



BIRZEIT UNIVERSITY

Faculty of Engineering & Technology – Electrical & Computer
Engineering Department

Second Semester 2021 – 2022

Computer Networks – ENCS3320.

Project #1

Name: Omar Tawafshah

Name: Ahmaide Al-Awadah

ID: 1191768

ID: 1190823

Section: 2

Instructor: Dr. Mohammed Helal

Date: 4th – 8th May 2022

❖Table of Content

1.	Part I	IV
1.1.	The Ping Command	IV
1.2.	Traceroute Command	VI
1.3.	Name Server Lookup Command	VII
2.	Part II	VIII
2.1.	English Web Page.....	VIII
2.2.	Arabic Web Page.....	XV
2.3.	Requesting an HTML file	XVIII
2.4.	Requesting a CSS File	XIX
2.5.	Requesting a JPG image	XX
2.6.	Requesting a PNG Image	XXI
2.7.	Status Code 307 Temporary Redirect.....	XXII
2.8.	Wrong Request.....	XXIV
3.	Appendix.....	XXV
3.1.	Server code (server.py)	XXV
3.2.	English webpage in HTML (main_en.html).....	XXVII
3.3.	Arabic webpage in HTML (main_ar.html)	XXIX
3.4.	Pages Design CSS File (main.css).....	XXX
3.5.	Button link HTML File (Link.html)	XXXII

❖Table of Figures

Figure 1: Pinging a device on the same network	IV
Figure 2: Pinging youtube.com	V
Figure 3: Tracerouting youtube.com	VI
Figure 4: nslookup youtube.com	VII
Figure 5: English web page header	VIII
Figure 6: Students info in the English web page	IX
Figure 7: Button link HTML page.....	X
Figure 8: W3school page.....	X
Figure 9: Command line for all the English webpage matters.....	XII
Figure 10: Server access from a mobile that is connected on the same local network	XIII
Figure 11: Command line for acceing the server using a mobile on the same local internet	XIV
Figure 12: Arabic webpage.....	XV
Figure 13: Command line for accessing the arabic webpage	XVII
Figure 14: Requesting HTML file command line	XVIII
Figure 15: CSS file in browser.....	XIX
Figure 16: Requesting CSS file command line.....	XIX
Figure 17: JPG image in browser.....	XX
Figure 18: Requesting JPG image command line	XX
Figure 19: PNG image in browser	XXI
Figure 20: Requesting PNG image command line.....	XXI
Figure 21: 307 Redirects command line	XXIII
Figure 22: Redirect to google.com	XXIII
Figure 23: Redirect to cnn.com	XXIII
Figure 24: Redirect to birzeit.edu	XXIII
Figure 25: Error file in browser	XXIV
Figure 26: Wrong request command line	XXIV

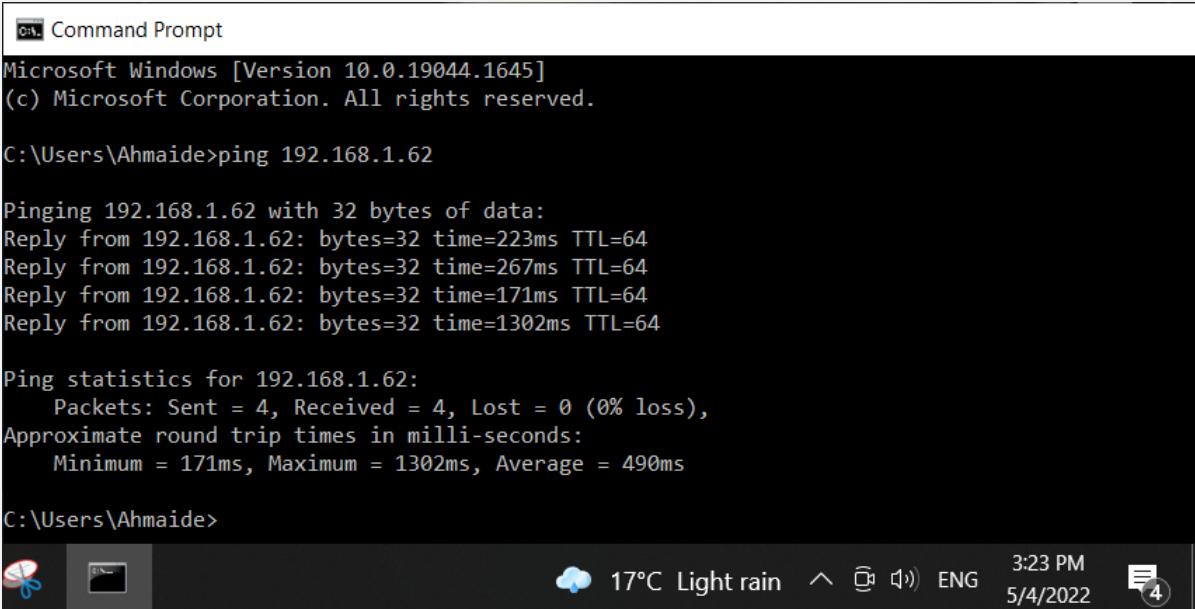
1. Part I

1.1. The Ping Command

The ping is a command-line utility, that can send a request to a specific device and waits for a response, in order to information such as the number of returned responses and the needed time for them to be returned from the receiver computer, in this project the ping command is used to ping a device from the same network and to ping the YouTube site.

1.1.1. Ping a device from the same Network

It can be seen from the figure below that a response was received from 192.168.1.62 where 4 packets were sent and they all have the same TTL (time to live), all packets are received with different delays and the average is 490ms. And this checks if the device is connected over the network.



```
Command Prompt
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ahmaide>ping 192.168.1.62

Pinging 192.168.1.62 with 32 bytes of data:
Reply from 192.168.1.62: bytes=32 time=223ms TTL=64
Reply from 192.168.1.62: bytes=32 time=267ms TTL=64
Reply from 192.168.1.62: bytes=32 time=171ms TTL=64
Reply from 192.168.1.62: bytes=32 time=1302ms TTL=64

Ping statistics for 192.168.1.62:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 171ms, Maximum = 1302ms, Average = 490ms

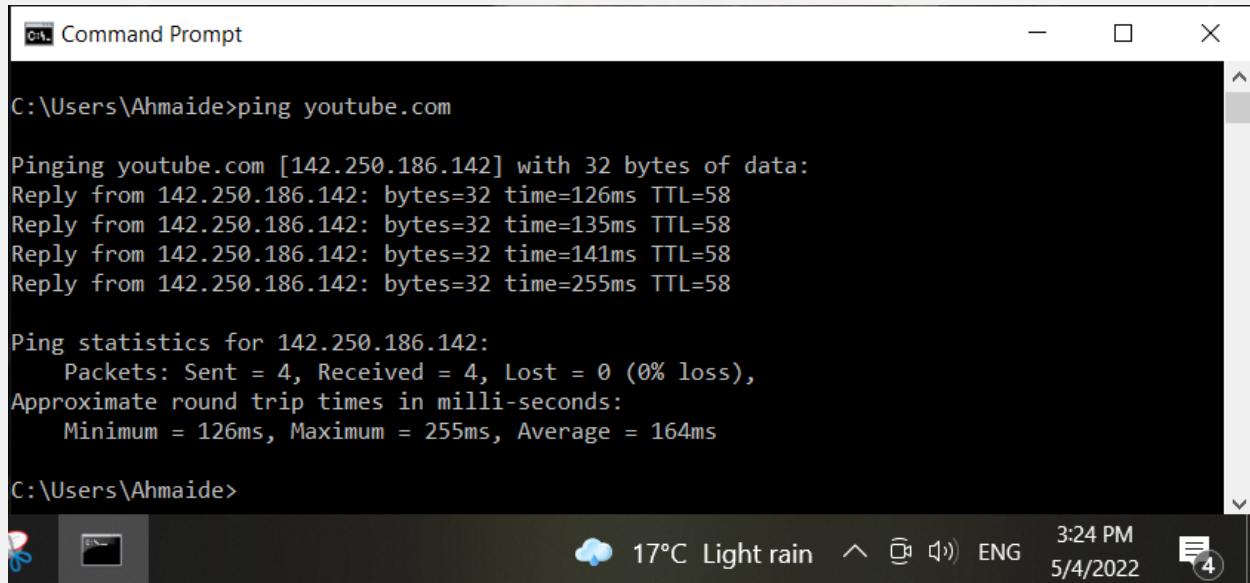
C:\Users\Ahmaide>
```

Figure 1: Pinging a device on the same network

1.1.2. Ping YouTube site

As shown in the figure below there is a response from youtube.com which also verifies that the device is connected to the network and the device access this location.

