



Department of Electrical and Computer
Engineering

ENCS3320-Computer Networks

Network Project #1

PREPARED BY

Ayham Maree 1191408

Sara Ammar 1191052

Alaa Sehwail 1191741

Mahmoud Shoman 1191648

INTRODUCTION

Abdalkarim Awad

SECTION 3

Date 19/11/2021

Contents

Abstract	3
Part 1:	4
Part 2	6
Part 3	8
Part 4	27

Abstract

In this project, the aim is to learn socket programming and how to get http response, especially how to write html and css code and make a network between server and client using port number, get the IP addresses of the devices and run the server using these IP Addresses that we use ipconfig in CMD to get the IP Address to the routing device to make the operation easier than connect the device using ethernet that is the aim of the project and we use a various programming language such as Java, Python etc.

Part 1:

1. Ping a device in the same network

The command “arp -a” was used to list all devices that are connected to the network with their IP address.

```
Microsoft Windows [Version 10.0.19043.1288]
c) Microsoft Corporation. All rights reserved.

:C:\Users\ADMIN>arp -a

Interface: 192.168.1.71 --- 0x4
Internet Address      Physical Address      Type
192.168.1.65          90-94-97-6a-a0-d7    dynamic
192.168.1.70          d0-ff-98-3c-44-d9    dynamic
192.168.1.72          d2-60-22-3c-87-1d    dynamic
192.168.1.76          ea-dd-9b-dd-22-bd    dynamic
192.168.1.95          f0-86-20-b5-c7-54    dynamic
192.168.1.254         dc-d9-ae-b0-83-58    dynamic
192.168.1.255         ff-ff-ff-ff-ff-ff    static
224.0.0.22             01-00-5e-00-00-16    static
224.0.0.251            01-00-5e-00-00-fb    static
224.0.0.252            01-00-5e-00-00-fc    static
239.255.255.250       01-00-5e-7f-ff-fa    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static

Interface: 192.168.56.1 --- 0xb
Internet Address      Physical Address      Type
192.168.56.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22             01-00-5e-00-00-16    static
224.0.0.251            01-00-5e-00-00-fb    static
224.0.0.252            01-00-5e-00-00-fc    static
239.255.255.250       01-00-5e-7f-ff-fa    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static
```

Figure 1: arp -a Command

Then, the command “ping” was used to ping one of the devices that were listed. It shows the amount the amount of packets sent. As a result, it shows the amount of packets received, the IP it was received from, the time-to-live (TTL) which is the amount of hops the packets did before exiting the network, and lastly the time it took to be received.

```
Microsoft Windows [Version 10.0.19043.1288]
c) Microsoft Corporation. All rights reserved.

:C:\Users\ADMIN>ping 192.168.1.65

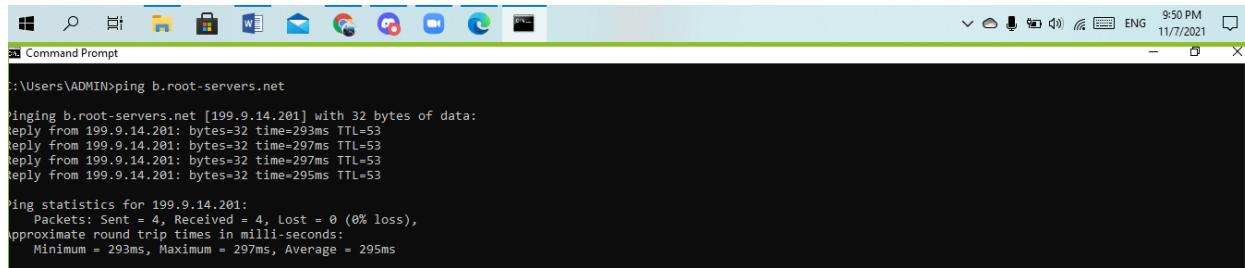
Pinging 192.168.1.65 with 32 bytes of data:
Reply from 192.168.1.65: bytes=32 time=247ms TTL=64
Reply from 192.168.1.65: bytes=32 time=204ms TTL=64
Reply from 192.168.1.65: bytes=32 time=75ms TTL=64
Reply from 192.168.1.65: bytes=32 time=100ms TTL=64

Ping statistics for 192.168.1.65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 75ms, Maximum = 247ms, Average = 156ms

:C:\Users\ADMIN>
```

Figure 2: ping Command - Device on the same Network

2-ping b.root-servers.net



```
C:\Users\ADMTN>ping b.root-servers.net

Pinging b.root-servers.net [199.9.14.201] with 32 bytes of data:
Reply from 199.9.14.201: bytes=32 time=293ms TTL=53
Reply from 199.9.14.201: bytes=32 time=297ms TTL=53
Reply from 199.9.14.201: bytes=32 time=297ms TTL=53
Reply from 199.9.14.201: bytes=32 time=295ms TTL=53

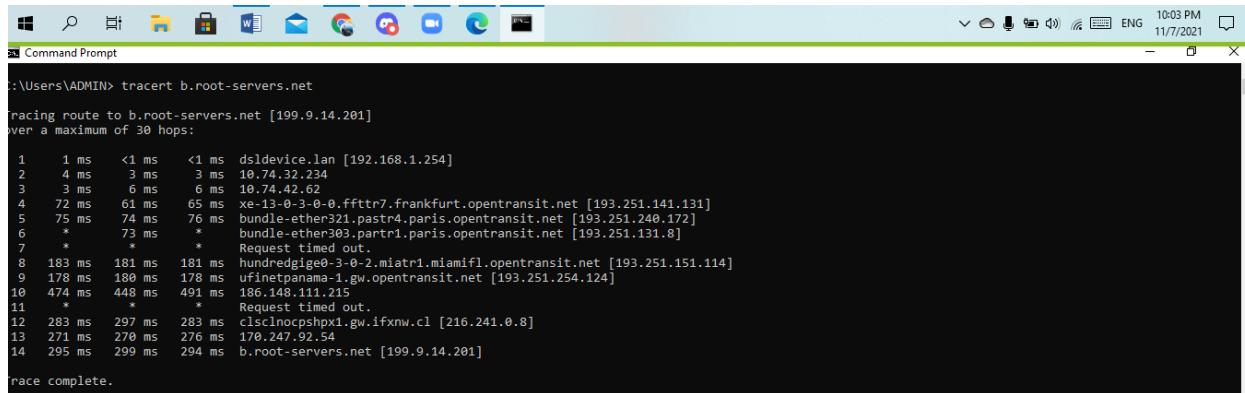
Ping statistics for 199.9.14.201:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    approximate round trip times in milli-seconds:
        Minimum = 293ms, Maximum = 297ms, Average = 295ms
```

Figure 3: ping Command - Online Website

3-tracert b.root-servers.net

Tracert, Traceroute or Trace Route, is a command-line tool that allows you to use a set of computer network diagnostic commands to troubleshoot TCP/IP problems. Tracert CMD can trace the path that an Internet Protocol (IP) packet takes to its destination.

Tracert tests the different paths taken by sent packets to reach the destination from source. The result consists of 5 columns, the first being the hop number (TTL). Tracert actually sends 3 packets, so the 3 columns after (TTL) are the time it takes for the packets to make each hop. The last column is the server at the specified hop. In our results, it took us 299 ms to retrieve data from the destination server as shown in the last hop



```
C:\Users\ADMIN> tracert b.root-servers.net

Tracing route to b.root-servers.net [199.9.14.201]
over a maximum of 30 hops:
1  1 ms    <1 ms    <1 ms  dslddevice.lan [192.168.1.254]
2  4 ms    3 ms    3 ms  10.74.32.234
3  3 ms    6 ms    6 ms  10.74.42.62
4  72 ms   61 ms   65 ms  xe-13-0-3-0-0.ffftr7.frankfurt.opentransit.net [193.251.141.131]
5  75 ms   74 ms   76 ms  bundle-ether321.pastr4.paris.opentransit.net [193.251.240.172]
6  *        73 ms   *      bundle-ether303.partr1.paris.opentransit.net [193.251.131.8]
7  *        *        *      Request timed out.
8  183 ms   181 ms   181 ms  hundredigge0-3-0-2.miatri.miamifl.opentransit.net [193.251.151.114]
9  178 ms   188 ms   178 ms  ufinetpanama-1.gw.opentransit.net [193.251.254.124]
10 474 ms   448 ms   491 ms  186.148.111.215
11  *        *        *      Request timed out.
12 283 ms   297 ms   283 ms  c1sc1norcpshpx1.gw.ifxnw.c1 [216.241.0.8]
13 271 ms   270 ms   276 ms  170.247.92.54
14 295 ms   299 ms   294 ms  b.root-servers.net [199.9.14.201]

Trace complete.
```

Figure 4: tracert Command

4-nslookup b.root-servers.net

Nslookup (stands for “Name Server Lookup”) is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS related problems.

