
      
      <ul>
        <li>My skills</li>
        <li>C programming langauge</li>
        <li>Java</li>
        <li>Python</li>
        <li>C++</li>
        <li>HTML and CSS</li>
        <li>Assembly langauge</li>
        <li>Data structure and algorithems</li>
      </ul>
      <ul>
        <li>My projects</li>
        <li>Data structure and algorithems projects</li>
        <li>Digital systems project</li>
        <li>ARM Assembly projects</li>
        <li>C programming projects</li>
        <li>Java projects</li>
        <li>Circuit analysis projects</li>
      </ul>
      <ul>
        <li>My hobbies</li>
        <li>swimming</li>
        <li>singing</li>
        <li>ride horses</li>
      </ul>
    </div>
  </div>
</section>
```

A screenshot of a Microsoft Teams meeting interface, identical to the one above but with a different code editor view. The code editor now shows lines 80 to 108, indicating a scroll down in the code. The content remains the same, describing the student's interests and projects.

```
<ul>
  <li>My hobbies</li>
  <li>swimming</li>
  <li>singing</li>
  <li>ride horses</li>
</ul>
</div>
</div>
</section>
<section class="student-info">
  <div class="box">
    <h2>Rouand Bawatneh 1211403</h2>
    <div class="member-box">
      <p><b>I'm a Third-year Computer Engineering student. I'm interested in the software major, I love cod</b></p>
      
      <ul>
        <li>My skills</li>
        <li>C programming langauge</li>
        <li>Java</li>
        <li>Python</li>
        <li>C++</li>
        <li>HTML and CSS</li>
        <li>Assembly langauge</li>
      </ul>
      <ul>
        <li>My projects</li>
        <li>Digital systems project</li>
        <li>ARM Assembly projects</li>
        <li>C programming projects</li>
        <li>java projects</li>
      </ul>
    </div>
  </div>
</section>
```

```
<ul>
    <li>Python</li>
    <li>C++</li>
    <li>HTML and CSS</li>
    <li>Assembly language</li>
</ul>
<ul>
    <p>My projects</p>
    <li>Digital systems project</li>
    <li>ARM Assembly projects</li>
    <li>C programming projects</li>
    <li>Java projects</li>
    <li>Circuit analysis projects</li>
</ul>
<ul>
    <p>My hobbies</p>
    <li>swimming</li>
    <li>listening to music</li>
</ul>
</div>
</div>
</section>
<section class="links">
    <p>Local HTML File: <a href="page.html">Click here</a></p>
    <p>External Link: <a href="https://www.w3schools.com/python/python_strings.asp">W3Schools Python Strings</a></p>
</section>
</body>
</html>
```

Figure 30: main\_en.html code

The html file is connected with a css file:

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #c088a4;
}

header {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 1em 0;
}

.blue-text {
    color: blue;
}

.course-info, .summary-box, .student-info, .images, .links {
    margin: 20px;
}

.box {
    background-color: #fff;
    padding: 15px;
    border: 1px solid #ddd;
    margin-bottom: 15px;
}
```

```
# styles.CSS > ...
15 .blue-text {
16 |   color: blue;
17 }
18 .course-info, .summary-box, .student-info, .images, .links {
19 |   margin: 20px;
20 }
21 .box {
22 |   background-color: #ffff;
23 |   padding: 15px;
24 |   border: 1px solid #ddd;
25 |   margin-bottom: 15px;
26 }
27 .member-box {
28 |   background-color: #f2f2f2;
29 |   padding: 10px;
30 |   border: 1px solid #ddd;
31 |   margin-top: 10px;
32 }
33 .img {
34 |   max-width: 100%;
35 |   height: auto;
36 |   margin-bottom: 15px;
37 }
38
39
40
41
42
```

Figure 31: styles.css code

The html code(for local html ):

```
<!DOCTYPE html>
<html> <head>
<meta charset="utf-8">
<title>test</title>
</head> <style>
.Group{
background-color: brown;
margin: 10px;
font-style: italic;
}
.box {
background-color: #ffff;
padding: 15px;
border: 1px solid #ddd;
margin-bottom: 15px;
}
</style>
<body class="Group">
<section class="box">
<h1>We Stand with Gaza</h1>

<ul>
<li>Group member</li>
<li>Saja Asfour 1210737</li>
<li>Shad Shreneh 1218444</li>
<li>Rouand Bawatneh 1211403</li>
</ul>
</section>
</body>
</html>
```

The screenshot shows a Microsoft Teams meeting interface with the code editor open. The file path is 'page.html'. The code displays an HTML page with a CSS style block. The style block defines a 'Group' class with a brown background color and italic font style, and a 'box' class with a white background color, 15px padding, 1px solid #ddd border, and 15px margin-bottom. The main content includes a section titled 'Stand with Gaza' with an image and a list of group members: Saja Asfour, Shabd Shretab, and Rouand Bawatneh. A link to 'main\_en.html' is provided for returning to the English version.

```

</head>
<style>
    .Group{
        background-color: #brown;
        margin: 10px;
        font-style: italic;
    }
    .box {
        background-color: #fff;
        padding: 15px;
        border: 1px solid #ddd;
        margin-bottom: 15px;
    }
</style>
<body class="Group">
    <section class="box">
        <h1>Stand with Gaza</h1>
        
        <ul>
            <li>Group member</li>
            <li>Saja Asfour 1210737</li>
            <li>Shabd Shretab 1210444</li>
            <li>Rouand Bawatneh 1211403</li>
        </ul>
        <p><a href="main_en.html">Click here</a></p>
    </section>
</body>

```

Figure 32: page.html code

2-If the request is /ar then the server response with **main\_ar.html** which is an Arabic version of main\_en.html

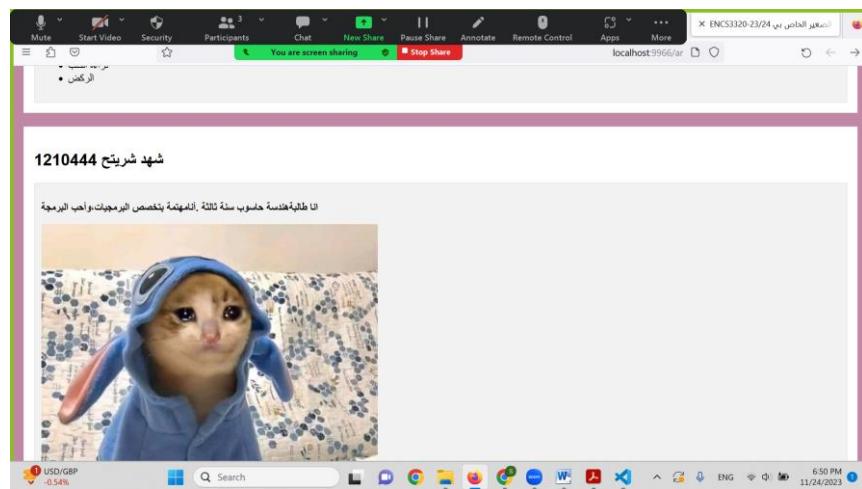
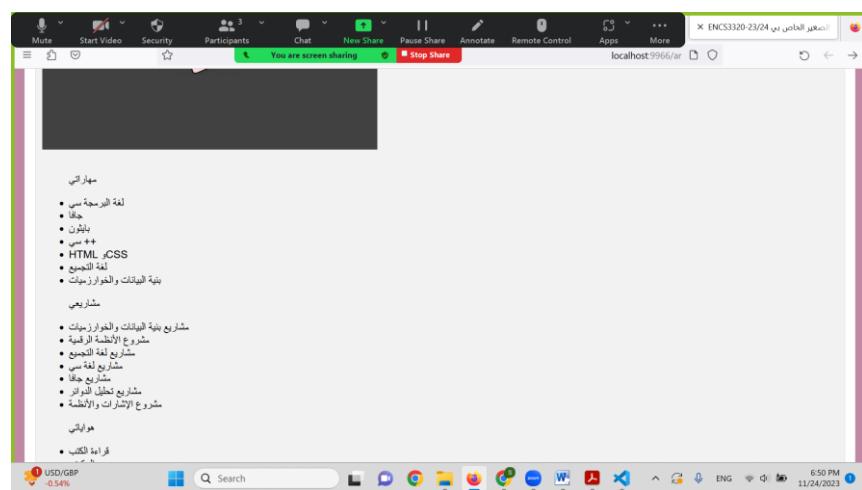
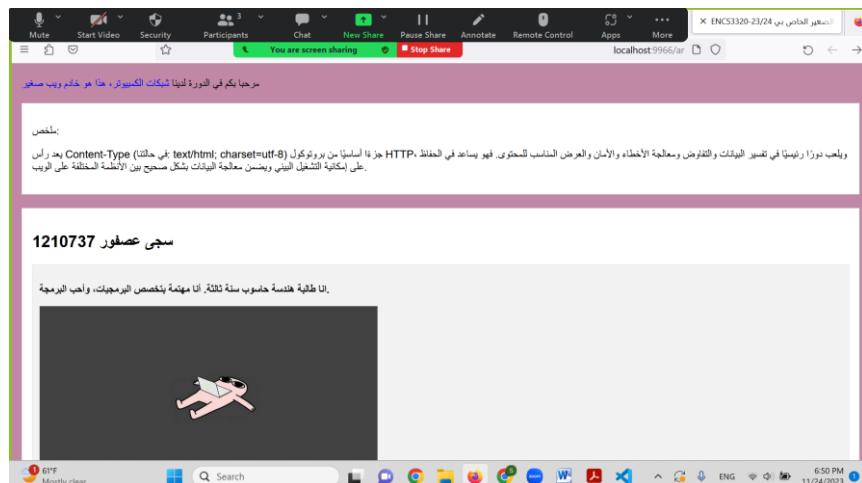
```

44 |     mhtml.close()
45 |     #If the request is /ar then the server response with main_ar.html which is an Arabic version of main_en.html
46 |     elif request_file=='ar':
47 |         connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
48 |         connectionSocket.send(f"Content-Type: text/html \r\n".encode())
49 |         connectionSocket.send(f"\r\n".encode())
50 |         mhtml=open('main_ar.html' , 'rb')
51 |         connectionSocket.send(mhtml.read())
52 |         mhtml.close() (variable) request File: str