So, 4 packets have been sent to youtube.com and they all were successfully returned as there is no loss in the packets and that the connection is excellent.



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The user has run the command "ping youtube.com". The output shows four successful ping requests to the IP address 142.250.186.142, with round-trip times ranging from 126ms to 255ms and a TTL of 58. Below this, ping statistics are displayed, showing 4 packets sent, 4 received, 0 lost, and 0% loss. Approximate round-trip times are given as minimum 126ms, maximum 255ms, and average 164ms. The Command Prompt window is set against a dark background with light-colored text. At the bottom of the screen, the taskbar is visible, showing icons for File Explorer, Task View, and Start. The system tray displays the date (5/4/2022), time (3:24 PM), battery level (4), and weather information (17°C, Light rain).

```
C:\Users\Ahmaide>ping youtube.com

Pinging youtube.com [142.250.186.142] with 32 bytes of data:
Reply from 142.250.186.142: bytes=32 time=126ms TTL=58
Reply from 142.250.186.142: bytes=32 time=135ms TTL=58
Reply from 142.250.186.142: bytes=32 time=141ms TTL=58
Reply from 142.250.186.142: bytes=32 time=255ms TTL=58

Ping statistics for 142.250.186.142:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 126ms, Maximum = 255ms, Average = 164ms

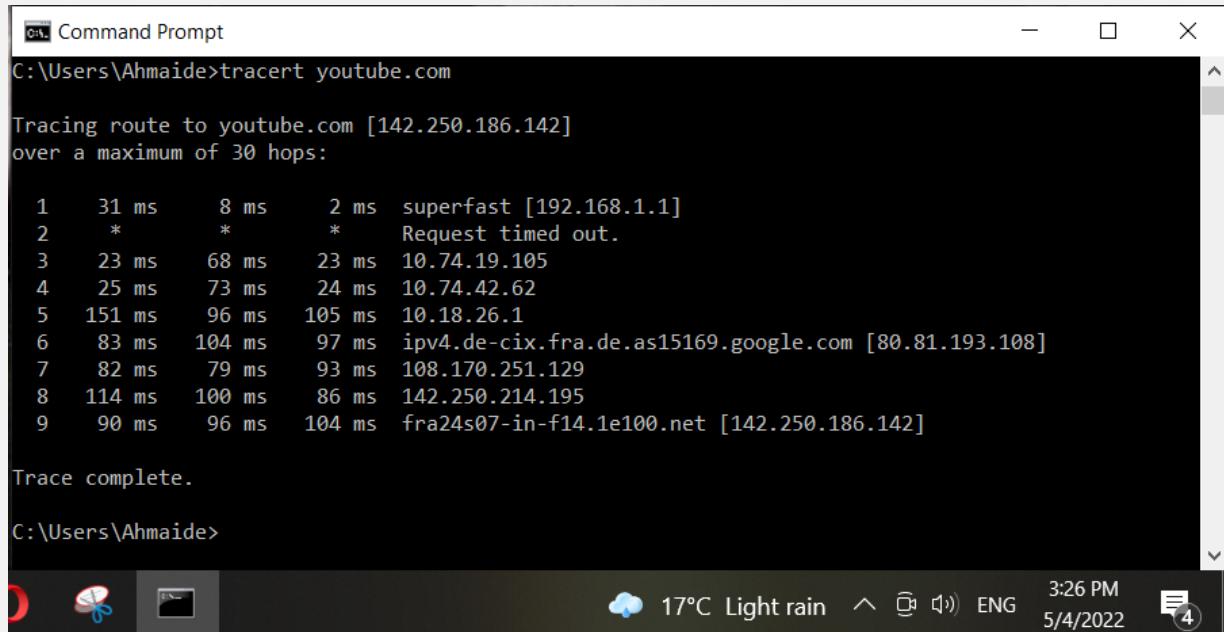
C:\Users\Ahmaide>
```

Figure 2: Pinging youtube.com

1.2. Traceroute Command

The traceroute command is a command that gives the path that the packet takes to get to the destination IP address or domain name, with information such as the time from a hop to another.

As shown in the figure below the devices that the packet took in order to get from the router at 192.168.1.1 to youtube.com at 142.250.186.142 where there is one request that timed out and 6 devices the packet took.



```
Command Prompt
C:\Users\Ahmaide>tracert youtube.com

Tracing route to youtube.com [142.250.186.142]
over a maximum of 30 hops:

 1  31 ms    8 ms    2 ms  superfast [192.168.1.1]
 2  *          *          * Request timed out.
 3  23 ms    68 ms   23 ms  10.74.19.105
 4  25 ms    73 ms   24 ms  10.74.42.62
 5  151 ms   96 ms  105 ms  10.18.26.1
 6  83 ms   104 ms   97 ms  ipv4.de-cix.fra.de.as15169.google.com [80.81.193.108]
 7  82 ms   79 ms   93 ms  108.170.251.129
 8  114 ms   100 ms   86 ms  142.250.214.195
 9  90 ms   96 ms  104 ms  fra24s07-in-f14.1e100.net [142.250.186.142]

Trace complete.

C:\Users\Ahmaide>
```

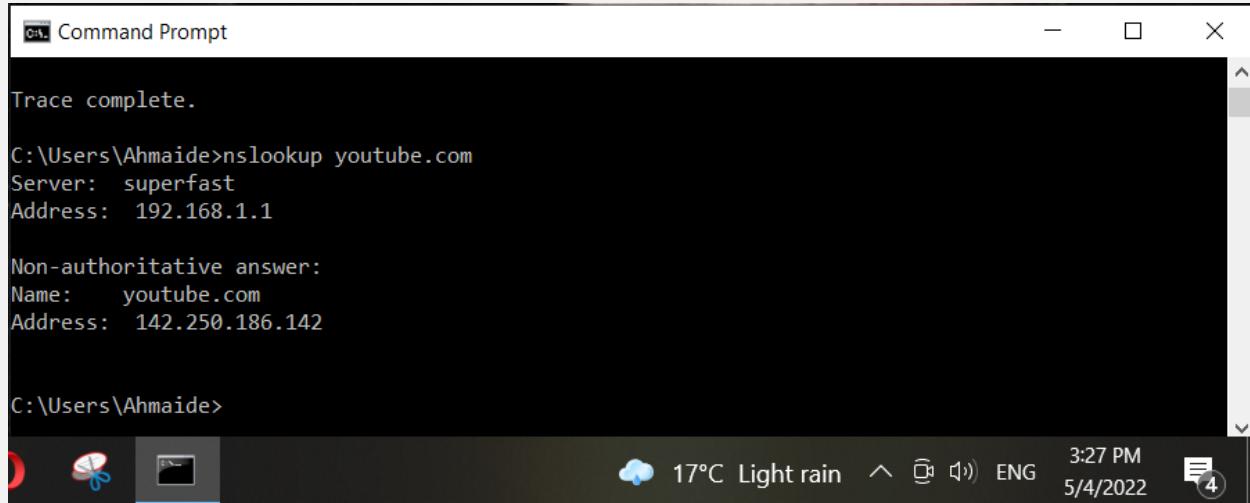
The screenshot shows a Windows Command Prompt window titled "Command Prompt". The user has run the command "tracert youtube.com". The output shows the traceroute path from the user's machine to youtube.com, which consists of 9 routers. The first two routers are marked as "Request timed out". The routers are listed with their IP addresses and the time taken for each hop. The traceroute ends with the destination "fra24s07-in-f14.1e100.net [142.250.186.142]". Below the command prompt, the taskbar is visible with icons for File Explorer, Task View, and Start. The system tray shows the date and time (3:26 PM, 5/4/2022), battery level (4), signal strength, and weather information (17°C, Light rain).