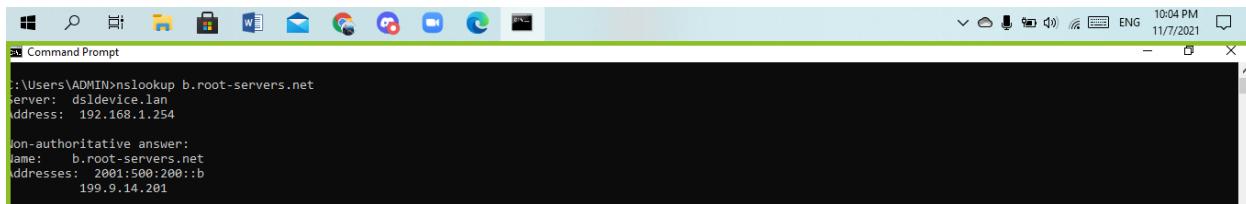


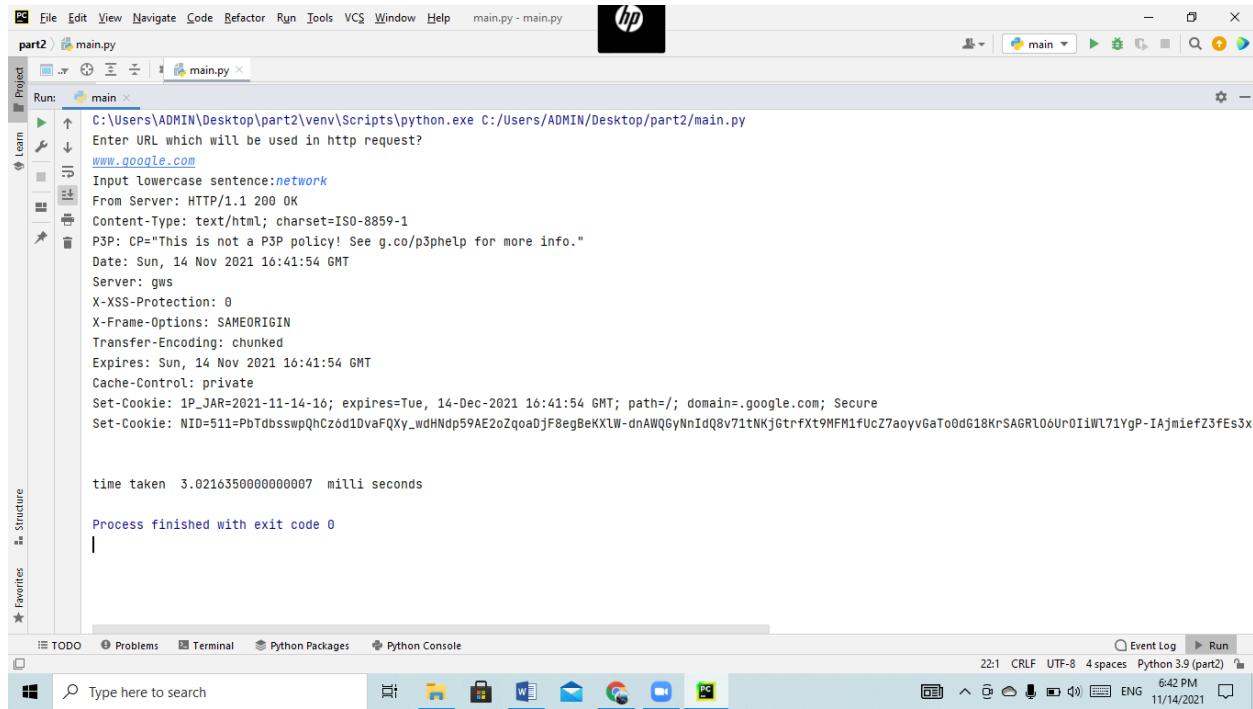
Figure 5: nslookup Command

Part 2:

Code:

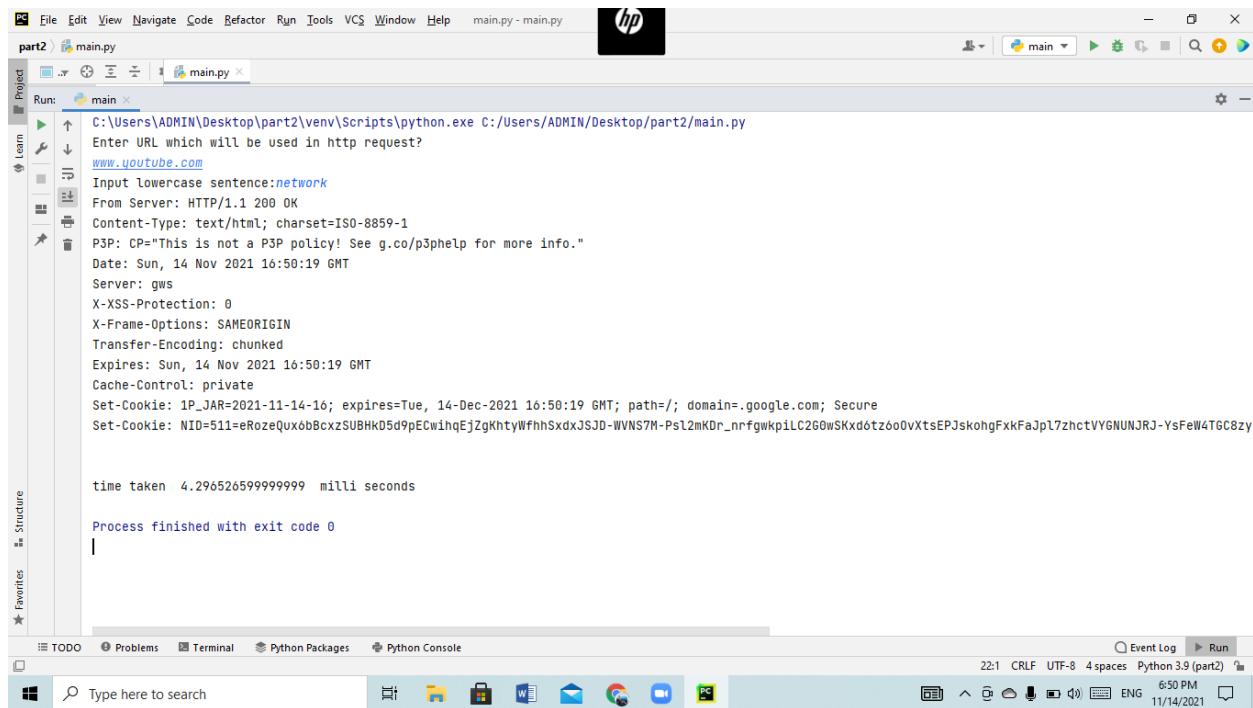
```
from socket import *
import timeit
# Agham Maree - 1191408
# Sara Ammar - 1191052
# Alaa Sehwil - 1191741
# Mahmoud Shouman -1191648
#####
serverName = input("Enter URL which will be used in http request?\n") #get the url of the website from user
name = "Hostname:" + str(serverName) + "\r\n\r\n" #convert the url to string hostname
serverPort=80 # integer that is the port number we use
clientSocket = socket(AF_INET,SOCK_STREAM) #make a socket and give it the name clientsocket
start = timeit.default_timer() # function to start time from the time we send the request to conect to the server
clientSocket.connect((serverName,serverPort)) #connect between the clientsocket and serversocket
sentence =input("Input lowercase sentence:") #input from user the message we will send to the website
clientSocket.send("HEAD / HTTP/1.1 \r\n".encode()) #send the head of http response after encode it
clientSocket.send(sentence.encode()) #send the message we need as a string
clientSocket.send(name.encode()) #send the server name (hostname) to the socket after encode it
modifiedSentence = clientSocket.recv(1024) #modify the sentence by the famous client proxy 1024
print("From Server:",modifiedSentence.decode()) # decode the message and get the http response after networking between client and server using socket
print ('time taken ', timeit.default_timer()-start,' seconds') #print the spent time of the operation
clientSocket.close() #close the client
#####
```

Head Http Response:



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help. The current file is main.py. The Run tab is selected, showing the command: C:\Users\ADMIN\Desktop\part2\venv\Scripts\python.exe C:/Users/ADMIN/Desktop/part2/main.py. Below this, it says "Enter URL which will be used in http request?" followed by "www.google.com". The code output pane displays the HTTP response for Google. It includes headers like Content-Type, Date, Server, and various cookies. The response body shows the HTML content of the Google homepage. At the bottom, it says "time taken 3.0216350000000007 milli seconds" and "Process finished with exit code 0".

Figure (6): head http response for Google Website



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help. The current file is main.py. The Run tab is selected, showing the command: C:\Users\ADMIN\Desktop\part2\venv\Scripts\python.exe C:/Users/ADMIN/Desktop/part2/main.py. Below this, it says "Enter URL which will be used in http request?" followed by "www.youtube.com". The code output pane displays the HTTP response for YouTube. It includes headers like Content-Type, Date, Server, and various cookies. The response body shows the HTML content of the YouTube homepage. At the bottom, it says "time taken 4.290526599999999 milli seconds" and "Process finished with exit code 0".

Figure (7): head http response for YouTube Website

Part 3:

Screenshots:

Main Page (localhost:6500, localhost:6500/, localhost:6500/index.html, localhost:5000/main.html)