```

Figure 33: code when request is /ar

## /ar: test from the same computer



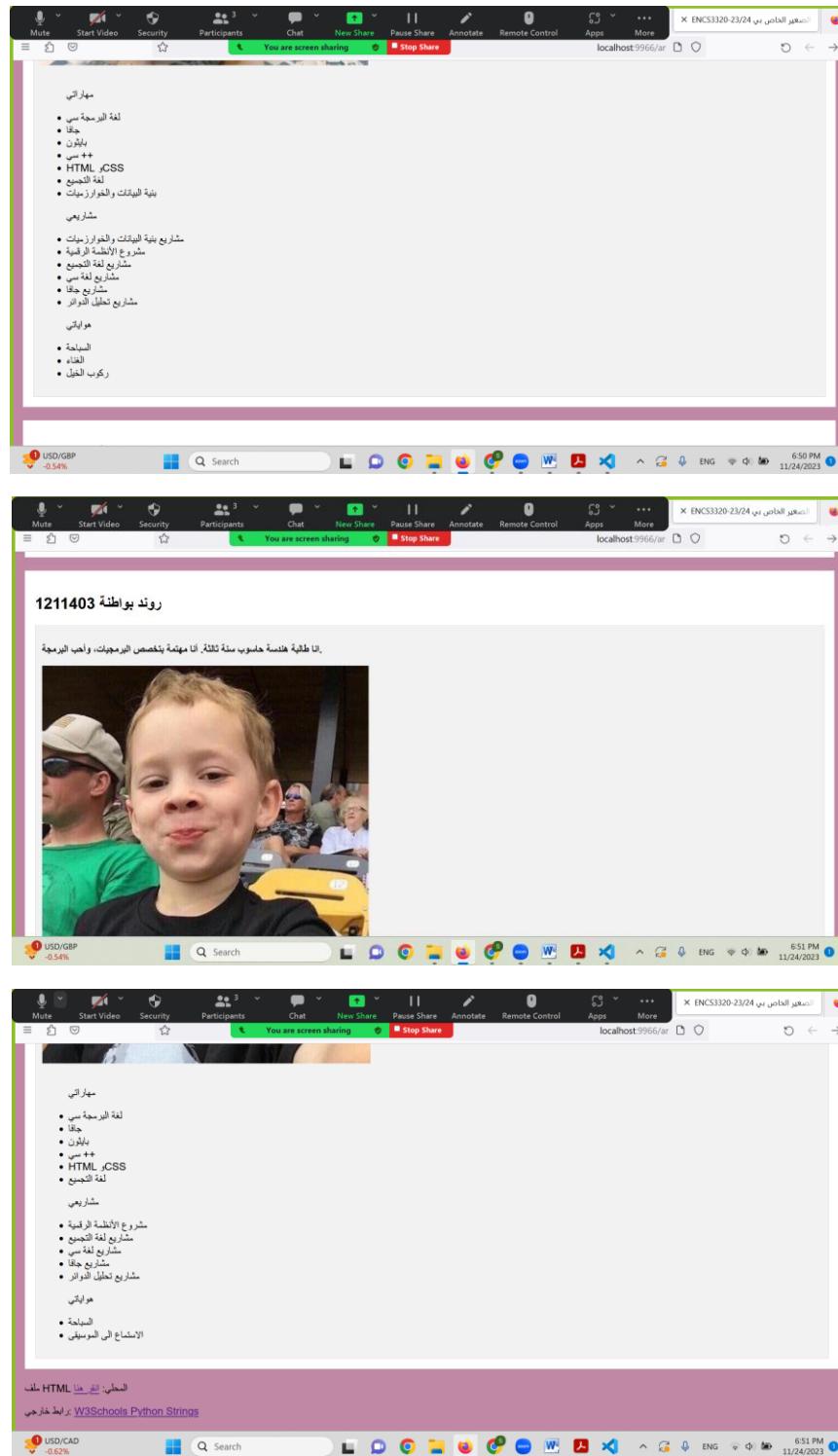


Figure 34:the result of the request /ar from the same computer

The screenshot shows a software interface with a terminal window and a file browser. The terminal window displays the output of a Python script named part3.py. The script is designed to handle requests for files like index.html or main.html. It prints the IP address (127.0.0.1) and port (50555). It also handles requests for styles.css and page.html, printing their respective IP addresses (127.0.0.1) and ports (50565 and 50573). The terminal also shows a GET request from a Firefox browser for index.html, with headers indicating the user agent (Mozilla/5.0), accept (text/html, application/xhtml+xml, application/xml;q=0.9, image/webp, \*/\*;q=0.8), accept language (ar, en-US;q=0.7, en;q=0.3), accept encoding (gzip, deflate, br), connection (keep-alive), upgrade-insecure-requests (1), sec-fetch-dest (document), sec-fetch-mode (navigate), sec-fetch-site (none), and sec-fetch-user (?1).

```
PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:/Users/DELL/Desktop/network/project1/codes/part3.py"
The server is ready to receive
('127.0.0.1', 50555)
IP: 127.0.0.1,Port: 50555
*****
('127.0.0.1', 50565)
IP: 127.0.0.1,Port: 50565
*****
('127.0.0.1', 50573)
IP: 127.0.0.1,Port: 50573
*****
GET /an HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

This screenshot is nearly identical to the one above, showing the same software interface with a terminal window and a file browser. The terminal window displays the output of the Python script part3.py. The script handles requests for index.html, styles.css, and page.html, printing their respective IP addresses (127.0.0.1) and ports (50574, 50575, and 50575). The terminal also shows a GET request from a Firefox browser for styles.css, with headers indicating the user agent (Mozilla/5.0), accept (text/css,\*/\*;q=0.1), accept language (ar,en-US;q=0.7,en;q=0.3), accept encoding (gzip, deflate, br), connection (keep-alive), referer (http://localhost:9966/ar), sec-fetch-dest (style), sec-fetch-mode (no-cors), and sec-fetch-site (same-origin). The response status is 200 OK.

```
*****  
The request File is: ar  
('127.0.0.1', 50574)  
IP: 127.0.0.1,Port: 50574  
*****  
GET /styles.css HTTP/1.1  
Host: localhost:9966  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0  
Accept: text/css,*/*;q=0.1  
Accept-Language: ar,en-US;q=0.7,en;q=0.3  
Accept-Encoding: gzip, deflate, br  
Connection: keep-alive  
Referer: http://localhost:9966/ar  
Sec-Fetch-Dest: style  
Sec-Fetch-Mode: no-cors  
Sec-Fetch-Site: same-origin  
*****  
The request File is: styles.css  
response status: 200 OK  
('127.0.0.1', 50575)  
IP: 127.0.0.1,Port: 50575  
*****  
GET /s.jpg HTTP/1.1
```

A screenshot of a video conferencing application interface. At the top, there are buttons for Mute, Start Video, Security, Participants (3), Chat, New Share, Pause Share, Annotate, Remote Control, Apps, and More. Below the toolbar, a code editor shows part3.py with the following code:

```
'127.0.0.1', 50575)
IP: 127.0.0.1,Port: 50575
*****
GET /s.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/ar
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin

*****
The request File is: s.jpg
response status: 200 OK

('127.0.0.1', 50576)
IP: 127.0.0.1,Port: 50576
*****
GET /sh.png HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/ar
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

The terminal window below shows the same log entries. The bottom status bar indicates Ln 35, Col 17, Spaces: 4, UTF-8, CRLF, Python 3.11.5 64-bit, Go Live, and the date/time 11/24/2023 6:53 PM.