Figure 3: Tracerouting youtube.com

1.3. Name Server Lookup Command

The name server lookup command (Nslookup) is a command that is used to query the Domain name System (DNS) in order to get the mapping between the DNS and the IP address which can help in identifying problems.

As shown in the figure below the DNS is asked for the IP address for the corresponding domain name, as the non-authoritative answer returns the address of 142.250.186.142.



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The output of the command "nslookup youtube.com" is displayed. It starts with "Trace complete.", followed by the command line "C:\Users\Ahmaide>nslookup youtube.com". The response shows the server is "superfast" at "192.168.1.1". A "Non-authoritative answer:" section follows, showing the name "youtube.com" and the address "142.250.186.142". The prompt "C:\Users\Ahmaide>" is visible at the bottom. The taskbar at the bottom of the screen shows icons for File Explorer, Task View, and Start, along with system status icons for weather (17°C Light rain), battery, signal, and network, and the date and time (3:27 PM, 5/4/2022).

```
Trace complete.

C:\Users\Ahmaide>nslookup youtube.com
Server:  superfast
Address: 192.168.1.1

Non-authoritative answer:
Name: youtube.com
Address: 142.250.186.142

C:\Users\Ahmaide>
```

Figure 4: nslookup youtube.com

2. Part II

In this part a server that has its own web pages was created using python programming language to make the server, and web languages CSS & HTML to design the web page.

The used port number is: 9000, the program creates the server socket the binds to it the port number.

The program server should wait until a request comes up, it takes the needed information and the type of request (HTML file, CSS file, image, etc.) and gives the receiver the needed page.

The program is divided into parts, and they are:

2.1. English Web Page

When the user enters the localhost, or localhost IP address followed by (:) and the port number (9000), with a request type of (/) or (/en), they'll should get the English web page that was implemented using HTML and designed using CSS, as shown in the figure below.



Figure 5: English web page header

As shown in figure 5, the English web page has the title of "ENCS3320-Simple Webserver" with a welcoming phrase and a cover jpg image and an extra png image in the top left corner.

The screenshot displays two separate student profiles side-by-side. Both profiles have a teal header and a purple footer.

Profile 1 (Left):

- Name:** Omar Tawafshah - 1191768
- Description:** I have done some great projects in the past semesters like:
 - Database project for SuperMarket.
- Skills:** Some of my skills are:
 - The ability to learn concepts and apply them to other problems.
 - Good in design.
- Hobbies:** My hobbies are:
 - FootBall.
 - Tennis.
 - Learning.
- Image:** A circular profile picture of a young man sitting outdoors.

Profile 2 (Right):

- Name:** Ahmaide Al-Awawdeh - 1190823
- Description:** I have done some great projects in the past semesters like:
 - A VHDL 8-bit structural comparator
- Skills:** Some of my skills are:
 - Using matlab and shell scripting
 - Design verification
- Hobbies:** My hobbies are:
 - Soccer
 - Volunteering
 - Hiking
- Image:** A circular profile picture of a young man sitting outdoors.

The browser interface at the top shows the title "ENC53320-Simple Webserver" and the address "localhost:9000". The footer of each profile includes a "Type here to search" bar, a taskbar with various icons, and a system tray showing weather (13°C Clear), time (123 AM 5/8/2022), and battery status.

Figure 6: Students info in the English web page

As shown in figure 6 the webpage also includes some information about the server creators (us), with an HTML link button, and another link (W3school), if the HTML link button is opened the page that is shown in the figure below will show up.

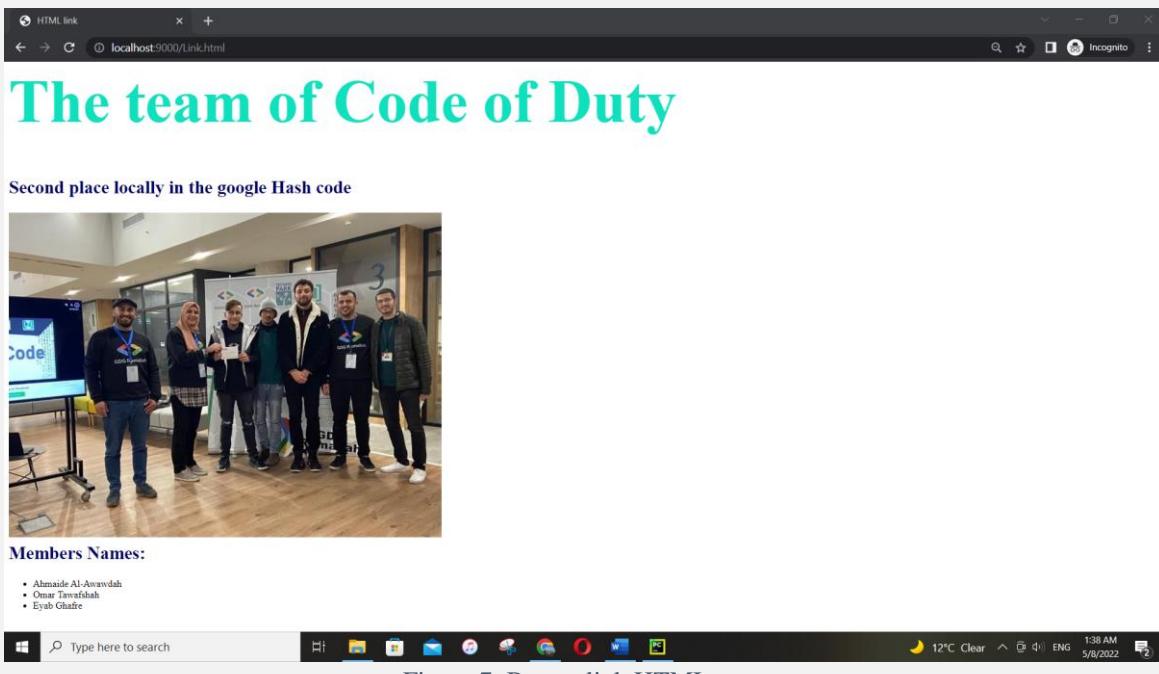


Figure 7: Button link HTML page

If the W3school link is clicked on the page in the figure below will show up.

HTML src Attribute

Example

An image is marked up as follows:

```

```

Definition and Usage

The required `src` attribute specifies the URL of the image.