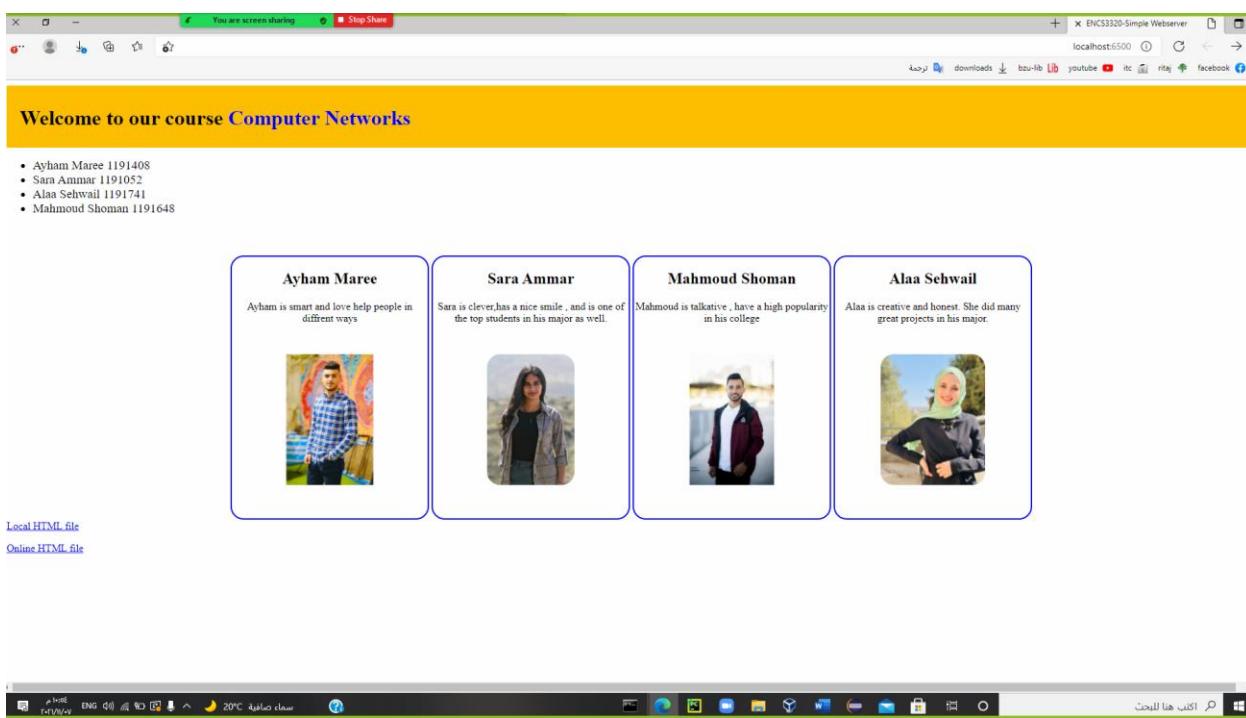


Figure 8.1: localhost:6500 Browser Window

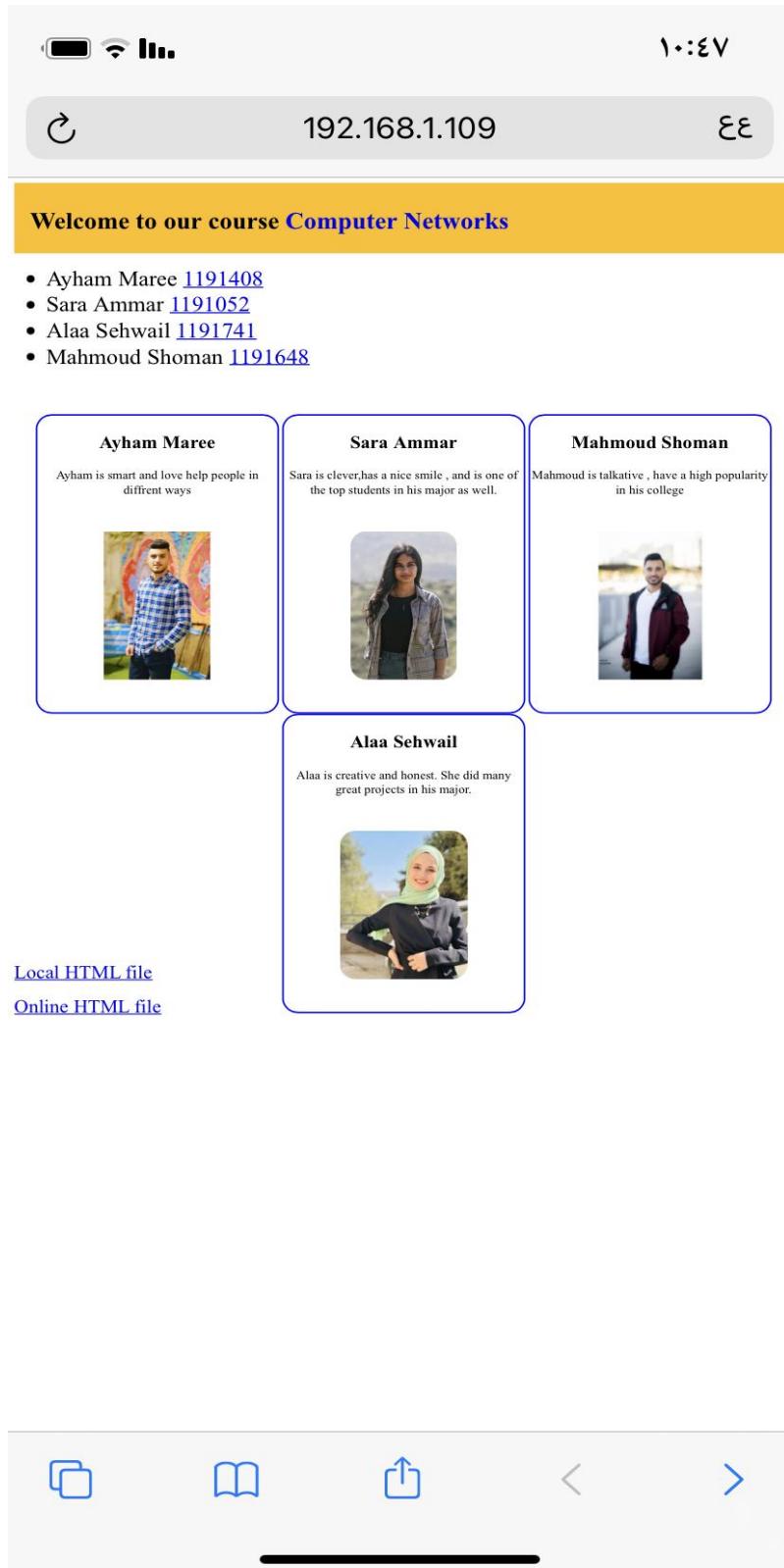


Figure 8.2: 192.168.109.1:6500 Browser Window use phone

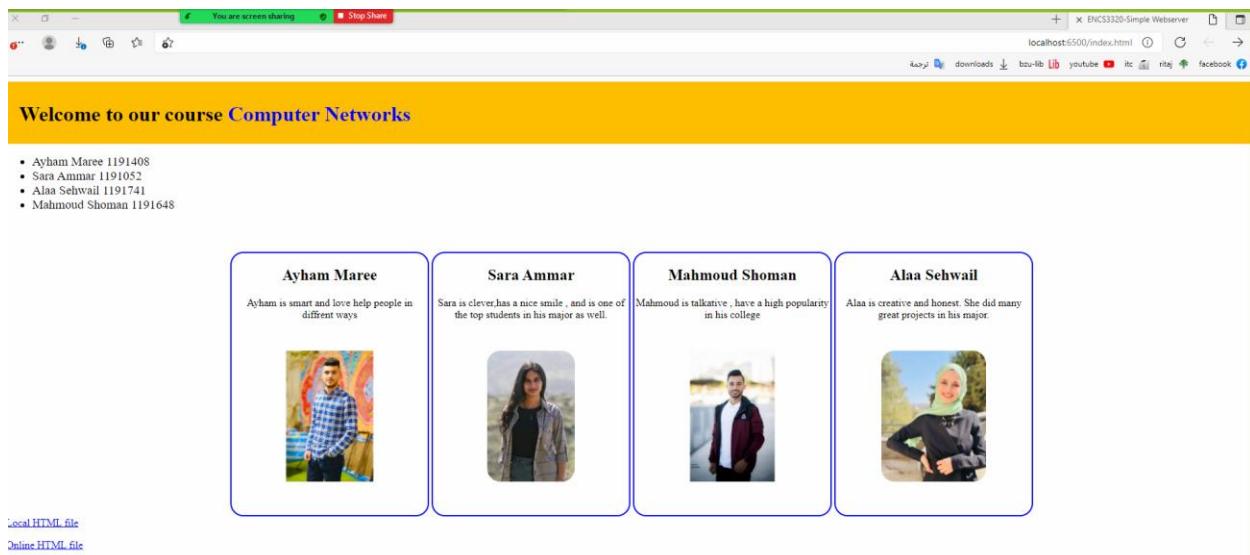


Figure 9: localhost:6500/index.html Browser Window

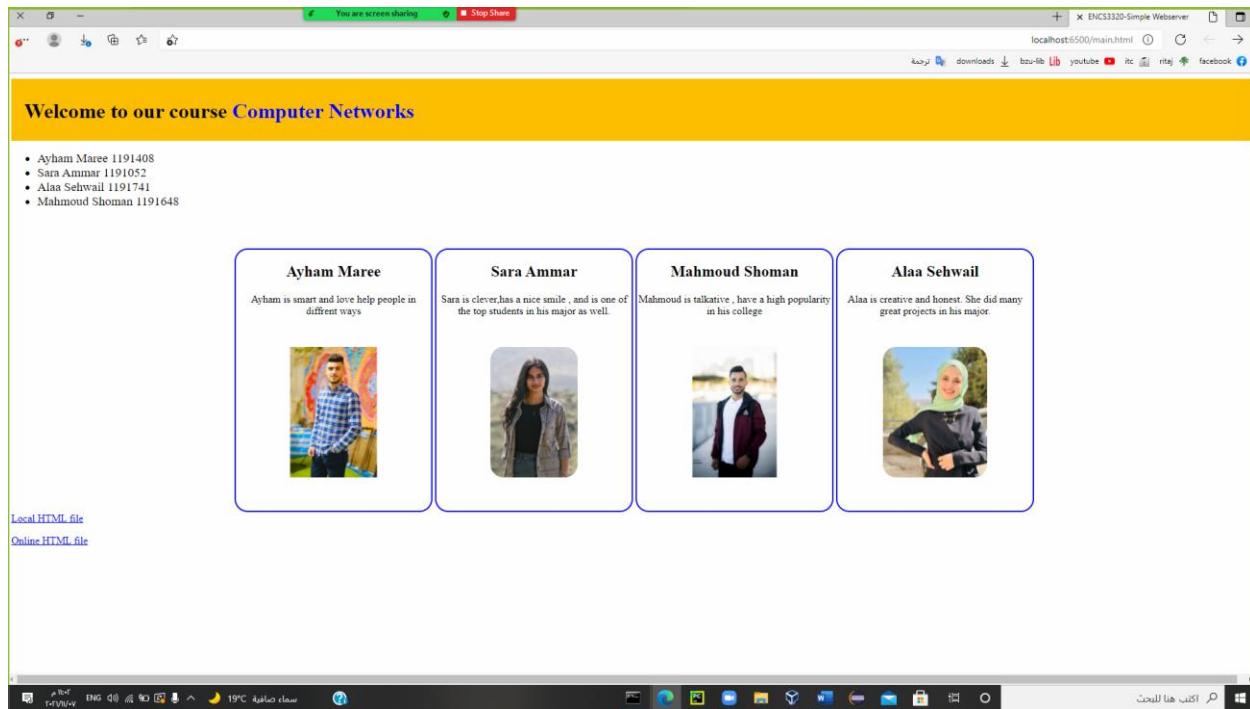
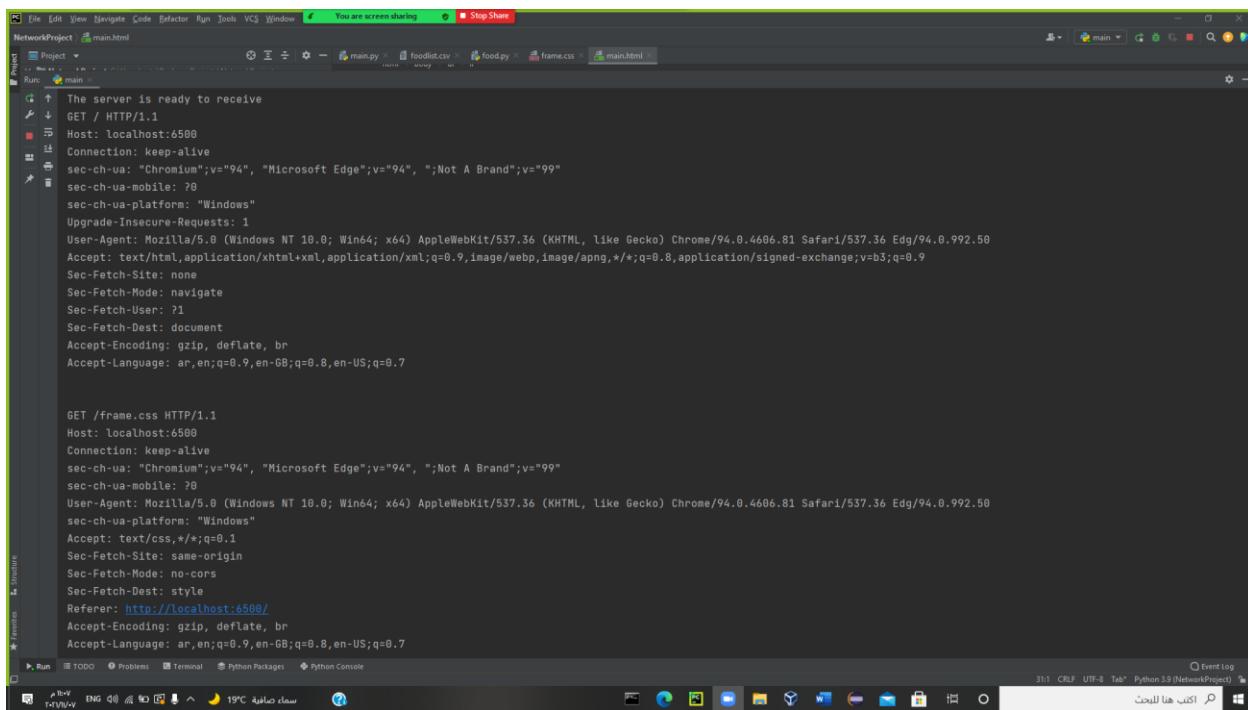