A second screenshot of the same video conferencing application interface. The code editor and terminal window content are identical to the first screenshot, showing the same log entries for file requests and responses. The bottom status bar indicates Ln 35, Col 17, Spaces: 4, UTF-8, CRLF, Python 3.11.5 64-bit, Go Live, and the date/time 11/24/2023 6:53 PM.

```

('127.0.0.1', 50578)
IP: 127.0.0.1, Port: 50578
*****
GET /r.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,*/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/ar
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin

*****
The request File is: r.jpg
response status: 200 OK

('127.0.0.1', 50577)
IP: 127.0.0.1, Port: 50577
*****

```

Figure 35:the output result of the request /ar from the same computer

And if we click هنا انقر

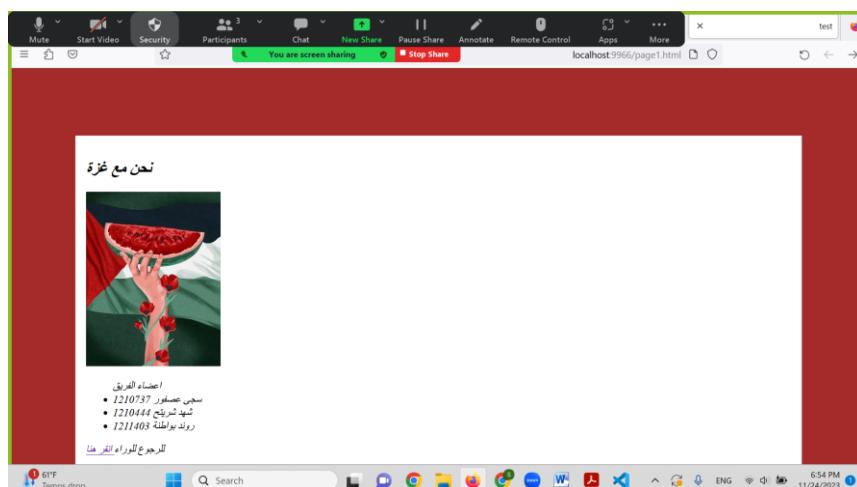


Figure 36:page when click the local html link

And if we click w3schools website:

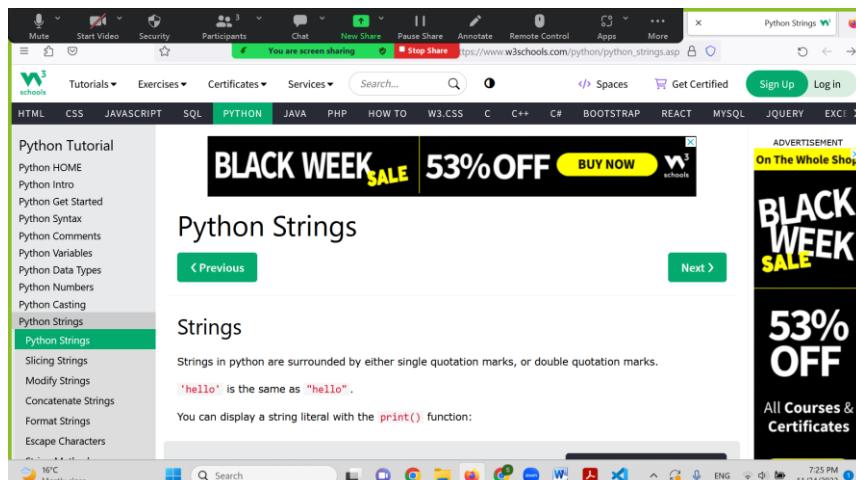


Figure 37:page when click the external link

```

Remaining Meeting Time: 06:14
part3.py ...
34 try:
35     #if the request is index.html/main.html or any file is found in the project file
36     #if the file is not found then exception is raised
37     #if the requested file is main.html or index.html
38     if request_file == '' or request_file=='main_en.html' or request_file== 'index.html' or request_file == 'en':
39         connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
*****
('127.0.0.1', 50594)
IP: 127.0.0.1, Port: 50594
*****
GET /page1.html HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.3
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/ar
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
*****
The request File is: page1.html
response status: 200 OK

Ln 35, Col 17  Spaces: 4  UTF-8  CRLF  Python  3.11.5 64-bit  Go Live  7:25 PM  11/24/2023
16°C  Mostly clear

main_en.html # style ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
-----[REDACTED]-----
Ln 35, Col 17  Spaces: 4  UTF-8  CRLF  Python  3.11.5 64-bit  Go Live  7:14 PM  11/24/2023
16°C  Mostly clear
part3.py ...
38     if request_file == '' or request_file=='main_en.html' or request_file== 'index.html' or request_file == 'en':
39         connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
*****
The request File is: page1.html
response status: 200 OK

('127.0.0.1', 50595)
IP: 127.0.0.1, Port: 50595
*****
GET /p.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,*/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/page1.html
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
*****
Ln 35, Col 17  Spaces: 4  UTF-8  CRLF  Python  3.11.5 64-bit  Go Live  7:14 PM  11/24/2023
16°C  Mostly clear
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
-----[REDACTED]-----
Ln 35, Col 17  Spaces: 4  UTF-8  CRLF  Python  3.11.5 64-bit  Go Live  7:14 PM  11/24/2023
16°C  Mostly clear

```

```

Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/page1.html
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin

*****
The request File is: p.jpg
response status: 200 OK

```

The screenshot shows a terminal window displaying the output of a file request. The terminal shows the command being run, the response headers (Accept-Encoding, Connection, Referer, Sec-Fetch-Dest, Sec-Fetch-Mode, Sec-Fetch-Site), and the file path (p.jpg) and response status (200 OK). The terminal interface includes a sidebar with project files like p.jpg, page.html, page1.html, part2Client.py, part2Server.py, part3.py, r.jpg, s.html, s.jpg, sh.png, and styles.CSS, along with OUTLINE and TIMELINE tabs. The bottom of the screen shows a Windows taskbar with various application icons and system status.

Figure 38:output when click the link

the screenshot from Iphone:

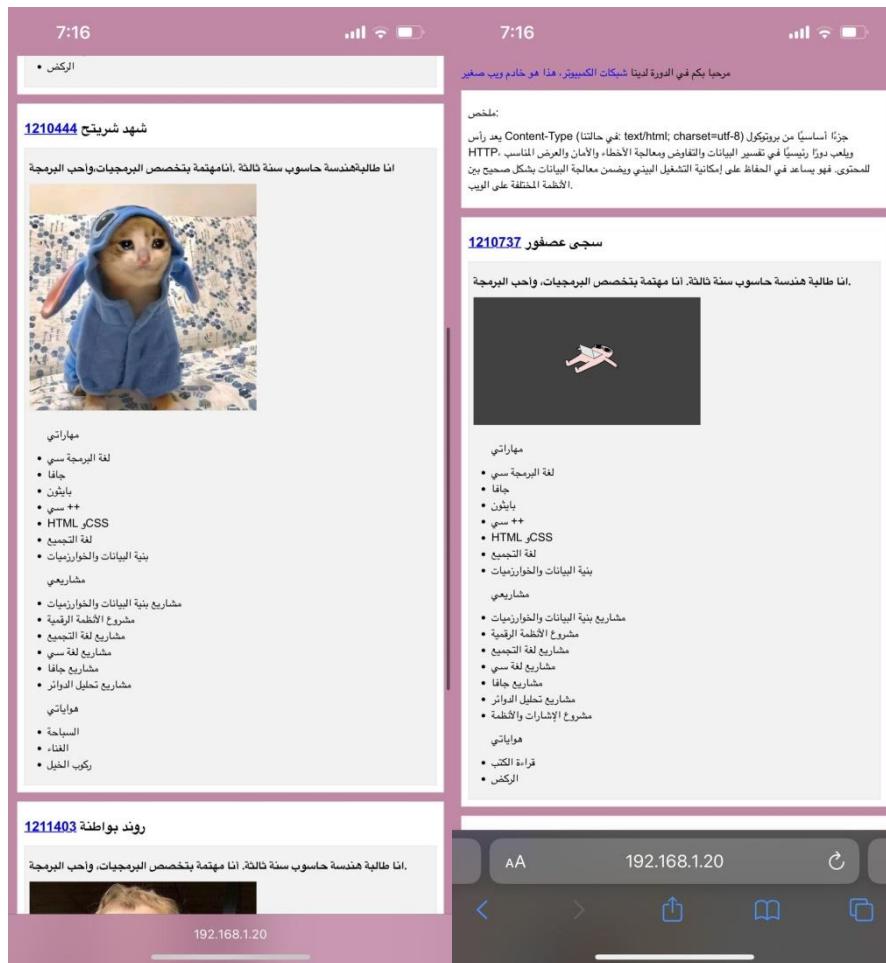




Figure 39:the page when request /ar from IPHONE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('192.168.1.3', 55182)
IP: 192.168.1.3,Port: 55182
*****
GET /ar HTTP/1.1
Host: 192.168.1.20:9966
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Accept-Encoding: gzip, deflate
Connection: keep-alive

*****
The request File is: ar
('192.168.1.3', 55183)
IP: 192.168.1.3,Port: 55183
*****
GET /styles.css HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
GET /styles.css HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
Accept: text/css,*/*;q=0.1
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.20:9966/ar
Accept-Encoding: gzip, deflate

*****
The request File is: styles.css
response status: 200 OK

('192.168.1.3', 55184)
IP: 192.168.1.3,Port: 55184
*****
GET /s.jpg HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
Accept: image/webp,image/avif,image/*;q=0.8,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.20:9966/ar
Accept-Encoding: gzip, deflate