Figure 8: W3school page

After going through these pages respectively the command line will show as shown in the figure below:

```

Run: server
C:\Users\Ahmaide\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/Ahmaide/PycharmProjects/NetworkProject/venv/server.py
The server has been read successfully!
The server Information:
IP Address: 127.0.0.1
Port: 64530
The user's request is:
GET / HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 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=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

IP Address: 127.0.0.1
Port: 64536
The user's request is: main.css
GET /main.css HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Sec-Fetch-User: ?1
Referer: http://localhost:9000/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

IP Address: 127.0.0.1
Port: 64538
The user's request is: logo.png
GET /logo.png HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: image/avif,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
Sec-Fetch-User: ?1
Referer: http://localhost:9000/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

IP Address: 127.0.0.1
Port: 64539
The user's request is: Omar.jpg
GET /Omar.jpg HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: image/avif,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
Sec-Fetch-User: ?1
Referer: http://localhost:9000/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

```

```

Run: server
IP Address: 127.0.0.1
Port: 64540
The user's request is: background.jpg
GET /background.jpg HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: image/avif,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:9000/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

IP Address: 127.0.0.1
Port: 64541
The user's request is: main.css
GET /main.css HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Accept: image/avif,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:9000/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

IP Address: 127.0.0.1
Port: 64541
The user's request is: Link.html
GET /Link.html HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:9000/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

IP Address: 127.0.0.1
Port: 64568
The user's request is: HashCode.jpg
GET /HashCode.jpg HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: image/avif,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:9000/Link.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

IP Address: 127.0.0.1
Port: 64570
|
```

Figure 9: Command line for all the English webpage matters

The webpage in the server was also accessed from a mobile device that was connected on the same router, as the ip address of the localhost was entered followed by the port number as shown in the figure below, with the HTML link.

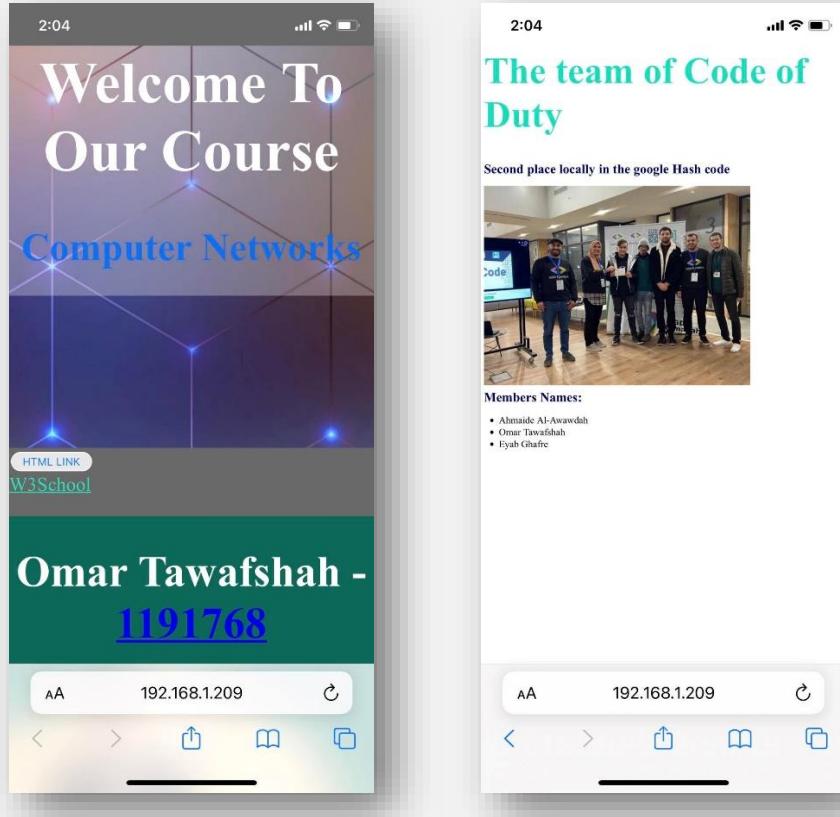


Figure 10: Server access from a mobile that is connected on the same local network

The command line for accessing on the mobile is shown in the figure below.

```
Run: server x
C:\Users\Ahmaide\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/Ahmaide/PycharmProjects/NetworkProject/venv/server.py
The server has been read successfully!
The server information:
IP Address: 192.168.1.62
Port: 56240
The user's request is:
GET / HTTP/1.1
Host: 192.168.1.209:9000
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 15_4_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.4 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

IP Address: 192.168.1.62
Port: 56241
The user's request is: main.css
GET /main.css HTTP/1.1
Host: 192.168.1.209:9000
Connection: keep-alive
Accept: text/css,*/*;q=0.1
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 15_4_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.4 Mobile/15E148 Safari/604.1
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Referer: http://192.168.1.209:9000/
Accept-Encoding: gzip, deflate
```

The figure consists of three vertically stacked screenshots of a Windows desktop environment. Each screenshot shows a terminal window titled 'Run' with the path 'server' and a command-line interface.

- Screenshot 1:** Shows requests for 'logo.png' and 'Omar.jpg'. The user's request is identified as 'Omar.jpg'. The terminal output includes headers such as 'User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 15_4_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.4 Mobile/15E148 Safari/604.1' and 'Accept-Language: en-GB,en-US;q=0.9,en;q=0.8'.
- Screenshot 2:** Shows a request for 'background.jpg'. The user's request is identified as 'Ahnade.jpg'. The terminal output includes headers such as 'User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 15_4_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.4 Mobile/15E148 Safari/604.1' and 'Accept-Language: en-GB,en-US;q=0.9,en;q=0.8'.
- Screenshot 3:** Shows a request for 'Link.html'. The user's request is identified as 'Link.html'. The terminal output includes headers such as 'User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 15_4_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.4 Mobile/15E148 Safari/604.1' and 'Accept-Language: en-GB,en-US;q=0.9,en;q=0.8'.

The desktop taskbar at the bottom shows icons for iTunes, Mail, and other applications. The system tray indicates the date as 5/6/2022 and the time as 2:15 AM, 2:16 AM, and 2:17 AM respectively. The status bar also shows the Python version (3.9) and project name (NetworkProject).

Figure 11: Command line for aceeing the server using a mobile on the same local internet

2.2. Arabic Web Page

This webpage is an arabic version of the previous page, it contains the same: informations, pictures, and links. The HTML file for both pages are the same, yet this one uses arabic labels, and its request ends with (/ar), as shown in the figure below.