Figure 10: localhost:6500/main.html Browser Window

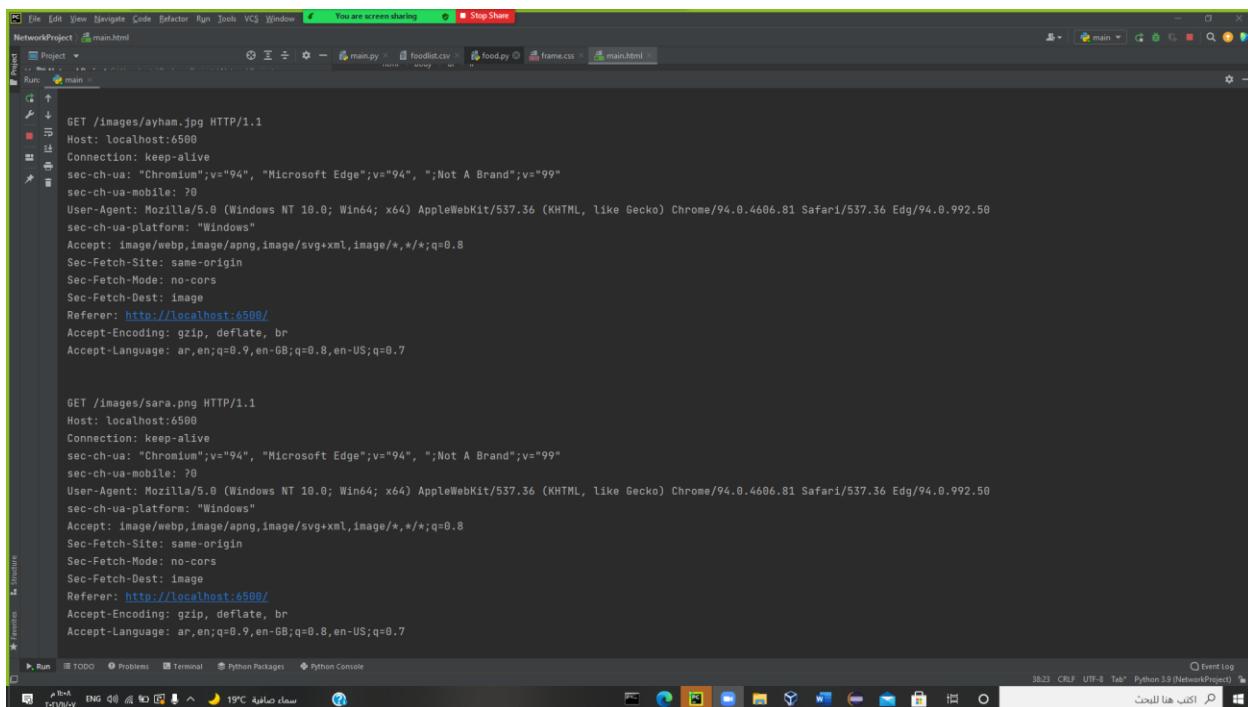


```
File Edit View Navigate Code Refactor Run Tools VCS Window You are screen sharing Stop Share
NetworkProject main.html
Project Run main
GET / HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b5;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7

GET /frame.css HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
sec-ch-ua-platform: "Windows"
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7

Event Log 31/1 CR LF UTF-8 Tab* Python 3.9 (NetworkProject)
Run TODO Problems Terminal Python Packages Python Console
19°C اسكندرية ENG QWERTY
أكتب هنا للبحث
```

Figure 11: Main Page HTTP Requests Printed on Command Line (1)

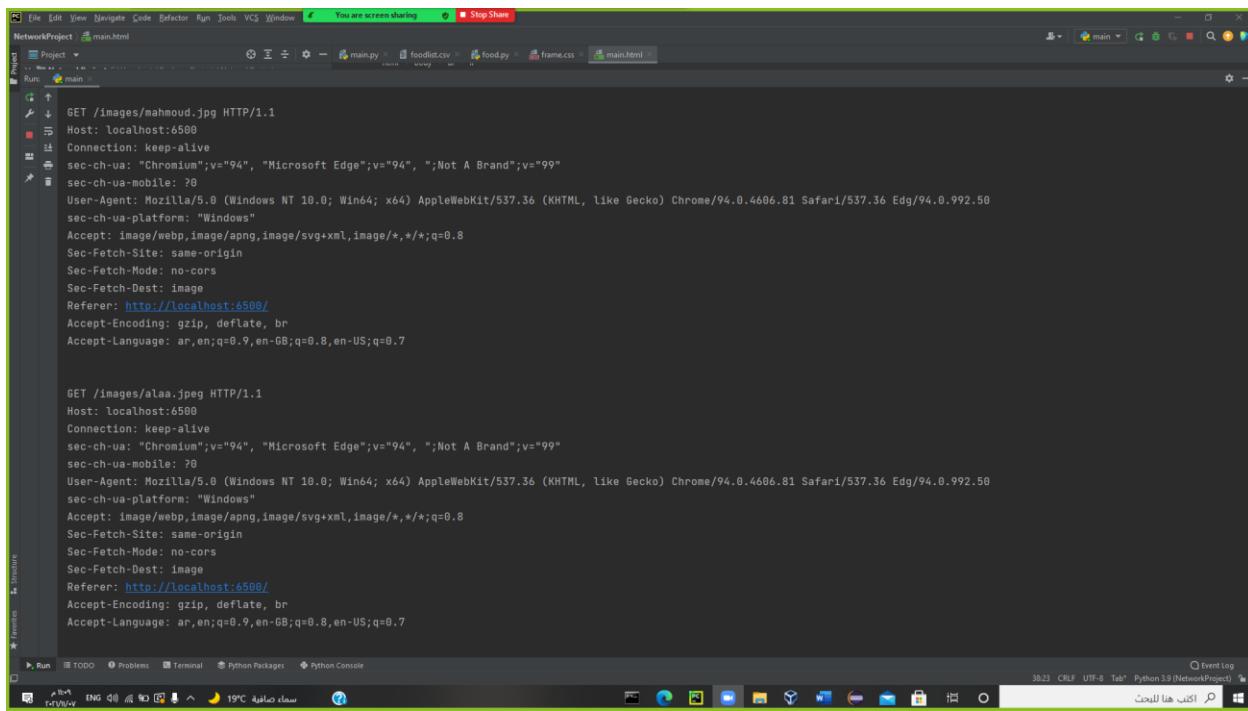


```
File Edit View Navigate Code Refactor Run Tools VCS Window You are screen sharing Stop Share
NetworkProject main.html
Project Run main
GET /images/ayham.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7

GET /images/sara.png HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7

Event Log 38/23 CR LF UTF-8 Tab* Python 3.9 (NetworkProject)
Run TODO Problems Terminal Python Packages Python Console
19°C اسكندرية ENG QWERTY
أكتب هنا للبحث
```

Figure 12: Main Page HTTP Requests Printed on Command Line (2)

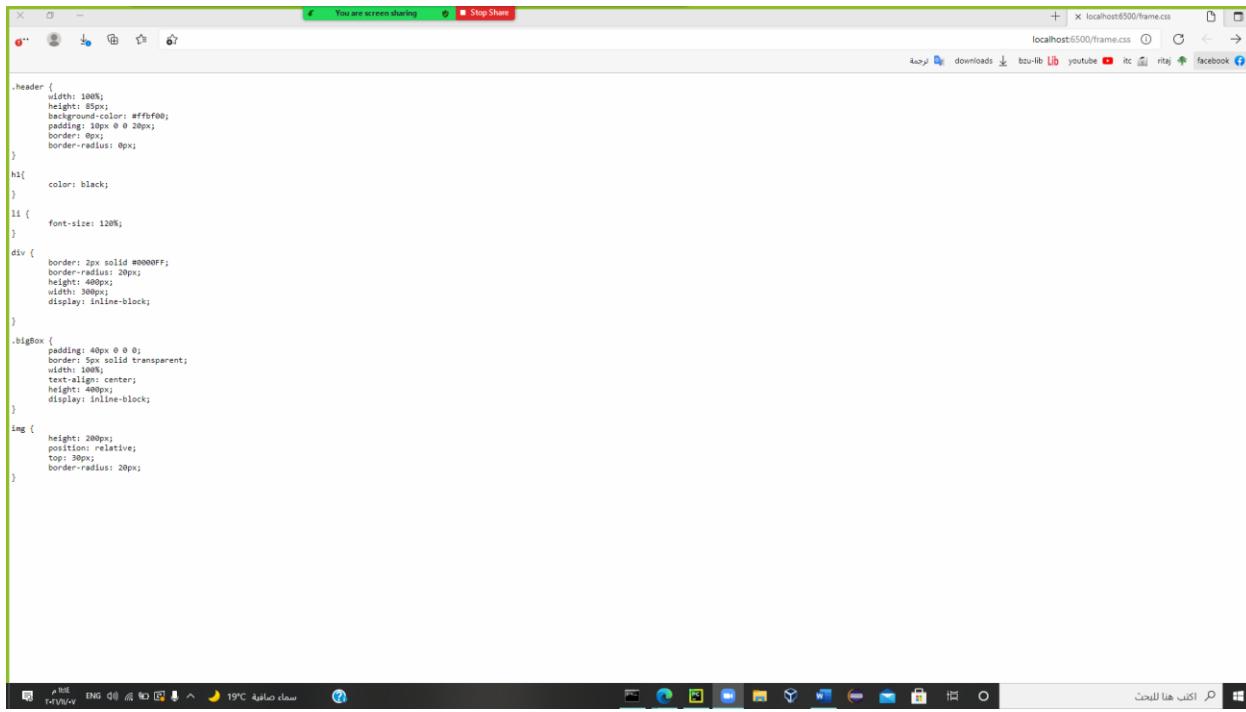


```
GET /images/mahmoud.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7

GET /images/alaa.jpeg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
```