Ln 35, Col 17 Spaces: 4 UTF-8 CRLF { Python 3.11.5 64-bit ⚡ Go Live 🔍 7:22 PM 1 11/24/2023

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Accept: image/webp,image/avif,image/*;q=0.8,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.20:9966/ar
Accept-Encoding: gzip, deflate

*****
The request File is: s.jpg
response status: 200 OK

('192.168.1.3', 55185)
IP: 192.168.1.3,Port: 55185
*****
GET /sh.png HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
Accept: image/webp,image/avif,image/*;q=0.8,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.20:9966/ar

Ln 35, Col 17 Spaces: 4 UTF-8 CRLF { Python 3.11.5 64-bit ⚡ Go Live 🔍 7:22 PM 1 11/24/2023

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Referer: http://192.168.1.20:9966/ar
Accept-Encoding: gzip, deflate

*****
The request File is: sh.png
response status: 200 OK

('192.168.1.3', 55186)
IP: 192.168.1.3,Port: 55186
*****
GET /r.jpg HTTP/1.1
Host: 192.168.1.20:9966
Connection: keep-alive
Accept: image/webp,image/avif,image/*;q=0.8,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 16_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.6 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.20:9966/ar
Accept-Encoding: gzip, deflate

Ln 35, Col 17 Spaces: 4 UTF-8 CRLF { Python 3.11.5 64-bit ⚡ Go Live 🔍 7:23 PM 1 11/24/2023

```

Figure 40:the output when request /ar from IPHONE

And if we click هنا انقر



Figure 41:the page when click local html link from IPHONE

Figure 42:the output when click local html link from IPHONE

And if we click w3schools website:

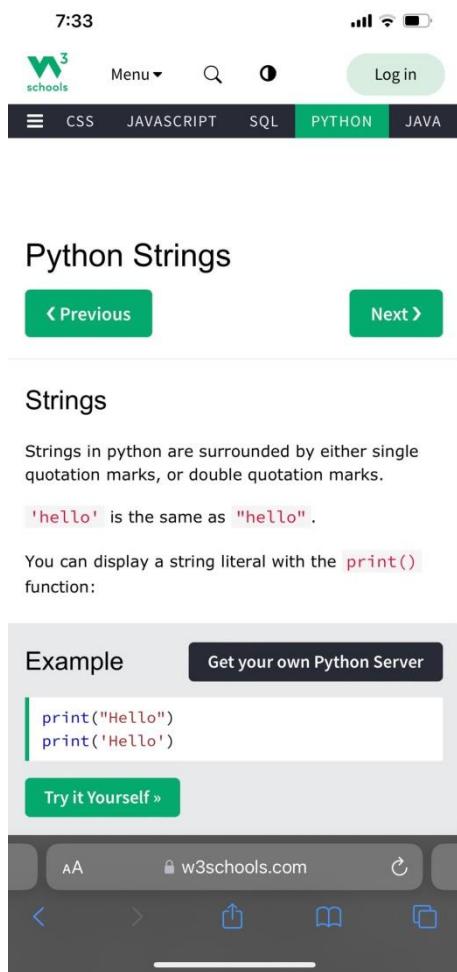
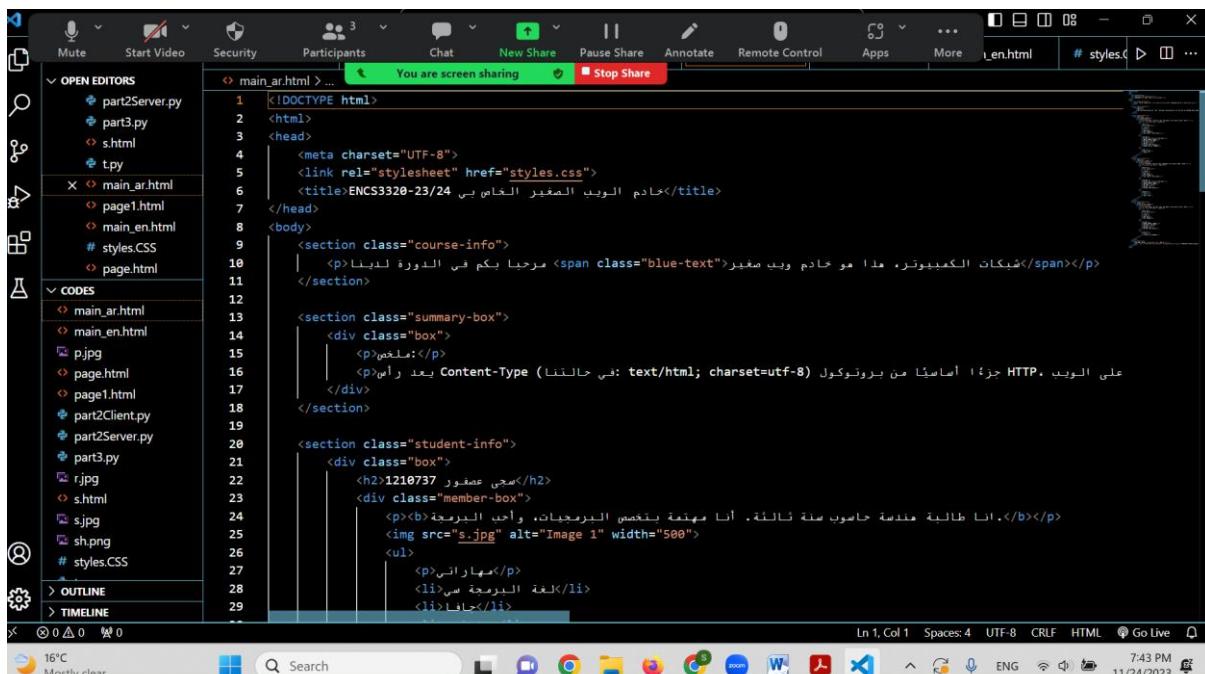


Figure 43:the page when click external link from IPHONE

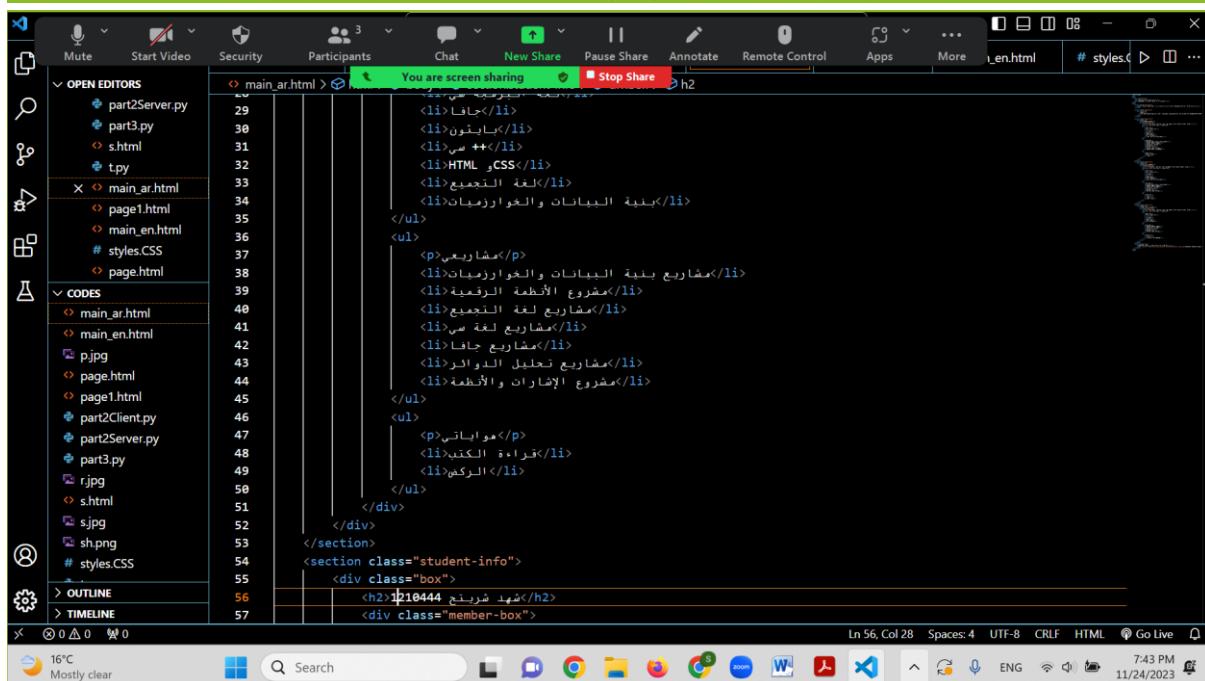
The HTML file is :



```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="styles.css">
<title>ENCS3320-23/24 حادم المعلم المعلم الخاص بـ</title>
</head>
<body>
<section class="course-info">
<p>مرحبا بك في الدورة لبيانات الكمبيوتر، هذا هو حادم ويب مفتوحة</p>
</section>
<section class="summary-box">
<div class="box">
<p>عنوان:<br/>1210737<br/>مقدور سجى</p>
<p>رقم:<br/>1210737<br/>مقدور سجى</p>