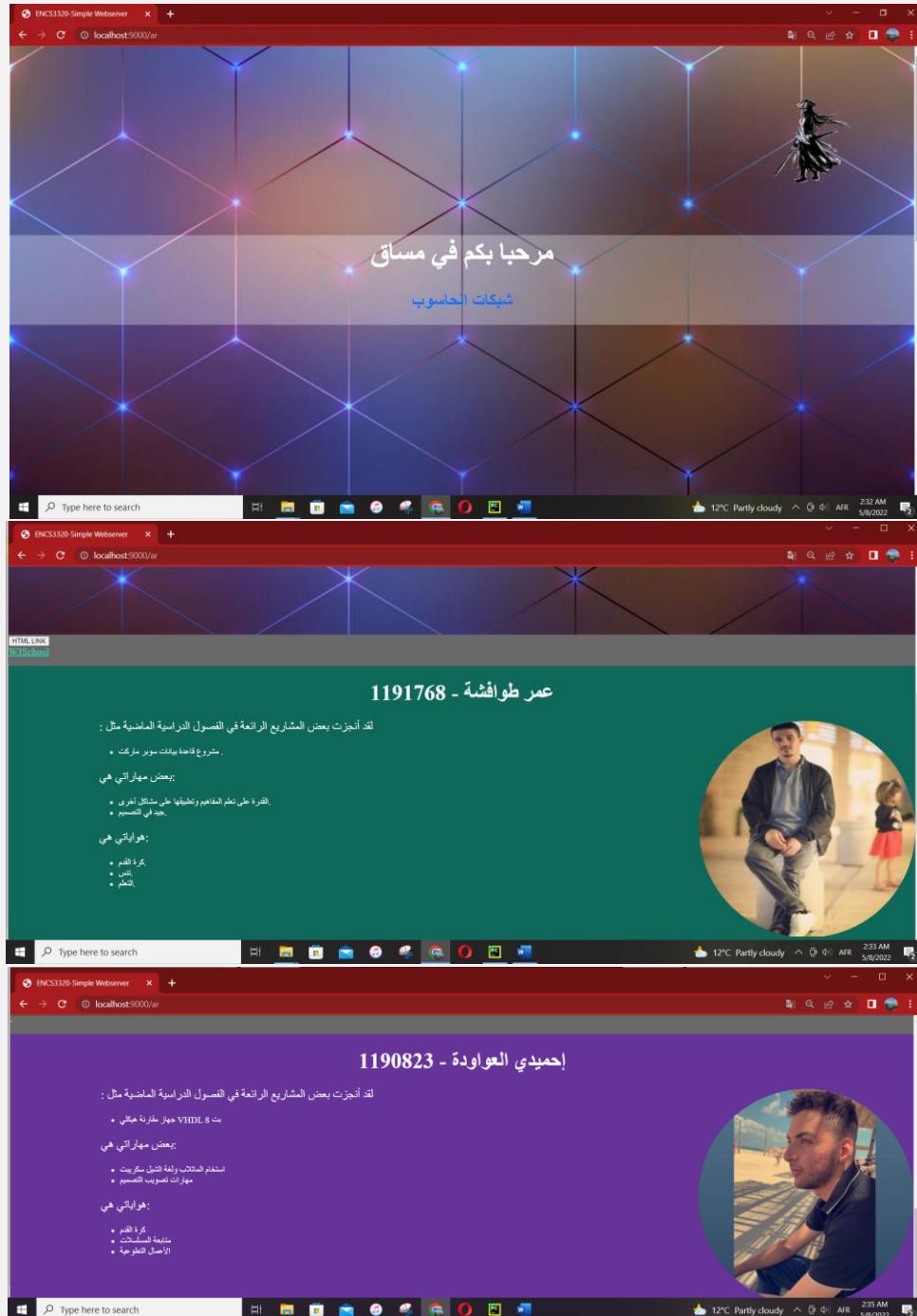


Figure 12: Arabic webpage

The command line for accessing the arabic webpage is displayed in the figure below.

The figure shows three separate terminal windows within the PyCharm IDE, all running the same Python script named 'server.py'. Each window displays a different log entry corresponding to a user request for an Arabic webpage. The logs include details such as IP address, port, user agent, and specific headers related to the Arabic content.

```
C:\Users\Ahmaide\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/Ahmaide/PycharmProjects/NetworkProject/venv/server.py
The server has been read successfully!
The server Information:
IP Address: 127.0.0.1
Port: 5789
The user's request is: ar
GET /ar HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 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=0.9
Purpose: prefetch
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm-4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 5789
The user's request is: main.css
GET /main.css HTTP/1.1

Run TODO Problems Terminal Python Packages Python Console Event Log
155:1 Python 3.9 (NetworkProject)
12°C Partly cloudy 2:47 AM 5/8/2022
```



```
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
Accept: application/signed-exchange;v=b3;q=0.7,*/*;q=0.8
Purpose: prefetch
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:9000/ar
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm-4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 5789
The user's request is: logo.png
GET /logo.png HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: application/signed-exchange;v=b3;q=0.7,*/*;q=0.8
Purpose: prefetch
Sec-Fetch-Site: same-origin
```



```
Run TODO Problems Terminal Python Packages Python Console Event Log
155:1 Python 3.9 (NetworkProject)
12°C Partly cloudy 2:47 AM 5/8/2022
```

The figure consists of three vertically stacked screenshots of a PyCharm terminal window. Each screenshot shows a list of network requests made by a browser to a local host (localhost:9000). The requests are for Arabic files: 'Omar.jpg', 'background.jpg', and 'favicon.ico'. The terminal output includes headers such as 'Sec-Fetch-Mode: no-cors', 'Sec-Fetch-Dest: image', and 'Accept-Language: en-US,en;q=0.9'. The environment is a Windows desktop with a taskbar at the bottom showing icons for various applications like File Explorer, Mail, and a browser.

```

Run: server
IP Address: 127.0.0.1
Port: 57897
The user's request is: Omar.jpg
GET /Omar.jpg HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: application/signed-exchange;v=b3;q=0.7,*/*;q=0.8
Purpose: prefetch
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:9000/ar
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm-403icef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 57898
The user's request is: background.jpg
GET /background.jpg HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: image/avif,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:9000/main.css
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm-403icef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 57908
The user's request is: Ahmaide.jpg
GET /Ahmaide.jpg HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: application/signed-exchange;v=b3;q=0.7,*/*;q=0.8
Purpose: prefetch
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:9000/ar
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm-403icef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 57908
The user's request is: favicon.ico
GET /favicon.ico HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: image/avif,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:9000/ar
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm-403icef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

```

Figure 13: Command line for accessing the arabic webpage

2.3. Requesting an HTML file

If the user asks to open an html file it will be opened, the figure below shows the command line for opening the HTML link as the one in figure 7.

The figure consists of four vertically stacked screenshots of a Windows desktop environment. The top two screenshots show the PyCharm IDE's 'Run' tool window. The first shows the server log output for a request to 'link.html', displaying detailed HTTP headers and a cookie. The second shows the server log output for a request to 'HashCode.jpg', also with detailed headers and a cookie. The bottom two screenshots show the browser output for these requests. The first browser screenshot shows the page source for 'link.html', which contains the text 'The server has been read successfully!'. The second browser screenshot shows the image for 'HashCode.jpg', which is a standard Windows file icon.

```
C:\Users\Ahmaide\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/Ahmaide/PycharmProjects/NetworkProject/venv/server.py
The server has been read successfully!
The server Information:
IP Address: 127.0.0.1
Port: 65465
The user's request is: link.html
GET /link.html HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 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=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm=4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 65467
The user's request is: HashCode.jpg
GET /HashCode.jpg HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-device: "Windows"
sec-ch-ua-platform: same-origin
sec-ch-ua-version: "5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36"
Accept: image/avif,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:9000/link.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm=4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 65468
The user's request is: favicon.ico
GET /favicon.ico HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: image/avif,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:9000/link.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm=4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 65469
The user's request is: HashCode.jpg
GET /HashCode.jpg HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-device: "Windows"
sec-ch-ua-platform: same-origin
sec-ch-ua-version: "5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36"
Accept: image/avif,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:9000/link.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm=4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6
```

Figure 14: Requesting HTML file command line

2.4. Requesting a CSS File

If the user requests to open any CSS file it can be opened, for example if the user requested to open the CSS file (that was used to design the english and arabic webpages) it will be opened as shown in the figure below, and a figure of the command line after it.

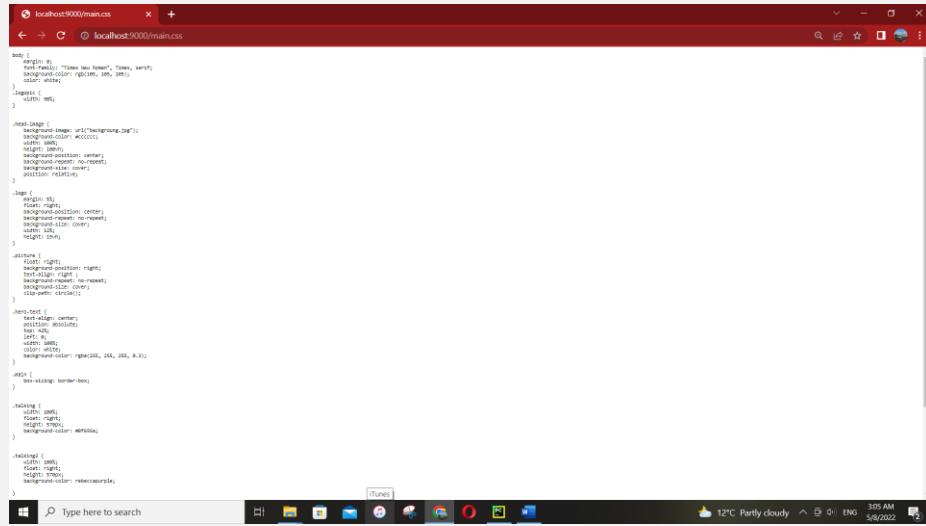


Figure 15: CSS file in browser

Figure 16: Requesting CSS file command line

2.5. Requesting a JPG image

If the user requests an existing JPG image the browser should open it as shown in the figure below, and the command line will print as the figure after it.