Figure 13: Main Page HTTP Requests Printed on Command Line (3)

CSS File:



A screenshot of a Microsoft Edge browser window. The address bar shows "localhost:6500/frame.css". The page content displays the following CSS code:

```
.header { width: 100%; height: 45px; background-color: #ffbf00; padding: 10px 0 20px; border: 0px; border-radius: 0px; }

h1{ color: black; }

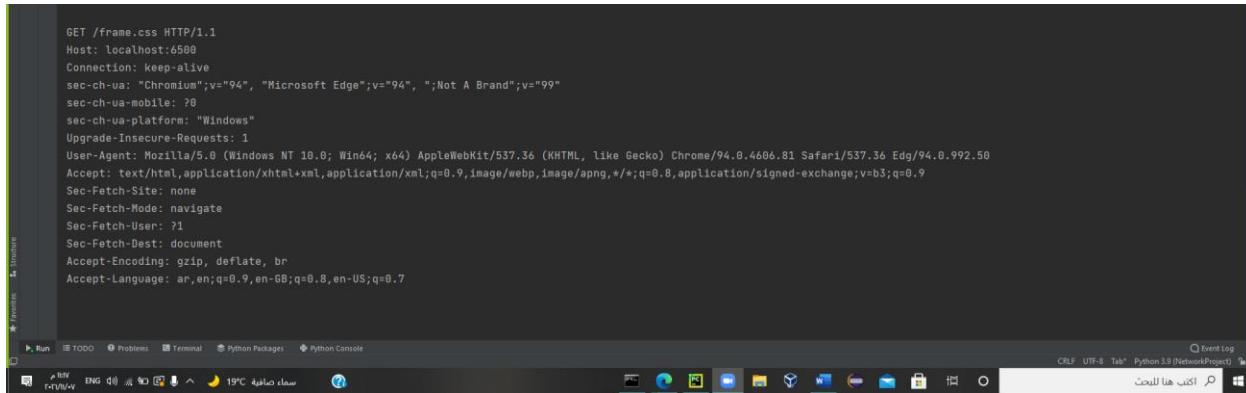
l1 { font-size: 120%; }

div { border: 2px solid #0000FF; border-radius: 20px; height: 400px; width: 300px; display: inline-block; }

.bigBox { padding: 40px 0 0 0; border: 5px solid transparent; width: 100%; text-align: center; height: 400px; display: inline-block; }

img { height: 200px; position: relative; top: 30px; border-radius: 20px; }
```

Figure 14: localhost:6500/Style.css Browser Window



A screenshot of a terminal window titled "Windows PowerShell". The command "curl localhost:6500/frame.css" has been run, and the output is displayed:

```
GET /frame.css HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
```

Figure 15: localhost:6500/Style.css HTTP Request Printed on Command Line

JPG Image:

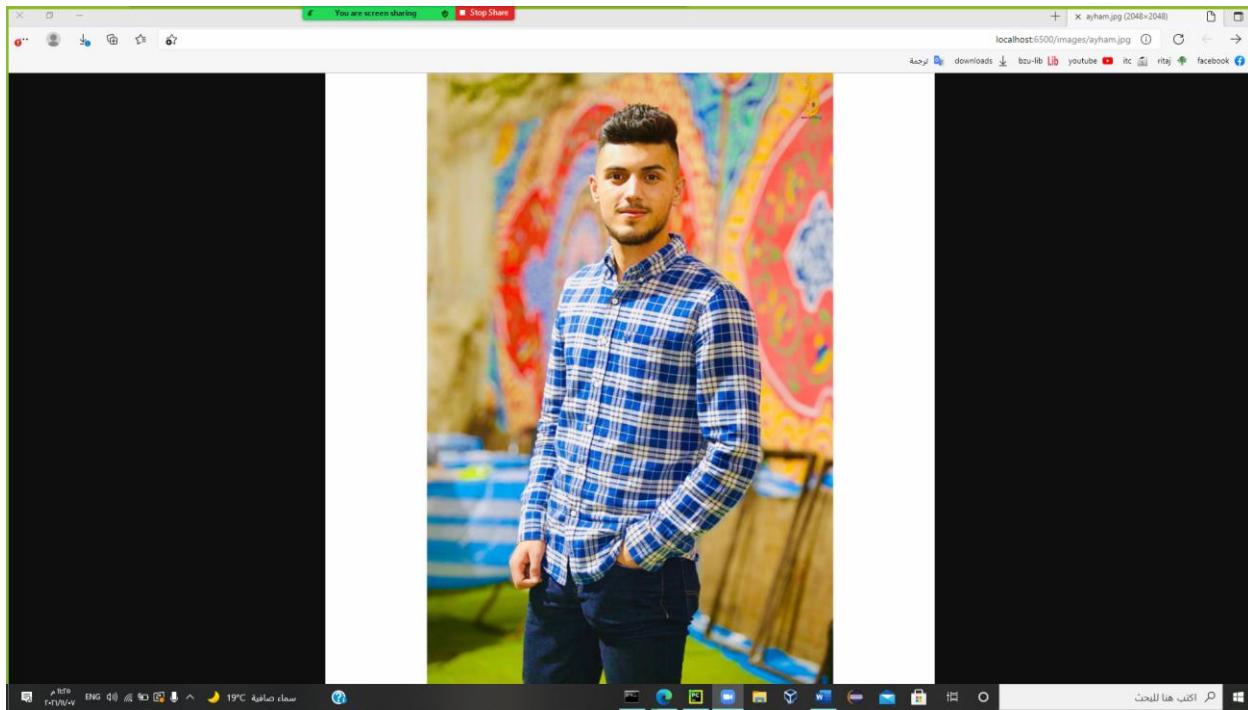


Figure 16: JPG image - Browser Window

```
C:\Users\actc\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/actc/PycharmProjects/NetworkProject/main.py
The server is ready to receive
GET /images/ayham.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
```

Figure 17: JPG image - HTTP Request Printed on Command Line

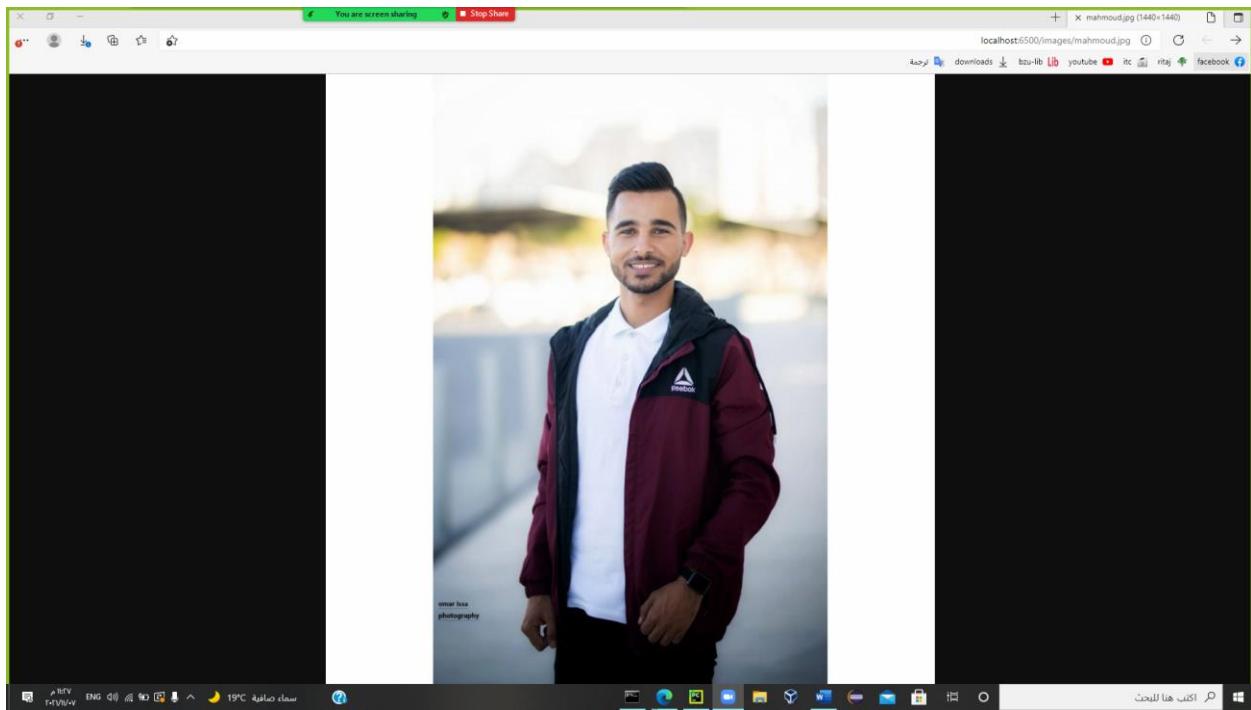


Figure 18: JPG image - Browser Window

```
Run main.py
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
GET /images/mahmoud.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
```