<ul>
<li>جامعة عجمان</li>
<li>لغة البرمجة سس</li>
<li>أداة طالبة متعددة حاسون منه ثالثة، أنا مهتمة بتحصين البرمجيات، وأحب البرمجة</li>

```



```

<ul>
<li>مشاريع بنية البيانات والخوارزميات</li>
<li>مشروع الأنظمة الرقمية</li>
<li>مشاريع لغة التعبير</li>
<li>مشاريع سس</li>
<li>مشروع جافا</li>
<li>مشروع تحليل الدوالز</li>
<li>مشروع الإشارات والأنظمة</li>

```

```
You are screen sharing Stop Share div.member-box > ul
57 <div class="member-box">
58   <p><b>انا طالبة هندسة حاسوب سنة ثالثة . أنا مهتمة بخصوص البرمجيات، وأحب البرمجة</b></p>
59   
61     <li>مهاراتي</li>
62     <li>لغة البرمجة سி</li>
63     <li>جافا</li>
64     <li>بايثون</li>
65     <li>سي ++</li>
66     <li>HTML و CSS</li>
67     <li>لغة التجميع</li>
68     <li>بنية البيانات والخوارزميات</li>
69   </ul>
70   <ul>
71     <li>مشاريع</li>
72     <li>مشاريع بنية البيانات والخوارزميات</li>
73     <li>مشروع الأنظمة الرقمية</li>
74     <li>مشاريع لغة التجميع</li>
75     <li>مشاريع لغة سி</li>
76     <li>مشاريع جافا</li>
77     <li>مشاريع تحليل الدوائر</li>
78   </ul>
79   <ul>
80     <li>مو اباتي</li>
81     <li>السياحة</li>
82     <li>الفن</li>
83     <li>ركوب الخيل</li>
84   </ul>
85 </div>
```

Ln 84, Col 22 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live

```
Search 7:44 PM 11/24/2023
You are screen sharing Stop Share div.member-box > ul > p
85   </div> Participants can now see your screen ×
86 </div>
87 </section>
88 <section class="student-info">
89   <div class="box">
90     <h2>1211403</h2> روند بواطنة
91   <div class="member-box">
92     <p><b>انا طالبة هندسة حاسوب سنة ثالثة . أنا مهتمة بخصوص البرمجيات، وأحب البرمجة</b></p>
93     
95       <li>مهاراتي</li>
96       <li>لغة البرمجة سيء</li>
97       <li>جافا</li>
98       <li>بايثون</li>
99       <li>سي ++</li>
100      <li>HTML و CSS</li>
101      <li>لغة التجميع</li>
102    </ul>
103    <ul>
104      <li>مشاريع</li>
105      <li>مشروع الأنظمة الرقمية</li>
106      <li>مشاريع لغة التجميع</li>
107      <li>مشاريع لغة سيء</li>
108      <li>مشاريع جافا</li>
109      <li>مشاريع تحليل الدوائر</li>
110    </ul>
111    <ul>
112      <li>مو اباتي</li>
113      <li>السياحة</li>
```

Ln 112, Col 28 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live

```

98     <li>بيانو</li>
99     <li>سي ++</li>
100    <li>HTML و CSS</li>
101    <li>لغة التجميع</li>
102    </ul>
103    <ul>
104        <p>مشاريع</p>
105        <li>مشروع الأنظمة الرقمية</li>
106        <li>مشاريع لغة التجميع</li>
107        <li>مشاريع لغة سي</li>
108        <li>مشاريع جافا</li>
109        <li>مشاريع تحليل الدولار</li>
110    </ul>
111    <ul>
112        <p>موابات</p>
113        <li>السلاسل</li>
114        <li>الاستماع إلى الموسيقى</li>
115    </ul>
116    </div>
117 </div>
118 </section>
119 <section class="links">
120     <p>المحلية : اتفق هنا <a href="page1.html"></a></p>
121     <p>الدولية : رابط خارجي <a href="https://www.w3schools.com/python/python_strings.asp">W3Schools Python Strings</a></p>
122 </section>
123 </body>
124 </html>
125

```

Ln 125, Col 1 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live

Search

8:01 PM 11/24/2023

Figure 44:main\_ar.html code

And the html code for the link:

```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="utf-8">
5         <title>test</title>
6     </head>
7     <style>
8         .Group{
9             background-color: #brown;
10            margin: 10px;
11            font-style: italic;
12        }
13        .box {
14            background-color: #fff;
15            padding: 15px;
16            border: 1px solid #ddd;
17            margin-bottom: 15px;
18        }
19    </style>
20    <body class="Group">
21        <section class="box">
22            <h1>دفن مع غرفة</h1>
23            
24            <ul>
25                <li>أعمال الفريدي</li>
26                <li>1210737 سجين عصبي</li>
27                <li>1210444 شرير</li>
28                <li>1211403 روبي بواطنة</li>
29            </ul>
30            <p>انقر هنا <a href="main_ar.html"></a></p>
31        </section>
32    </body>
33 </html>

```

Ln 30, Col 46 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live

16°C Mostly clear

8:02 PM 11/24/2023

Search

8:02 PM 11/24/2023

Figure 45:page1.html code

**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.**

```
52 |     mhtml.close()
53 |     #if the request is an .html file then the server should send the requested html file with Content-Type: text/
54 |     elif '.html' in request_File:
55 |         connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
56 |         connectionSocket.send(f"Content-Type: text/html \r\n".encode())
57 |         connectionSocket.send(f"\r\n".encode())
58 |         print(' response status: 200 OK\n\n')
59 |         f= open(str(request_File), 'rb')
60 |         connectionSocket.send(f.read())
61 |         f.close()
```

Figure 46:the code when request is an .html file

If we search for any .html like page.html :

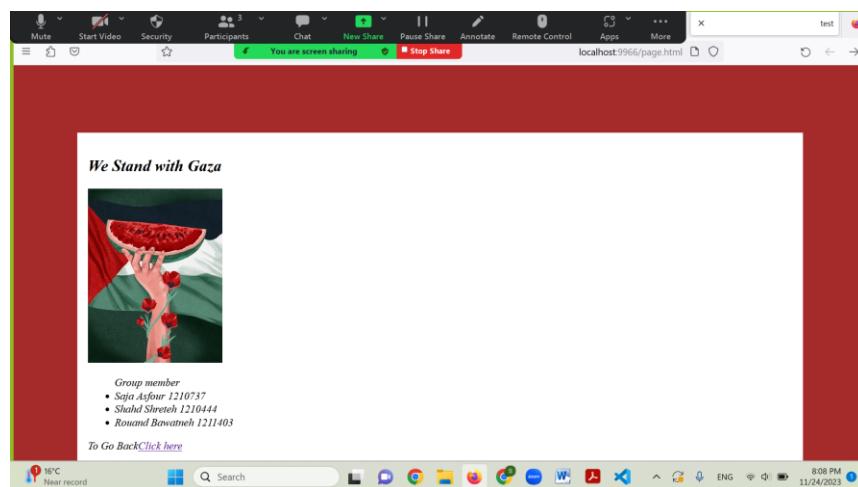


Figure 47:the page when request is page.html

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('127.0.0.1', 49183)
IP: 127.0.0.1,Port: 49183
*****
GET /page.html HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

*****
The request File is: page.html
response status: 200 OK

('127.0.0.1', 49184)
IP: 127.0.0.1,Port: 49184
*****
GET /p.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,*/*
Ln 33, Col 8 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live ⌂
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,*/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost:9966/page.html
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin

*****
The request File is: p.jpg
response status: 200 OK

('127.0.0.1', 49191)
IP: 127.0.0.1,Port: 49191
*****
GET /page.html HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Ln 33, Col 8 Spaces: 4 UTF-8 CRLF HTML ⚡ Go Live ⌂
```



The screenshot shows two terminal windows side-by-side. Both windows have tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The TERMINAL tab displays the following command-line session:

```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Pragma: no-cache
Cache-Control: no-cache

*****
The request File is: page.html
response status: 200 OK

('127.0.0.1', 49192)
IP: 127.0.0.1, Port: 49192
*****
GET /p.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: image/avif,image/webp,/*
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
DNT: 1
Connection: keep-alive
Referer: http://localhost:9966/page.html
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
Pragma: no-cache
Cache-Control: no-cache

```

Below the terminal window, the operating system taskbar is visible, showing various application icons and the system clock (8:10 PM, 11/24/2023).