Figure 17: JPG image in browser

```
C:\Users\Ahsaide\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/Ahsaide/PycharmProjects/NetworkProject/venv/server.py
The server has been read successfully!
The server Information:
IP Address: 127.0.0.1
Port: 51085
The user's request is: jpg
GET /jpg HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 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=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm=403icef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 51086
The user's request is: .jpg
GET /.jpg HTTP/1.1
Host: localhost:9000
```

```
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 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=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm=403icef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 51088
```

Figure 18: Requesting JPG image command line

2.6. Requesting a PNG Image

If the user requests an existing PNG image the browser should open it as shown in the figure below, and the command line will print as the figure after it.

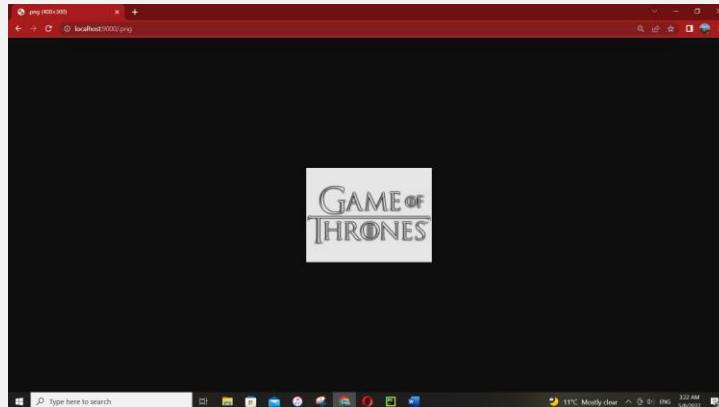


Figure 19: PNG image in browser

```
Run: server
C:\Users\Ahmaide\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/Ahmaide/PycharmProjects/NetworkProject/venv/server.py
The server has been read successfully!
The server Information:
IP Address: 127.0.0.1
Port: 53098
The user's request is: .png
GET /.png HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 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=0.9
Purpose: prefetch
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm-4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 53101
The user's request is: favicon.ico
GET /favicon.ico HTTP/1.1

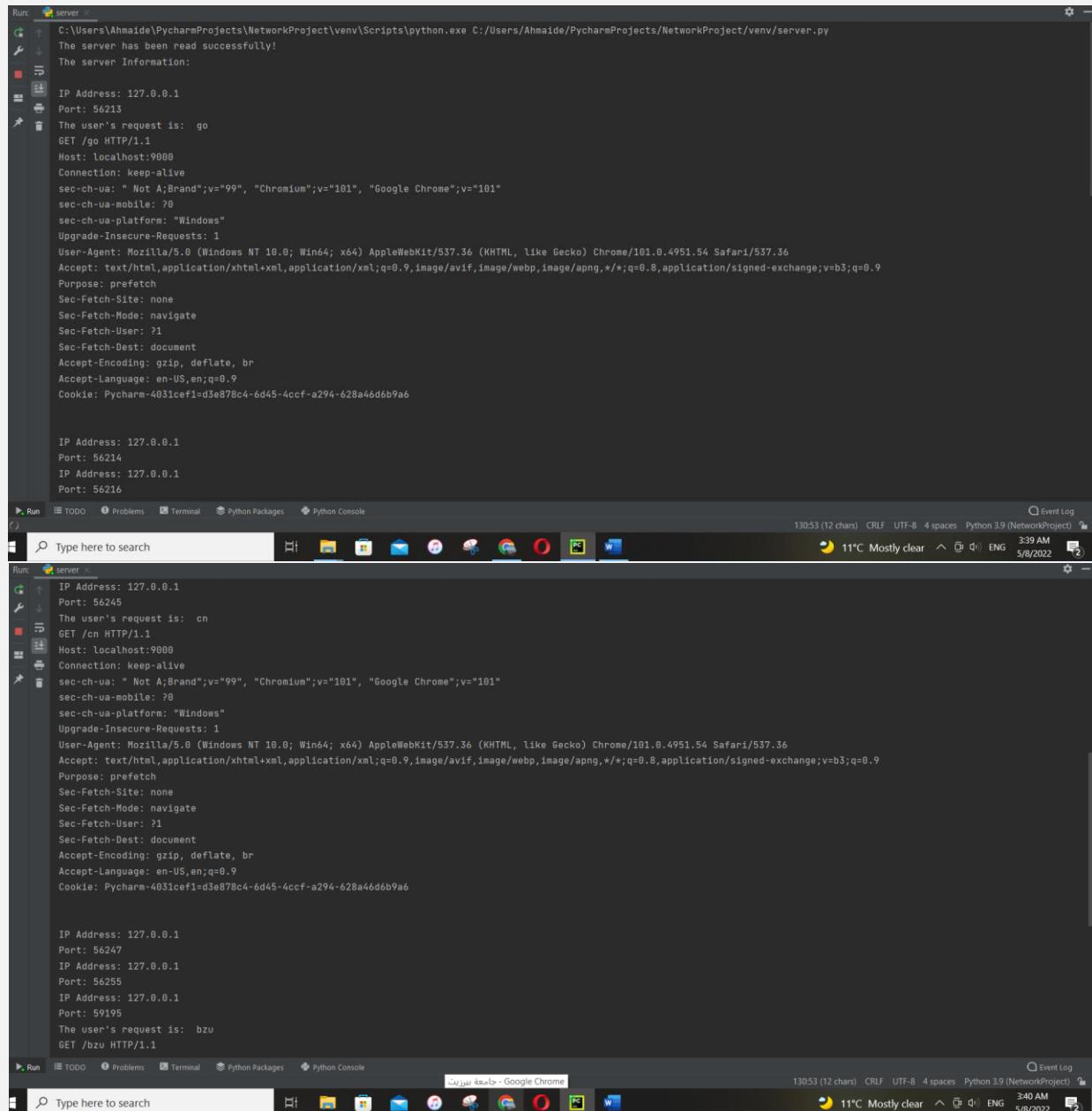
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: image/avif,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:9000/.png
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm-4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6
```

Figure 20: Requesting PNG image command line

2.7. Status Code 307 Temporary Redirect

The 307 status code can redirect to any of these three sites (google.com with the request go, cnn.com with the request cn, birzeit.edu with the request bzu).

The command line for requesting these three websites in order is displayed in the figure below.



```
Run: server
C:\Users\Ahmaide\PycharmProjects\NetworkProject\venv\Scripts\python.exe C:/Users/Ahmaide/PycharmProjects/NetworkProject/venv/server.py
The server has been read successfully!
The server Information:
IP Address: 127.0.0.1
Port: 56213
The user's request is: go
GET /go HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 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=0.9
Purpose: prefetch
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm=4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 56214
IP Address: 127.0.0.1
Port: 56216
Run TODO Problems Terminal Python Packages Python Console Event Log
13053 (12 chars) CRLF UTF-8 4 spaces Python 3.9 (NetworkProject)
11°C Mostly clear 3:39 AM 5/8/2022

Run: server x
IP Address: 127.0.0.1
Port: 56245
The user's request is: cn
GET /cn HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 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=0.9
Purpose: prefetch
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm=4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 56247
IP Address: 127.0.0.1
Port: 56255
IP Address: 127.0.0.1
Port: 59195
The user's request is: bzu
GET /bzu HTTP/1.1
Run TODO Problems Terminal Python Packages Python Console Event Log
13053 (12 chars) CRLF UTF-8 4 spaces Python 3.9 (NetworkProject)
11°C Mostly clear 3:40 AM 5/8/2022
```

```
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
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/101.0.4951.54 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=0.9
Purpose: prefetch
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: Pycharm-4031cef1=d3e878c4-6d45-4ccf-a294-628a46d6b9a6

IP Address: 127.0.0.1
Port: 59205
IP Address: 127.0.0.1
Port: 59251
```

Figure 21: 307 Redirects command line

The figures below shows that these requests redirect to these sites.