Figure 19: JPG image - HTTP Request Printed on Command Line

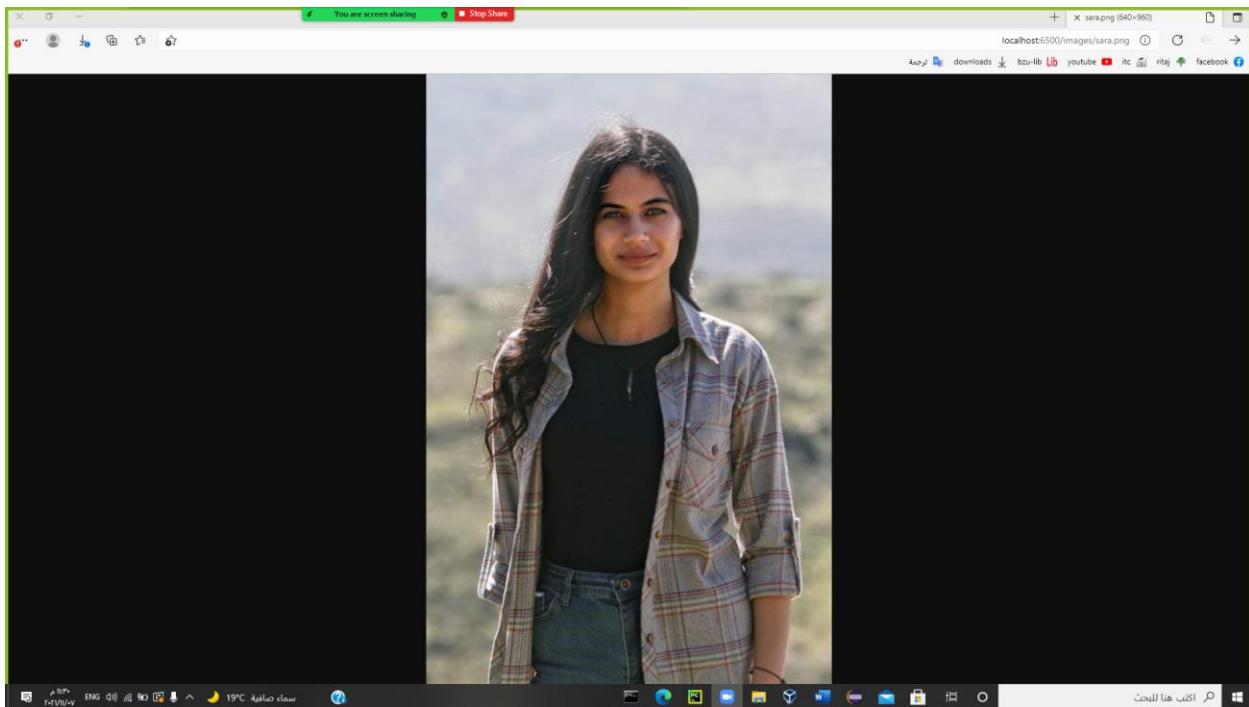


Figure 20: PNG image - Browser Window

```
C:\Users\actc\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/actc/PycharmProjects/NetworkProject/main.py
The server is ready to receive
GET /images/sara.png HTTP/1.1
Host: localhost:6500
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
```

Figure 21: PNG image - HTTP Request Printed on Command Line

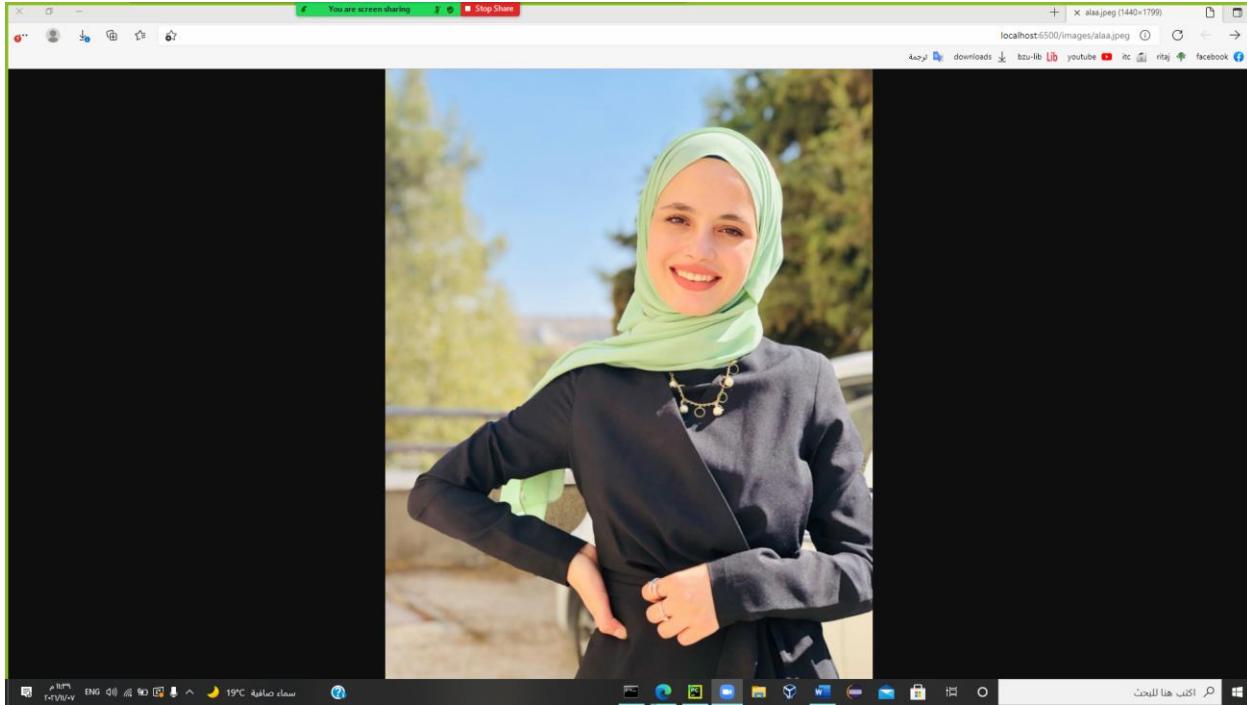


Figure 22: JPEG image - Browser Window

```
C:\Users\actc\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/actc/PycharmProjects/NetworkProject/main.py
The server is ready to receive
GET /images/alaa.jpeg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
```

Figure 23: JPEG image - HTTP Request Printed on Command Line

Sort By Name:

	Price	Name	ID
30	Burger	2	1
50	Sushi	3	2
10	Falafel	4	3
70	Pizza	5	4
15	Shawarma	6	5
20	Misto	7	6
25	Spaghetti	8	7
120	Parique	9	8
150	Marsaf	10	9
40	Musakhan	11	10
		12	11
		13	12
		14	13
		15	14
		16	15
		17	16
		18	17
		19	18
		20	19
		21	20
		22	21
		23	22
		24	23
		25	24
		26	25
		27	26
		28	27
		29	28
		30	29
		31	30
		32	31
		33	32
		34	33
		35	34
		36	35
		37	36
		38	37
		39	38

Figure 24: CSV File to Get Data From

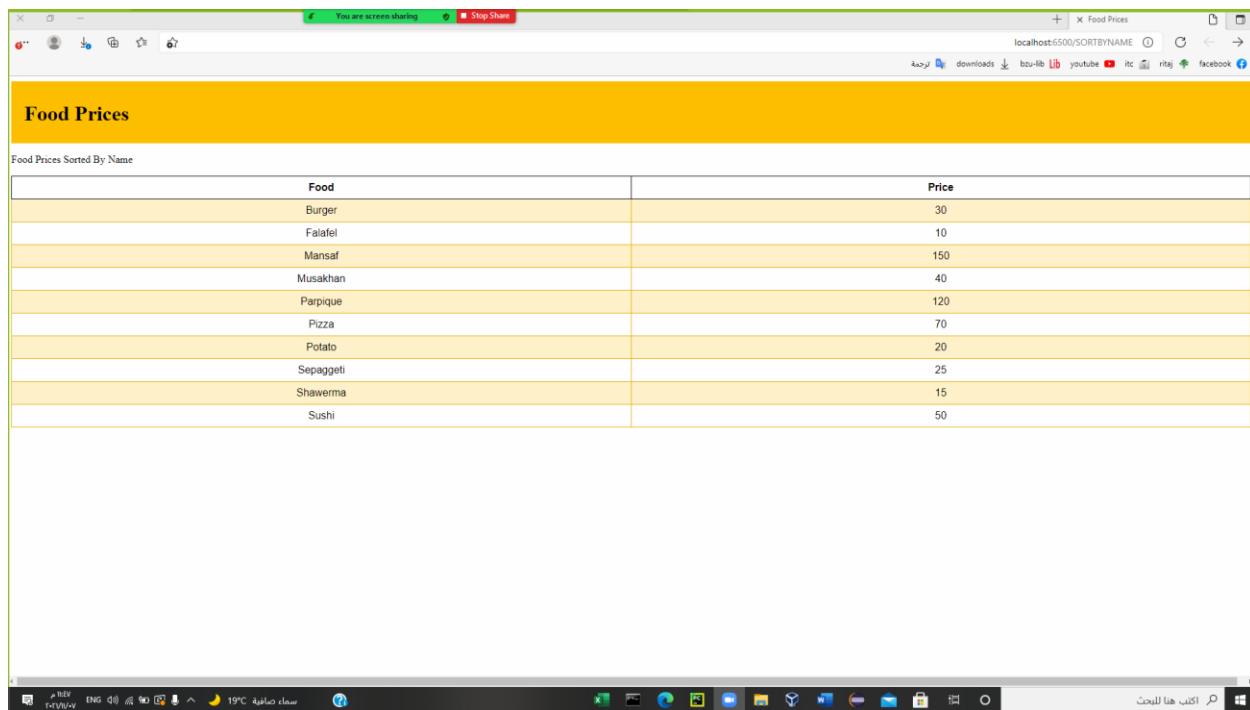


Figure 25: SortByName Browser Window

```
Run: main.py
C:\Users\actc\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/actc/PycharmProjects/NetworkProject/main.py
The server is ready to receive
GET /SORTBYNAME HTTP/1.1
Host: localhost:6500
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
```

Figure 26: SortByName HTTP Request Printed on Command Line

Sort By Price:

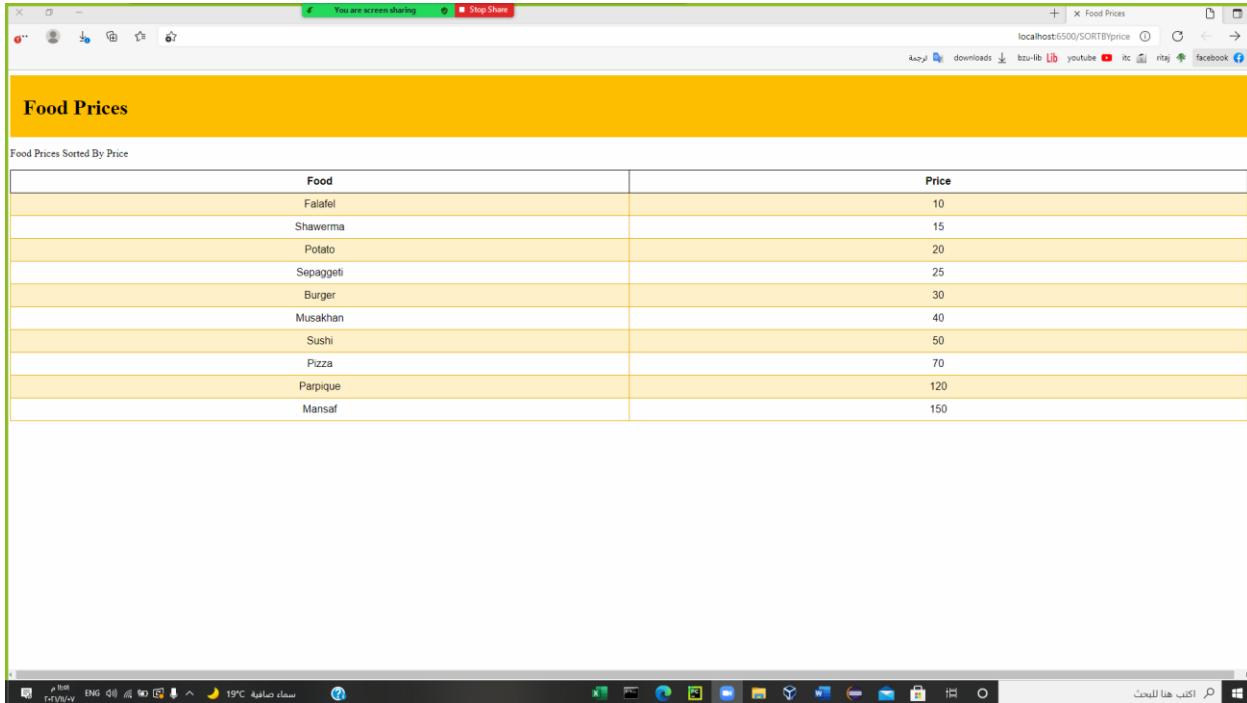


Figure 27: SortByPrice Browser Window

A screenshot of the PyCharm IDE's terminal window. The title bar says "main". The terminal shows the command: "C:\Users\actc\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/actc/PycharmProjects/NetworkProject/main.py". Below this, the server logs state: "The server is ready to receive". A series of HTTP request logs follow:

```
C:\Users\actc\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/actc/PycharmProjects/NetworkProject/main.py
The server is ready to receive
GET /SORTBYprice HTTP/1.1
Host: localhost:6500
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="94", "Microsoft Edge";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.50
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
```