Figure 48:the output when request is an .html file

4-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:

```
62 |     if request_type == 'GET':
63 |         #if the request is a .css file then the server should send the requested css file with Content-Type: text/css
64 |         elif '.css' in request_file:
65 |             connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
66 |             connectionSocket.send(f"Content-Type: text/css \r\n".encode())
67 |             connectionSocket.send(f"\r\n".encode())
68 |             print('response status: 200 OK\n')
69 |             f= open(str(request_file), 'rb')
70 |             connectionSocket.send(f.read())
71 |             f.close()
```

Figure 49:the code when request is an .css file

If we search for any .css file like(styles.css):

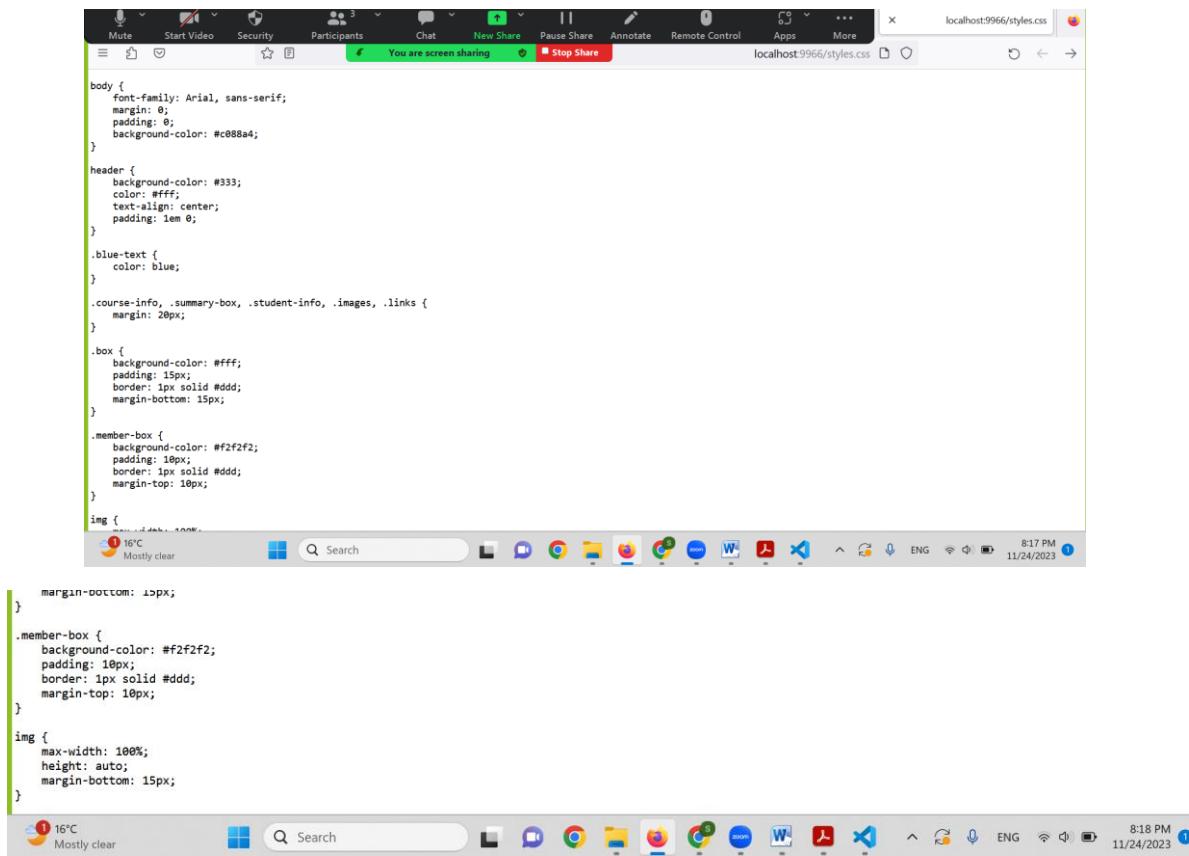


Figure 50:the page when request is styles.css file

The terminal window shows the following output:

```

PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('127.0.0.1', 49233)
IP: 127.0.0.1, Port: 49233
*****
GET /styles.css HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

*****
The request File is: styles.css
response status: 200 OK

```

The browser screenshot shows the URL `localhost:9966/styles.css` and the content of the CSS file.

Figure 51:the output when request is an .css file

**5-if the request is a .png then the server should send the png image with Content-Type: image/png. You can use any image.**

```

70 |         f.close()
71 |     #if the request is a .png then the server should send the png image with Content-Type: image/png.
72 |     elif '.png' in request_File:
73 |         connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
74 |         connectionSocket.send(f"Content-Type: image/png \r\n".encode())
75 |         connectionSocket.send(f"\r\n".encode())
76 |         print('response status: 200 OK\r\n')
77 |         f= open(str(request_File), 'rb')
78 |         connectionSocket.send(f.read())
79 |         f.close()

```

Figure 52:the code when request is an .png file

If we search for any .png (like sh.png):

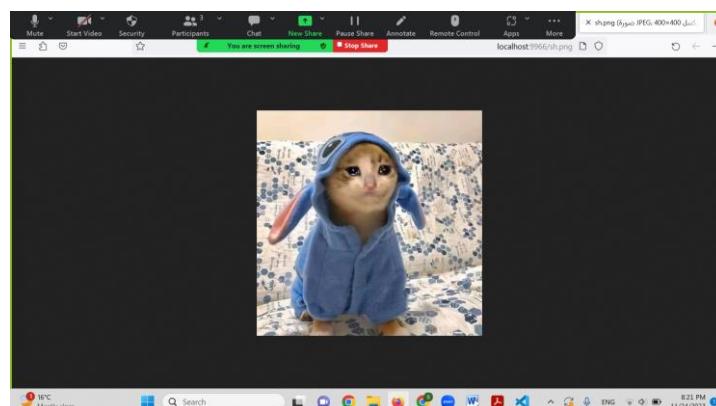


Figure 53:the page when request is sh.png

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('127.0.0.1', 49241)
IP: 127.0.0.1, Port: 49241
*****
GET /sh.png HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.3
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

*****
The request File is: sh.png
response status: 200 OK

```

Ln 63, Col 37 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit Go Live 8:22 PM 11/24/2023

Figure 54:the output when request is an .png file

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

```

80
81
82
83
84
85
86
87
88
    .close()
#6- if the request is a .jpg then the server should send the jpg image with Content-Type: image/jpeg.
elif '.jpg' in request_File:
    connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
    connectionSocket.send(f"Content-Type: image/jpeg \r\n".encode())
    connectionSocket.send(f"\r\n".encode())
    print('response status: 200 OK\n\n')
    f= open(str(request_File), 'rb')
    connectionSocket.send(f.read())
    f.close()

```

Figure 55:the code when request is an .jpg file

If we search for any .jpg (like s.jpg):

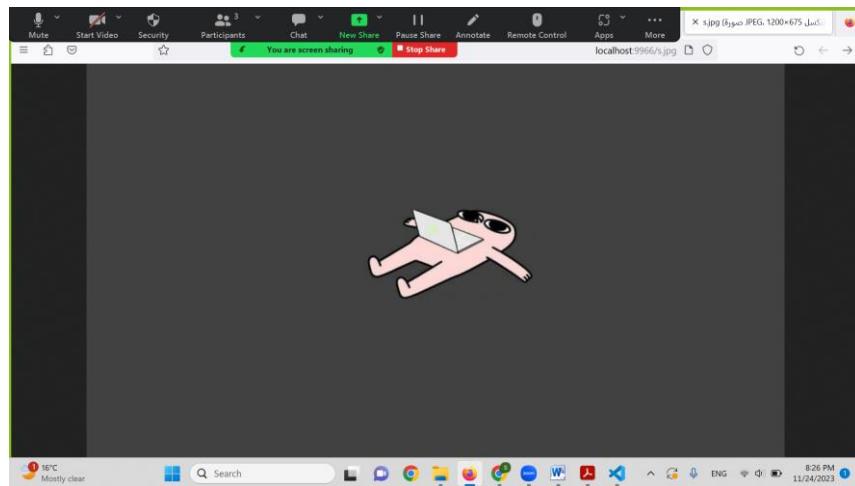


Figure 56:the page when request is s.jpg

And the output in the terminal:

The screenshot shows a terminal window with the following text output:

```
PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('127.0.0.1', 49253)
IP: 127.0.0.1, Port: 49253
*****
GET /s.jpg HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

*****
The request File is: s.jpg
response status: 200 OK
```

The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The status bar at the bottom shows Ln 95, Col 22, Spaces: 4, UTF-8, CRLF, Python 3.11.5 64-bit, Go Live, 8:26 PM, 11/24/2023.

Figure 57:the output when request is an .jpg file

7-Use the status code **307 Temporary Redirect** to redirect the following

- a. If the request is **/cr** then redirect to cornell.edu website
- b. If the request is **/so** then redirect to stackoverflow.com website
- c. If the request is **/rt** then redirect to ritaj website

/cr:

```
#If the request is /cr then redirect to cornell.edu website
elif request_File =='cr':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())
    connectionSocket.send("Location: https://www.cornell.edu/\r\n".encode())
```

Figure 58:the code when request is /cr

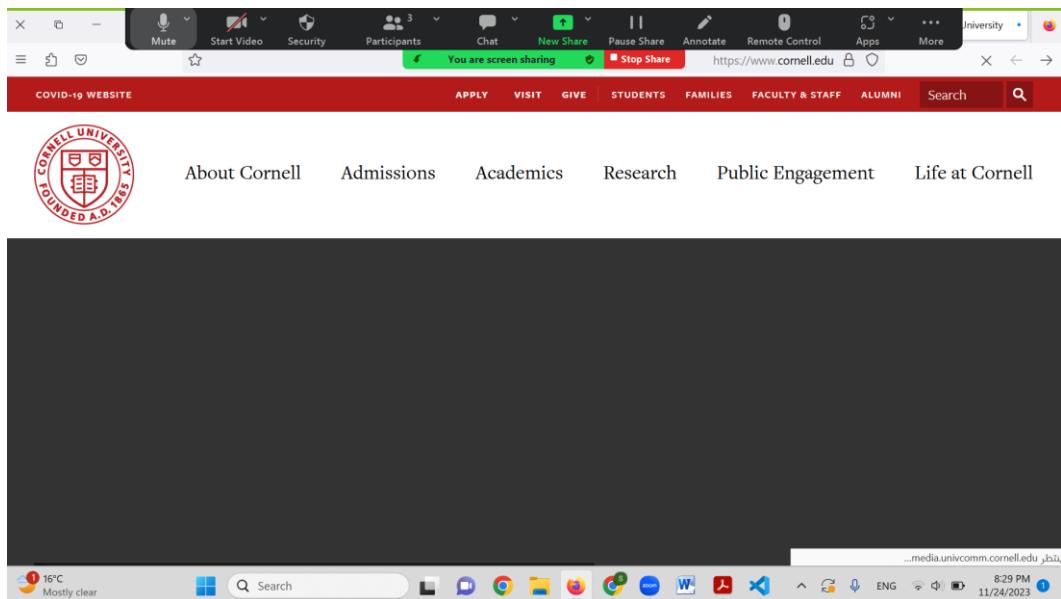


Figure 59:the page when request is /cr

The cornell website is opened(<https://www.cornell.edu/>)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('127.0.0.1', 49259)
IP: 127.0.0.1,Port: 49259
*****
GET /cr HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

*****
The request File is: cr

```

Figure 60:the output when request is /cr

/so:

```
93 |     #If the request is /so then redirect to stackoverflow.com website
94 |     elif request_File =='so':
95 |         connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())
96 |         connectionSocket.send("Location: https://stackoverflow.com\r\n".encode())
```

Figure 61:the code when request is /so

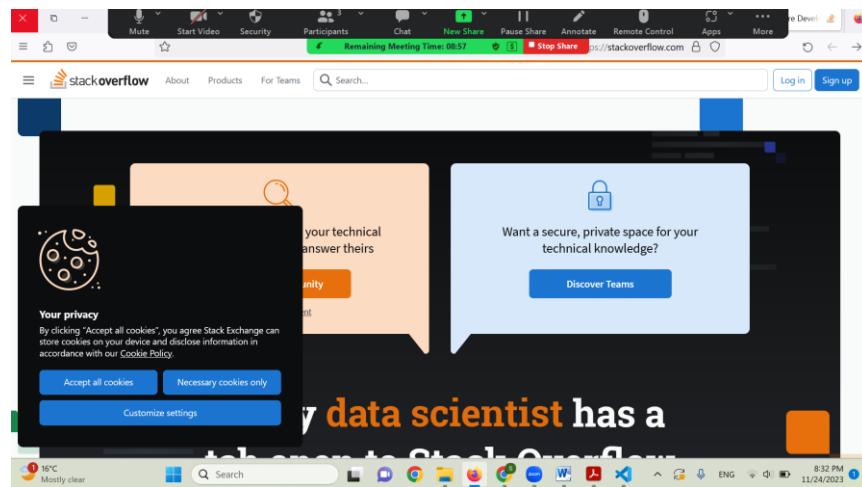


Figure 62:the page when request is /so

The stackoverflow.com website (<https://stackoverflow.com/>) is opened

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('127.0.0.1', 49315)
IP: 127.0.0.1, Port: 49315
*****
GET /so HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

*****
The request File is: so
```

Figure 63:the output when request is /so

/rt:

```
97     #If the request is /rt then redirect to ritaj website
98     elif request_file =='rt':
99         connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())
100        connectionSocket.send("Location: https://ritaj.birzeit.edu/register/\r\n".encode())
```

Figure 64:the code when request is /rt



Figure 65:the page when request is /rt

The ritaj website <https://ritaj.birzeit.edu/register/> is opened

```
PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('127.0.0.1', 49350)
IP: 127.0.0.1, Port: 49350
*****
GET /rt HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

*****
The request File is: rt
```

Figure 66:the output when request is /rt

- 8- 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 404" 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

```

else:
    raise Exception('Not found')
#if the file is not found in the project
except Exception as e:
    connectionSocket.send(f"HTTP/1.1 404 Not Found\r\n".encode())
    connectionSocket.send(f"Content-Type: text/html \r\n".encode())
    connectionSocket.send(f"\r\n".encode())
    print(request_File + "test")
    print('\b\bResponse status: 404 Not Found')
    f='<!DOCTYPE html><html>' \
    '<style>*{ text-align: center; }' \
    '#Error{ color: red;}#name{ font-weight: bold;}</style>' \
    '<head> <title>Error 404</title></head>' \
    '<body> <div id="Error"> <h1>The file is not found</h1> </div>' \
    '<hr> <div id="name"> <p>Saja Asfour - 1210737</p> <p>Shahd Shreneh - 1210444</p>' \
    '<p>Rouand Bawatneh -1211403</p> </div><hr> <div>' \
    '<p> Ip Adress: ' + str(IP) + ', Port Number: ' +str(port)+\
    '</p> </div> </body></html>'
    connectionSocket.send(f.encode())

```

Figure 67:the code when request is wrong or request does not exist

If we search for anything not of our cases like /s :

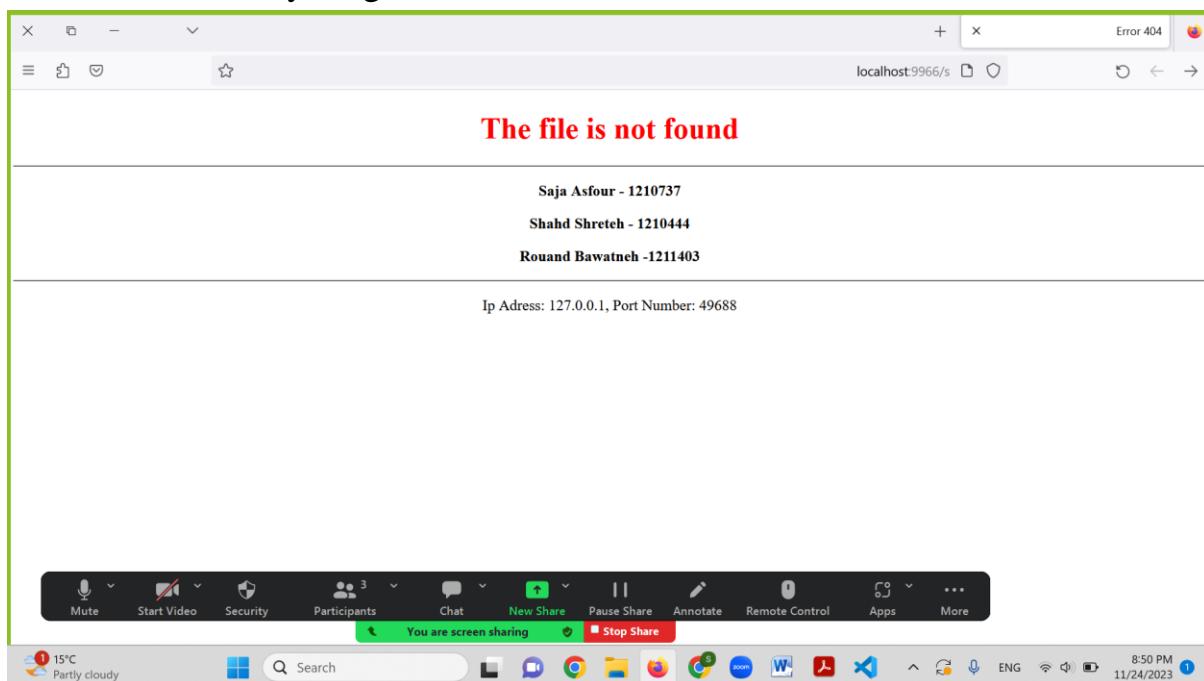


Figure 68:the page when request is wrong or request does not exist

The terminal window shows the output of a Python script running on port 49688. The server is ready to receive requests. A client request for 's' is received, and the response status is 404 Not Found.