Figure 22: Redirect to google.com



Figure 23: Redirect to cnn.com

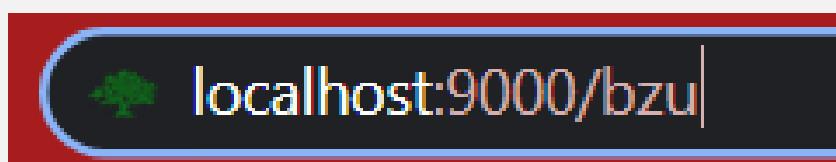


Figure 24: Redirect to birzeit.edu

2.8. Wrong Request

If the request was wrong or wasn't any of the previous ones, the following error HTML file will be opened.



Figure 25: Error file in browser

The following figure shows the command line for the wrong request.

A screenshot of a PyCharm IDE showing a terminal window. The terminal output displays two separate log entries for requests to 'localhost:9000/aaaaa' and 'localhost:9000/favicon.ico'. Both entries show the server reading the request, extracting user-agent information (Mozilla/5.0), and listing various HTTP headers including 'Accept', 'Sec-Fetch-Site', 'Sec-Fetch-Mode', and 'Cookie'. The logs also mention the IP address (127.0.0.1) and port (62924). The PyCharm interface includes toolbars, a navigation bar, and status bars at the bottom indicating the date/time (5/8/2022, 3:58 AM) and system status (11°C, Mostly clear).

Figure 26: Wrong request command line

3. Appendix

3.1. Server code (server.py)

```
from socket import *
from wsgiref.util import request_uri

# The hosting server port (entered in the browser as localhost:9000)
port = 9000

# The server's entirety socket
serverSocket = socket(AF_INET, SOCK_STREAM)

# Adding the port number to the the server socket
serverSocket.bind(("", port))

# HTTP response
serverSocket.listen(1)

print("The server has been read successfully!")
print("The server Information: \n")

# The loop that goes on as the users keeps searching
# As many time as they want on the same port
while True:

    #Receiving a request from another socket
    connectionSocket, address = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode('utf-8')

    # Getting the IP address & the port number and printing them
    IP_address = address[0]
    Connected_port = address[1]
    print("IP Address: " + str(IP_address))
    print("Port: " + str(Connected_port))
    workOn=False

    # To only work if there is information from the connected socket
    if sentence != '':
        sentence_list = sentence.split(' ')
        reqFile = sentence_list[1]
        request = (reqFile).split(" ")[0]
        request = request.lstrip('/')
        workOn=True
    else:
        connectionSocket.close()

    # Here are all the cases for the types of requests
    if workOn:
        req=request
        if(req == ''):
            req = 'en'
        print("The user's request is: ", request)
        try:

            # English Website
            if request == '' or request == 'en':
                connectionSocket.send(f"HTTP/1.1 200 ok \r\n".encode())
                disPage= 'main_en.html'
                disfile = open(disPage, "rb")
                response = disfile.read()
                disfile.close()
                connectionSocket.send(f"\r\n".encode())
                connectionSocket.send(response)
            elif request == 'de':
                connectionSocket.send(f"HTTP/1.1 200 ok \r\n".encode())
                disPage= 'main_de.html'
                disfile = open(disPage, "rb")
                response = disfile.read()
                disfile.close()
                connectionSocket.send(f"\r\n".encode())
                connectionSocket.send(response)
            elif request == 'fr':
                connectionSocket.send(f"HTTP/1.1 200 ok \r\n".encode())
                disPage= 'main_fr.html'
                disfile = open(disPage, "rb")
                response = disfile.read()
                disfile.close()
                connectionSocket.send(f"\r\n".encode())
                connectionSocket.send(response)
            else:
                connectionSocket.send(f"HTTP/1.1 404 not found \r\n".encode())
                disPage= '404.html'
                disfile = open(disPage, "rb")
                response = disfile.read()
                disfile.close()
                connectionSocket.send(f"\r\n".encode())
                connectionSocket.send(response)
        except:
            connectionSocket.send(f"HTTP/1.1 500 internal server error \r\n".encode())
            disPage= '500.html'
            disfile = open(disPage, "rb")
            response = disfile.read()
            disfile.close()
            connectionSocket.send(f"\r\n".encode())
            connectionSocket.send(response)
```

```

connectionSocket.close()

# Arabic Website
elif request == 'ar':
    connectionSocket.send(f"HTTP/1.1 200 ok \r\n".encode())
    disPage = 'main_ar.html'
    disfile = open(disPage, "rb")
    response = disfile.read()
    disfile.close()
    connectionSocket.send(f"\r\n".encode())
    connectionSocket.send(response)
    connectionSocket.close()

# 307 Temporary Redirect to google.com
elif request == 'go':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect \r\n".encode())
    response = (b"location: http://www.google.com")
    connectionSocket.send(response)
    connectionSocket.close()

# 307 Temporary Redirect to cnn.com
elif request == 'cn':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect \r\n".encode())
    response = (b"location: https://edition.cnn.com")
    connectionSocket.send(response)
    connectionSocket.close()

# 307 Temporary Redirect to the arabic Birzeit.edu
elif request == 'bzu':
    connectionSocket.send("HTTP/1.1 307 Temporary Redirect \r\n".encode())
    response = (b"location: https://www.birzeit.edu/ar")
    connectionSocket.send(response)
    connectionSocket.close()

# Open an HTML file
elif request.endswith('.html'):
    connectionSocket.send(f"HTTP/1.1 200 ok \r\n".encode())
    disfile = open(request, "rb")
    response = disfile.read()
    disfile.close()
    connectionSocket.send(f"\r\n".encode())
    connectionSocket.send(response)
    connectionSocket.close()

# Open a CSS file
elif request.endswith('.css'):
    connectionSocket.send(f"HTTP/1.1 200 ok \r\n".encode())
    disfile = open(request, "rb")
    response = disfile.read()
    disfile.close()
    connectionSocket.send(f"\r\n".encode())
    connectionSocket.send(response)
    connectionSocket.close()

# Open a city jpg picture
elif request.endswith(".jpg"):
    connectionSocket.send(f"HTTP/1.1 200 ok \r\n".encode())
    try:
        disfile = open(request, "rb")
    except Exception as exp:
        disfile = open('city.jpg', "rb")
    response = disfile.read()
    connectionSocket.send(f"\r\n".encode())
    connectionSocket.send(response)
    connectionSocket.close()

# Open a png picture of game of thrones logo
elif request.endswith('.png'):
    connectionSocket.send(f"HTTP/1.1 200 ok \r\n".encode())
    try:
        disfile = open(request, "rb")

```

```

        except Exception as exp:
            disfile = open('gameOfThrones.png', "rb")
            response = disfile.read()
            connectionSocket.send(f"\r\n".encode())
            connectionSocket.send(response)
            connectionSocket.close()

        # Any other Case to give the error file
        else :
            connectionSocket.send("HTTP/1.1 404 Not Found\r\n".encode())
            response = (
                '<html><title>Error</title><body><center><h1 style="color:red">Error
404: The file is not found</h1><hr><p style= "font-weight: bold;">Omar Tawafshah - 1191768</p><p
style= "font-weight: bold;">Ahmaide Al-Awadheh - 1190823</p><hr><h2>IP: ' + str(
                IP_address) + ', Port: ' + str(Connected_port) +
                '</h2></center></body></html>').encode()
                connectionSocket.send(f"\r\n".encode())
                connectionSocket.send(response)
                connectionSocket.close()

            # Also give the error file if an actuall error accured
            except Exception as DidNotWork:
                connectionSocket.send("HTTP/1.1 404 Not Found\r\n".encode())
                response = ('<html><title>Error</title><body><center><h1 style="color:red">Error 404:
The file is not found</h1><hr><p style= "font-weight: bold;">Omar Tawafshah - 1191768</p><p
style= "font-weight: bold;">Ahmaide Al-Awadheh - 1190823</p><hr><h2>IP: ' + str(
                    IP_address) + ', Port: ' + str(Connected_port) +
                    '</h2></center></body></html>').encode()

                print(sentence)