Figure 28: SortByPrice HTTP Request Printed on Command Line

Error 404:

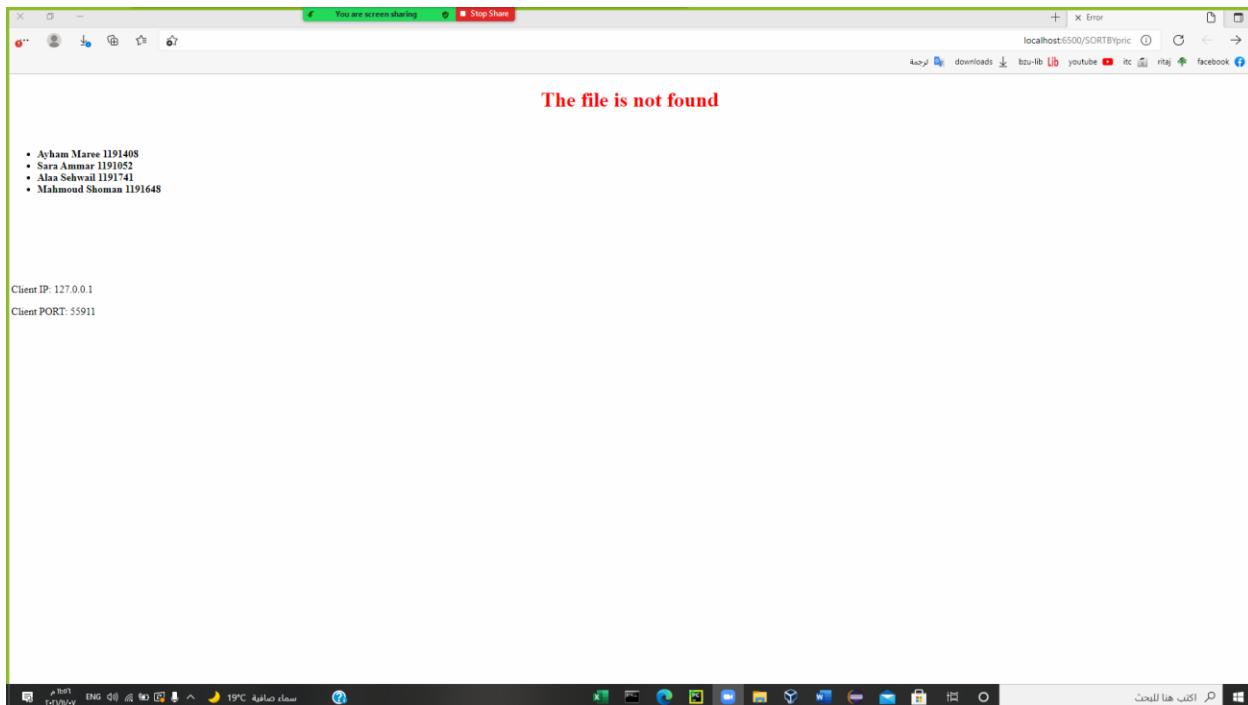


Figure 29: Error 404 Browser Window

A screenshot of a PyCharm IDE showing a terminal window. The terminal output shows an incoming HTTP request from a browser. The request details include the method (GET), URL (/SORTBypric), host (localhost:6500), and various headers such as Connection, Cache-Control, Sec-Ch-Ua, Sec-Ch-Ua-Mobile, Sec-Ch-Ua-Platform, Upgrade-Insecure-Requests, User-Agent, Accept, Sec-Fetch-Site, Sec-Fetch-Mode, Sec-Fetch-User, Sec-Fetch-Dest, Accept-Encoding, and Accept-Language. The response message "The server is ready to receive" is also visible at the top of the terminal.

Figure 30: Not existed page HTTP request printed on command line

Codes: Server Code – Python

```
from socket import *
from food import food

foodList = []
PORT = 6500

# defining the socket, and binding it to the port
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('', PORT))
# socket listening for response
serverSocket.listen(1)

print('The server is ready to receive')

# a function to read the food csv file
def readFile(fileName):
    foodList.clear()
    with open(fileName) as f:
        line = " "
        next(f) # To ignore reading the first line on the csv file (header line)
        while line:
            line = f.readline()

            if line != "":
                lineSplit = line.split(',') # splitting the csv file lines by commas to get the desired data
                Food = food(lineSplit[0], lineSplit[1])
                foodList.append(Food) # putting the file information in a list

# return name function to use as a key in the list sort function
def retName(Food):
    return Food.name

# return price function to use as a key in the list sort function
def retPrice(Food):
    return float(Food.price)

# sort the list by name after storing the file data in it
def sortByName():
    readFile('foodlist.csv')
    foodList.sort(key=retName)

# sort the list by price after storing the file data in it
def sortByPrice():
    readFile('foodlist.csv')
    foodList.sort(key=retPrice)
```

```
while True:
    connection, add = serverSocket.accept()
    sentence = connection.recv(1024).decode('utf-8')

    requesting_file = sentence.split(' ')[1] # from the request sentence, getting the requested file
    requestedFile = requesting_file.lstrip('/') # removing the first / to get the requested file name

    if requestedFile == '' or requestedFile == "index.html": # default request
        requestedFile = 'main.html' # Load main.html file as default

    try:
        sortedBy = ''

        # accepting different file formats
        if requestedFile.endswith(".jpg"):
            requestedType = 'image/jpg'

        elif requestedFile.endswith(".jpeg"):
            requestedType = 'image/jpeg'

        elif requestedFile.endswith(".png"):
            requestedType = 'image/png'

        elif requestedFile.endswith(".css"):
            requestedType = 'text/css'

        elif requestedFile.upper() == "SORTBYNAME":
            sortByName()
            sortedBy = 'Name'
            requestedType = 'text/html'

        elif requestedFile.upper() == "SORTBYPRICE":
            sortByPrice()
            sortedBy = 'Price'
            requestedType = 'text/html'

        else:
            requestedType = 'text/html'

        if requestedFile.upper() != "SORTBYNAME" and requestedFile.upper() != "SORTBYPRICE":
            file = open(requestedFile, 'rb') # opening the requested file
            response = file.read() # reading the file
            file.close() # closing the file
    except:
        connection.sendall("HTTP/1.1 404 Not Found\r\nContent-Type: text/html\r\n\r\n")

```

```

else:
    response = ('<!DOCTYPE html><html><head><title>Food Prices</title><style type="text/css">.header {' +
        'width:100%;height: 85px;background-color: #ffbf00;padding: 10px 0 0 20px;border: ' +
        '0px;border-radius:0px;}table {font-family: arial, sans-serif;border-collapse: ' +
        'collapse;width: 100%;}td, th {border: 1px solid #e6ac00;text-align: center;padding: ' +
        '8px; width: 50%; font-weight: normal;}tr:nth-child(even) {background-color: ' +
        '#fff2cb;}</style></head><body><div class="header"><h1 style="color: black;">Food ' +
        'Prices</h1></div><p>Food Prices Sorted By ' + sortedBy + '</p><table><tr><th style="border: ' +
        '1px solid black; font-weight: bold;">Food</th><th style="border: 1px solid black; ' +
        'font-weight: bold;">Price</th></tr>').encode()
    for item in foodList:
        response += ('<tr><th>' + str(item.name) + '</th><th>' + str(item.price) + '</th></tr>').encode()
    response += '</table></body></html>'.encode()

header = 'HTTP/1.1 200 OK\r\n' # the first part of the header to send.
header += 'Content-Type: ' + str(requestedType) + '\r\n\r\n'

except Exception as e: # Exception if the request the user has entered doesn't exist
    header = 'HTTP/1.1 404 Not Found\r\n'
    header += 'Content-Type: text/html\r\n\r\n'
    response = '<!DOCTYPE html><html><head><title>Error</title><style type="text/css">h1 {text-align: center;}li ' +
        '{font-weight: bold;}</style></head><body><h1 style="color:red">The file is not ' +
        'found</h1><br><ul><li> Ayham Maree 1191408</li><li> Sara Ammar 1191052</li><li> Alaa ' +
        'Sehwail 1191741</li><li> Mahmoud Shoman 1191648</li></ul><div style="position: relative; top: 120px;"><p>Client IP: ' + \
        str(add[0]) + '</p><p>Client PORT: ' + str(add[1]) + '</p></div></body></html>'
    response = response.encode()

final_response = header.encode() + response # encoding the header and adding the response to the request
connection.send(final_response) # sending the final response with all parts of header
connection.close()
print(sentence) # Print the HTTP request on the terminal window

```

```

class food:
    name = ''
    price = 0

    def __init__(self, name, price):
        self.name = name
        self.price = price

    def __repr__(self):
        return self.name + ',' + self.price