```
PS C:\Users\DELL\Desktop\network\project1\codes> python -u "c:\Users\DELL\Desktop\network\project1\codes\part3.py"
The server is ready to receive
('127.0.0.1', 49688)
IP: 127.0.0.1,Port: 49688
*****
GET /s HTTP/1.1
Host: localhost:9966
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:120.0) Gecko/20100101 Firefox/120.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: ar,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1

*****
The request File is: s
s
Response status: 404 Not Found
```

The browser screenshot shows a 404 Not Found error page with the message 'The requested URL was not found on this server.'

Figure 69:the output when request is wrong or request does not exist

## python code :

The image shows three vertically stacked terminal windows from a code editor, likely Visual Studio Code, running on a Windows operating system. Each window displays a Python script named `part3.py` and its execution output.

**Top Terminal:**

```
1  from socket import *
2  #server port,socket initialization and type
3  portNum=9966
4  #creating a TCP socket for incoming request
5  serverSocket = socket(AF_INET,SOCK_STREAM)
6  #associate the server port number with this socket
7  serverSocket.bind(('',portNum))
8  #the server listen for TCP connection requests from the client with i queued connections
9  serverSocket.listen(1)
10 #print a message to tell the client that the server is ready to receive
11 print ('The server is ready to receive')
12 #start getting requests:
13 while True:
14     #when a client sends a TCP connection requests
15     connectionSocket, address = serverSocket.accept()
16     #create "connectionSocket" dedicated to this client
17     sent=connectionSocket.recv(2048).decode()
18     print(address)
19     IP= address[0]
20     port=address[1]
21     print("IP: "+ str(IP) +",Port: "+ str(port))
22     print("*****")
23     print(sent)
24     print("*****")
25     #if the sentence is not empty, the requested file is gotten from request header
```

**Middle Terminal:**

```
('127.0.0.1', 49688)
IP: 127.0.0.1,Port: 49688
```

**Bottom Terminal:**

```
Ln 123, Col 29  Spaces: 4  UTF-8  CRLF  { Python  3.11.5 64-bit  ⚡ Go Live  8:56 PM  11/24/2023
```

```
25  #if the sentence is not empty, the requested file is gotten from request header
26  if sent !='':
27      request_File=sent.split(' ')[1].replace('/','')
28      print("The request File is: "+request_File)
29  else:
30      #if the request is empty the connection is closed
31      connectionSocket.close()
32      continue
33  #exception in case the request file is not found
34  try:
35      #if the request is index.html/main.html or any file is found in the project file
36      #if the file is not found then exception is raised
37      #if the requested file is main.html or index.html
38      if request_File == '' or request_File=='main_en.html' or request_File== 'index.html' or request_File =='en':
39          connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
40          connectionSocket.send(f"Content-Type: text/html \r\n".encode())
41          connectionSocket.send(f"\r\n".encode())
42          mhtml=open('main_en.html' , 'rb')
43          connectionSocket.send(mhtml.read())
44          mhtml.close()
45          #If the request is /ar then the server response with main_ar.html which is an Arabic version of main_en.html
46          elif request_File=='ar':
47              connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
48              connectionSocket.send(f"Content-Type: text/html \r\n".encode())
49              connectionSocket.send(f"\r\n".encode())
```

**Bottom Status Bar:**

```
Ln 48, Col 64  Spaces: 4  UTF-8  CRLF  { Python  3.11.5 64-bit  ⚡ Go Live  8:56 PM  11/24/2023
```

```
part3.py > ...
47 connectionSocket.send(f"HTTP/1.1 200 OK\r\n\r\n".encode())
48 connectionSocket.send(f"Content-Type: text/html \r\n\r\n".encode())
49 connectionSocket.send(f"\r\n\r\n".encode())
50 mhtml=open('main_ar.html' , 'rb')
51 connectionSocket.send(mhtml.read())
52 mhtml.close()
53 #if the request is an .html file then the server should send the requested html file with Content-Type: text/
54 elif '.html' in request_File:
55     connectionSocket.send(f"HTTP/1.1 200 OK\r\n\r\n".encode())
56     connectionSocket.send(f"Content-Type: text/html \r\n\r\n".encode())
57     connectionSocket.send(f"\r\n\r\n".encode())
58     print('response status: 200 OK\r\n\r\n')
59     f= open(str(request_File), 'rb')
60     connectionSocket.send(f.read())
61     f.close()
62 #if the request is a .css file then the server should send the requested css file with Content-Type: text/css
63 elif '.css' in request_File:
64     connectionSocket.send(f"HTTP/1.1 200 OK\r\n\r\n".encode())
65     connectionSocket.send(f"Content-Type: text/css \r\n\r\n".encode())
66     connectionSocket.send(f"\r\n\r\n".encode())
67     print('response status: 200 OK\r\n\r\n')
68     f= open(str(request_File), 'rb')
69     connectionSocket.send(f.read())
70     f.close()
71 #if the request is a .png then the server should send the png image with Content-Type: image/png.
```

part3.py > ...

```
71 #if the request is a .png then the server should send the png image with Content-Type: image/png.
72 elif '.png' in request_File:
73     connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
74     connectionSocket.send(f"Content-Type: image/png \r\n".encode())
75     connectionSocket.send(f"\r\n".encode())
76     print('response status: 200 OK\n\n')
77     f= open(str(request_File), 'rb')
78     connectionSocket.send(f.read())
79     f.close()
80 #6- if the request is a .jpg then the server should send the jpg image with Content-Type: image/jpeg.
81 elif '.jpg' in request_File:
82     connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
83     connectionSocket.send(f"Content-Type: image/jpeg \r\n".encode())
84     connectionSocket.send(f"\r\n".encode())
85     print('response status: 200 OK\n\n')
86     f= open(str(request_File), 'rb')
87     connectionSocket.send(f.read())
88     f.close()
89 #If the request is /cr then redirect to cornell.edu website
90 elif request_File == 'cr':
91     connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())
92     connectionSocket.send("Location: https://www.cornell.edu/\r\n".encode())
93 #If the request is /so then redirect to stackoverflow.com website
94 elif request_File == 'so':
95     connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + × ... ^ ×

('127.0.0.1', 49688)
IP: 127.0.0.1,Port: 49688
```

```

part3.py > ...
94 elif request_File == 'so':
95     connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())
96     connectionSocket.send("Location: https://stackoverflow.com/\r\n".encode())
97 #If the request is /rt then redirect to ritaj website
98 elif request_File == 'rt':
99     connectionSocket.send("HTTP/1.1 307 Temporary Redirect\r\n".encode())
100    connectionSocket.send("Location: https://ritaj.birzeit.edu/register/\r\n".encode())
101 #this is a handler only in order not to get not found error
102 elif 'favicon.ico' ==request_File:
103     print()
104 else:
105     raise Exception('Not found')
106 #if the file is not found in the project
107 except Exception as e:
108     connectionSocket.send(f"HTTP/1.1 404 Not Found\r\n".encode())
109     connectionSocket.send(f"Content-Type: text/html \r\n".encode())
110     connectionSocket.send(f"\r\n".encode())
111     print(request_File +"test")
112     print('b\bResponse status: 404 Not Found')
113     f=<!DOCTYPE html><html> \
114     '<style>{ text-align: center; }' \
115     '#Error{ color: red; }#name{ font-weight: bold; }</style>' \
116     '<head> <title>Error 404</title></head> \
117     '<body> <div id="Error"> <h1>The file is not found</h1> </div>' \
118     '<hr> <div id="name"> <p>Saja Asfour - 1210737</p> <p>Shahd Shreteh - 1210444</p>' \
119     '<p>Rouand Bawatneh -1211403</p> </div><hr> <div>' \
120     '<p> Ip Adress: ' + str(IP)+ ', Port Number: ' +str(port)+\
121     '</p> </div> </body></html>'
122     connectionSocket.send(f.encode())
123     connectionSocket.close()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

('127.0.0.1', 49688)  
IP: 127.0.0.1,Port: 49688

Ln 117, Col 64 Spaces: 4 UTF-8 CRLF { Python 3.11.5 64-bit ⚡ Go Live ⌂

powershell Code

8:58 PM 11/24/2023

```

part3.py > ...
100 |     connectionSocket.send("Location: https://ritaj.birzeit.edu/register/\r\n".encode())
101 #this is a handler only in order not to get not found error
102 elif 'favicon.ico' ==request_File:
103     print()
104 else:
105     raise Exception('Not found')
106 #if the file is not found in the project
107 except Exception as e:
108     connectionSocket.send(f"HTTP/1.1 404 Not Found\r\n".encode())
109     connectionSocket.send(f"Content-Type: text/html \r\n".encode())
110     connectionSocket.send(f"\r\n".encode())
111     print(request_File +"test")
112     print('b\bResponse status: 404 Not Found')
113     f=<!DOCTYPE html><html> \
114     '<style>{ text-align: center; }' \
115     '#Error{ color: red; }#name{ font-weight: bold; }</style>' \
116     '<head> <title>Error 404</title></head> \
117     '<body> <div id="Error"> <h1>The file is not found</h1> </div>' \
118     '<hr> <div id="name"> <p>Saja Asfour - 1210737</p> <p>Shahd Shreteh - 1210444</p>' \
119     '<p>Rouand Bawatneh -1211403</p> </div><hr> <div>' \
120     '<p> Ip Adress: ' + str(IP)+ ', Port Number: ' +str(port)+\
121     '</p> </div> </body></html>'
122     connectionSocket.send(f.encode())
123     connectionSocket.close()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

('127.0.0.1', 49688)  
IP: 127.0.0.1,Port: 49688

Ln 123, Col 29 Spaces: 4 UTF-8 CRLF { Python 3.11.5 64-bit ⚡ Go Live ⌂

powershell Code

8:58 PM 11/24/2023

Figure 70:the python code for part three

**Conclusion :**

in this project we learn processes on different hosts communicate through network applications and we tried to write simple applications in python to understand how connection happens on TCP protocol and we learn how to make web page by html and css.