```

3.2. English webpage in HTML (main_en.html)

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ENC83320-Simple Webserver</title>
    <link rel="stylesheet" href="main.css">
</head>
<body>
    <div class="head-image">
        <div class="logo">
            
        </div>
        <div class="hero-text">
            <h1 style="font-size:60px; margin-top: 5px;">Welcome To Our Course</h1>
            <h1 style="font-size:40px; color:rgba(0,109,255,0.87); font-weight: bold;">Computer
Networks</h1>
        </div>
    </div>
    <div>
        <a style="font-size:20px; margin-top: 5px; color: #cb0c2a" href="Link.html"><button> HTML
LINK </button></a> <!-- Button link --></a>
    </div>
    <div>
        <a style="font-size:20px; margin-top: 5px; color: #0fe0bd"
href="https://www.w3schools.com/tags/att_img_src.asp"> W3School</a>
    </div>
    <br>
    <div class="main" style="margin-bottom: 15px;">
        <div class="talking">

```

```

<div style="text-align: center;">
    <h1 class="textName"> Omar Tawafshah - 1191768</h1>
</div>
<div class="picture">
    
</div>
<div class="textInfo">
    <p style="font-size: 135%;">I have done some great projects in the past semesters like:</p>
    <ul>
        <li>Database project for SuperMarket.</li>
    </ul>
    <p style="font-size: 135%;">Some of my skills are:</p>
    <ul>
        <li>The ability to learn concepts and apply them to other problems.</li>
        <li>Good in design.</li>
    </ul>
    <p style="font-size: 135%;">My hobbies are:</p>
    <ul>
        <li>FootBall.</li>
        <li>Tennis.</li>
        <li>Learning.</li>
    </ul>
</div>
</div>
<h5>.</h5>
<div class="main" style="margin-top: 5px">
    <div class="talking2">
        <div style="text-align: center;">
            <h1 class="textName"> Ahmaide Al-Awadheh - 1190823</h1>
        </div>
        <div class="picture">
            
        </div>
        <div class="textInfo">
            <p style="font-size: 135%;">I have done some great projects in the past semesters like:</p>
            <ul>
                <li>A VHDL 8-bit structural comparator</li>
            </ul>
            <p style="font-size: 135%;">Some of my skills are:</p>
            <ul>
                <li>Using matlab and shell scripting</li>
                <li>Design verification</li>
            </ul>
            <p style="font-size: 135%;">My hobbies are:</p>
            <ul>
                <li>Soccer</li>
                <li>Volunteering</li>
                <li>Hiking</li>
            </ul>
        </div>
    </div>
</div>

</div>
</body>
</html>

```

3.3. Arabic webpage in HTML (main_ar.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ENCS3320-Simple Webserver</title>
    <link rel="stylesheet" href="main.css">
</head>
<body>
    <div class="head-image">
        <div class="logo">
            
        </div>
        <div class="hero-text">
            <h1 style="font-size:60px; margin-top: 5px;">مساق فى بكم مرحبا</h1>
            <h1 style="font-size:40px; color:rgba(0,109,255,0.87); font-weight: bold;">شبكات</h1>
            <h2>الحساسية</h2>
        </div>
        <div>
            <a style="font-size:20px; margin-top: 5px; color: #cb0c2a" href="Link.html"><button> HTML
LINK </button></a> <!-- Button link --></a>
        </div>
        <div>
            <a style="font-size:20px; margin-top: 5px; color: #0fe0bd"
href="https://www.w3schools.com/tags/att_img_src.asp"> W3School</a>
        </div>
        <br>
        <div class="main" style="margin-bottom: 15px;">
            <div class="talking">
                <div style="text-align: center;">
                    <h1 class="textName">1191768 - طوافشة عمر</h1>
                </div>
                <div class="picture">
                    
                </div>
                <div class="textInfo">
                    <p style="font-size: 135%; ">أنجزت لقد : الدراسية الفصول في الراقصة المشاريع بعض ممثل الماضية</p>
                    <ul>
                        <li>ماركت سوبر بيانات قاعدة مشروع</li>
                    </ul>
                    <p style="font-size: 135%; ">هي مهاراتي بعض</p>
                    <ul>
                        <li>أخرى مشاكل على وتطبيقاتها المفاهيم تعلم على القدرة</li>
                        <li>.التصميم في جيد</li>
                    </ul>
                    <p style="font-size: 135%; ">هي هو اياتي</p>
                    <ul>
                        <li>القدم كرة</li>
                        <li>تونس</li>
                        <li>.التعلم</li>
                    </ul>
                </div>
            </div>
            <h5>.</h5>
            <div class="main" style="margin-top: 5px">
                <div class="talking2">
                    <div style="text-align: center;">
                        <h1 class="textName">1190823 - العواودة إحميدى</h1>
                    </div>
                    <div class="picture">
                        
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
```

```


الدراسية الفصول في الرائعة المشاريع بعض أنجزت لقد :>



ممثل الماضية



- بت 8 VHDL هيكل مقاومة جهاز



: هي مهاراتي بعض



- سكريبت الشيل ولغة الماتلاب استخدام
- التصميم تصويب مهارات



: هي هو اياتي



- القدم كرة
- المسلسلات متتابعة
- التطوعية الأعمال


```

3.4. Pages Design CSS File (main.css)

```

body {
    margin: 0;
    font-family: "Times New Roman", Times, serif;
    background-color: rgb(105, 105, 105);
    color: white;
}

.logopic {
    width: 90%;
}

.head-image {
    background-image: url("backgroung.jpg");
    background-color: #cccccc;
    width: 100%;
    height: 100vh;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
    position: relative;
}

.logo {
    margin: 5%;
    float: right;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
    width: 12%;
    height: 19vh;
}

.picture {
    float: right;
    background-position: right;
    text-align: right ;
    background-repeat: no-repeat;
    background-size: cover;
}

```

```
    clip-path: circle();
}

.hero-text {
    text-align: center;
    position: absolute;
    top: 42%;
    left: 0;
    width: 100%;
    color: white;
    background-color: rgba(255, 255, 255, 0.3);
}

.main {
    box-sizing: border-box;
}

.talking {
    width: 100%;
    float: right;
    height: 570px;
    background-color: #0f695a;
}

.talking2 {
    width: 100%;
    float: right;
    height: 570px;
    background-color: rebeccapurple;
}

.talking3 {
    width: 100%;
    float: right;
    height: 570px;
    background-color: #cb0c2a
}

.textInfo {
    margin-left: 10%;
    width: 80%;
    font-size: 120%;
}

.textName {
    font-size: 45px;
}
```

3.5. Button link HTML File (Link.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>HTML link</title>
</head>
<body>
    <h1 style="font-size:100px; margin-top: 5px; color: #0fe0bd">The team of Code of Duty</h1>
    <h2 style="font-size:30px; margin-top: 5px; color: #006">Second place locally in the google Hash
    code</h2>
    
    <div>
        <h2 style="font-size:30px; margin-top: 5px; color: #006">Members Names: </h2>
        <ul>
            <li>Ahmaide Al-Awadah</li>
            <li>Omar Tawafshah</li>
            <li>Eyab Ghafre</li>
        </ul>
    </div>

</body>
</html>
```