```

Main Page HTML Code:

```
<!DOCTYPE html>
<html>
<head>
    <title>ENCS3320-Simple Webserver</title>
    <link rel="stylesheet" href="frame.css">

</head>

<body>
    <div class="header">
        <h1>Welcome to our course <span style="...">Computer Networks</span></h1>
    </div>

    <ul>
        <li> Ayham Maree 1191408</li>
        <li> Sara Ammar 1191052</li>
        <li> Alaa Sehwail 1191741</li>
        <li> Mahmoud Shoman 1191648</li>
    </ul>

    <div class="bigBox">
        <div>
            <h2>Ayham Maree</h2>
            <p>Ayham is smart and love help people in diffrent ways </p>
            
        </div>

        <div>
            <h2>Sara Ammar</h2>
            <p>Sara is clever, has a nice smile , and is one of the top students in his major as well.</p>
            
        </div>

        <div>
            <h2>Mahmoud Shoman</h2>
            <p>Mahmoud is talkative , have a high popularity in his college </p>
            
        </div>

        <div>
            <h2>Alaa Sehwail</h2>
            <p>Alaa is creative and honest. She did many great projects in his major.</p>
            
        </div>
    </div>

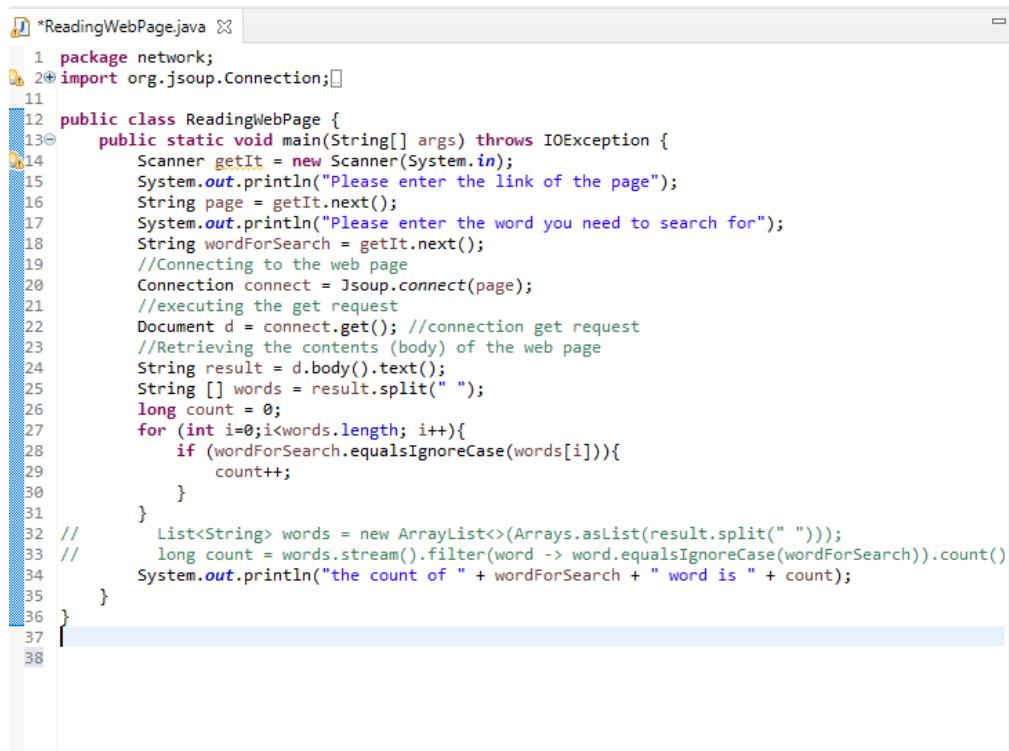
    <a href="testFile.html" target="_blank">Local HTML file</a>
    <p></p>
    <a href="https://www.w3schools.com/tags/tag_div.ASP" target="_blank">Online HTML file</a>
</body>
</html>
```

Cascading Style Sheet Code:

```
.header {  
    width: 100%;  
    height: 85px;  
    background-color: #ffff00;  
    padding: 10px 0 0 20px;  
    border: 0px;  
    border-radius: 0px;  
}  
  
h1{  
    color: black;  
}  
  
li {  
    font-size: 120%;  
}  
  
div {  
    border: 2px solid #0000FF;  
    border-radius: 20px;  
    height: 400px;  
    width: 300px;  
    display: inline-block;  
}  
  
.bigBox {  
    padding: 40px 0 0 0;  
    border: 5px solid transparent;  
    width: 100%;  
    text-align: center;  
    height: 400px;  
    display: inline-block;  
}  
  
img {  
    height: 200px;  
    position: relative;  
    top: 30px;  
    border-radius: 20px;  
}
```

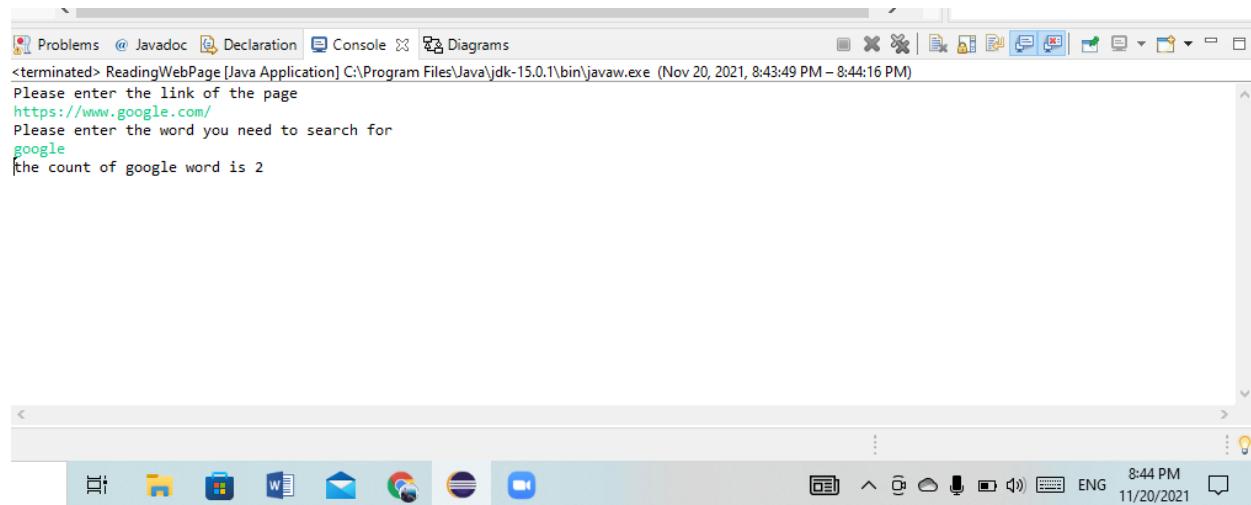
Part 4:

Code:



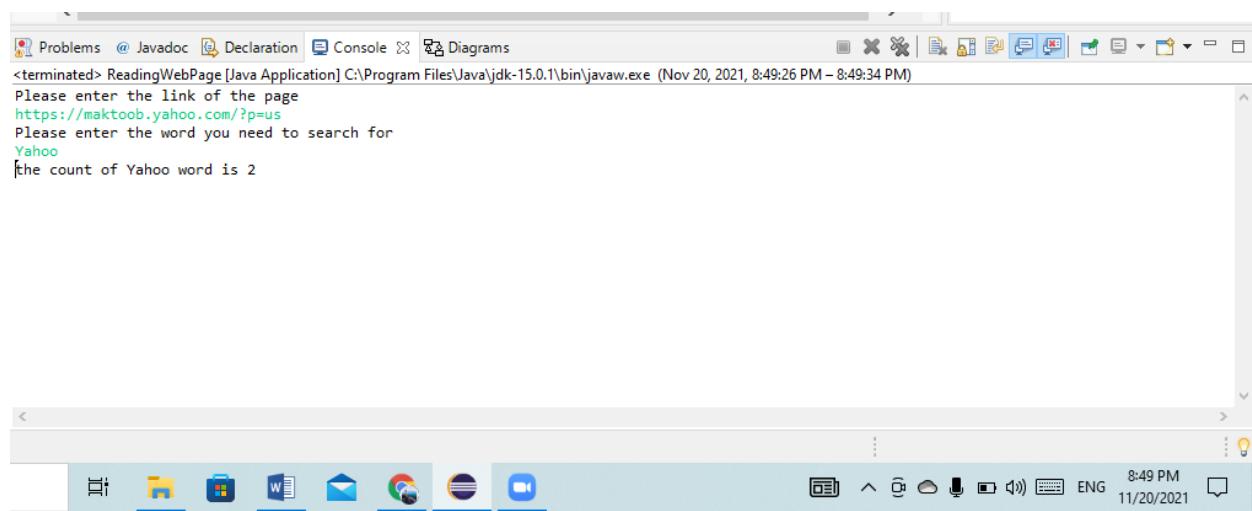
```
1 package network;
2 import org.jsoup.Connection;
3
4 public class ReadingWebPage {
5     public static void main(String[] args) throws IOException {
6         Scanner getIt = new Scanner(System.in);
7         System.out.println("Please enter the link of the page");
8         String page = getIt.nextLine();
9         System.out.println("Please enter the word you need to search for");
10        String wordForSearch = getIt.nextLine();
11        //Connecting to the web page
12        Connection connect = Jsoup.connect(page);
13        //executing the get request
14        Document d = connect.get(); //connection get request
15        //Retrieving the contents (body) of the web page
16        String result = d.body().text();
17        String [] words = result.split(" ");
18        long count = 0;
19        for (int i=0;i<words.length; i++){
20            if (wordForSearch.equalsIgnoreCase(words[i])){
21                count++;
22            }
23        }
24        //List<String> words = new ArrayList<>(Arrays.asList(result.split(" ")));
25        //long count = words.stream().filter(word -> word.equalsIgnoreCase(wordForSearch)).count();
26        System.out.println("the count of " + wordForSearch + " word is " + count);
27    }
28 }
```

Figure 30: Code using Jsoup programming



```
<terminated> ReadingWebPage [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Nov 20, 2021, 8:43:49 PM – 8:44:16 PM)
Please enter the link of the page
https://www.google.com/
Please enter the word you need to search for
google
the count of google word is 2
```

Figure 31: Result of count word(google) in website (www.google.com)



The screenshot shows a Java application window with a title bar containing tabs for 'Problems', 'Javadoc', 'Declaration', 'Console' (which is selected), and 'Diagrams'. The console output window displays the following text:

```
<terminated> ReadingWebPage [Java Application] C:\Program Files\Java\jdk-15.0.1\bin\javaw.exe (Nov 20, 2021, 8:49:26 PM – 8:49:34 PM)
Please enter the link of the page
https://maktoob.yahoo.com/?p=us
Please enter the word you need to search for
Yahoo
the count of Yahoo word is 2
```

Below the window is a taskbar with various icons for Microsoft Office applications (Word, Excel, PowerPoint, etc.) and system icons. The system tray shows the date and time as '11/20/2021 8:49 PM'.

Figure 32: Result of count word(yahoo) in website (www.yahoo.com)