



BIRZEIT UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING

Computer Networks
ENCS3320

Project 1 Report

Prepared by:

Tariq Odeh 1190699

Qays Safa 1190880

Mahmoud Samara 1191602

Sec: 2

Instructor: Dr. Abdalkarim Awad

Date: 11th November 2021

Table of Contents

1. Part I:	1
1.1. Ping a device in the same network	1
1.2. ping b.root-servers.net	2
1.3. tracert b.root-servers.net	3
1.4. nslookup b.root-servers.net	4
2. Part II	5
2.1. Explanation	5
2.2. Response	5
2.3. Full Code with comments	6
3. Part III	7
3.1. Main Page	7
3.2. PNG Image	16
3.3. JPG Image	17
3.4. Sort By Price	19
3.5. Sort By Name	21
3.6. Error 404	24
3.7. Full Code with comments	26
3.8. HTML Code	30
3.9. CSS Code	35
4. References	37

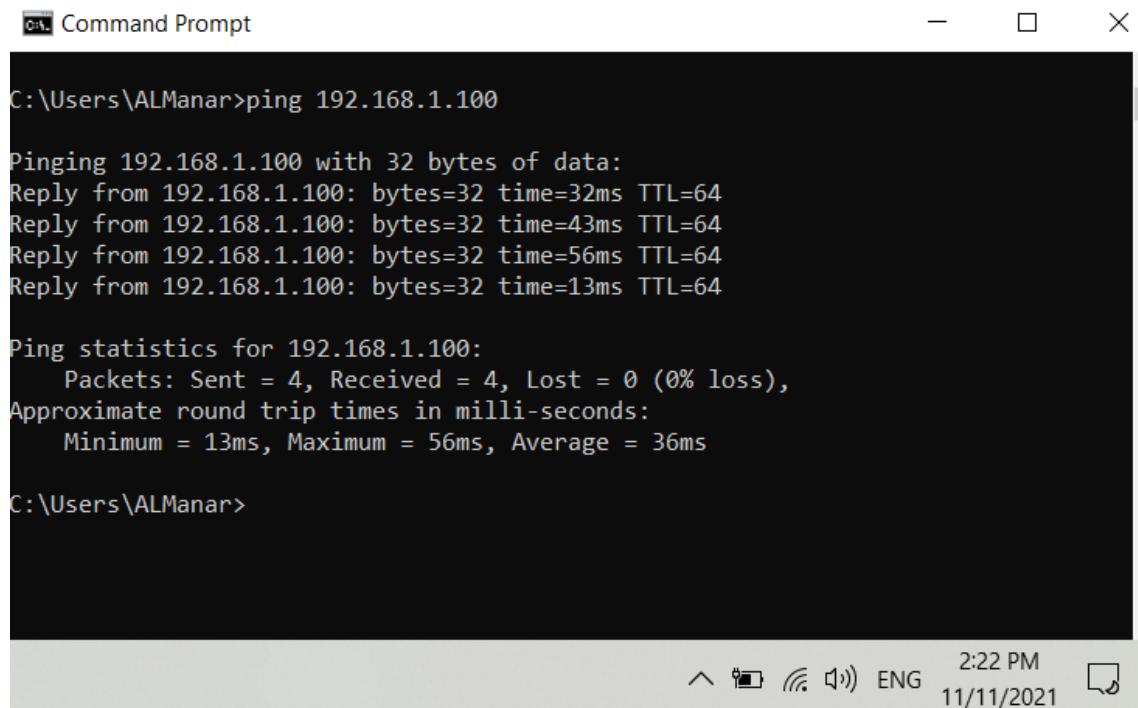
List Of Figures

Figure 1: ping Command - Device on the same Network.....	1
Figure 2: ping b.root-servers.net.....	2
Figure 3: tracert Command	3
Figure 4: nslookup Command.....	4
Figure 5: HTTP response from www.google.com webserver.....	5
Figure 6: localhost:6500 browser window – 1	7
Figure 7: localhost:6500 browser window – 2	8
Figure 8: localhost:6500 browser window – 3	8
Figure 9: localhost:6500 browser window – 4	9
Figure 10: localhost:6500 browser window - 5	9
Figure 11: localhost:6500/index.html browser window	10
Figure 12: localhost:6500/main.html browser window	10
Figure 13: Online HTML file browser window (button)	11
Figure 14: Local HTML file browser window (button).....	11
Figure 15: Main Page HTTP requests printed on command Line - 1	12
Figure 16: Main Page HTTP requests printed on command Line - 2	12
Figure 17: Main Page HTTP requests printed on command Line - 3	13
Figure 18: Main Page HTTP requests printed on command Line - 4	13
Figure 19: Localhost:6500 or Localhost:6500/index.html From phone	14
Figure 20: Local host:6500 and Online host (button) From phone	15
Figure 21: localhost:6500/mahmoud.png browser window	16
Figure 22: localhost:6500/mahmoud.png HTTP requests printed on command line.....	16
Figure 23: localhost:6500/qays.jpg browser window	17
Figure 24: localhost:6500/qays.jpg HTTP requests printed on command line	17
Figure 25: localhost:6500/qays.jpg and localhost:6500/ mahmoud.png browser from phone ...	18
Figure 26: text file that contains the names of the items	19
Figure 27: localhost:6500/SortByPrice browser window	19
Figure 28: SortByPrice HTTP requests printed on command line - 1	20
Figure 29: SortByPrice HTTP requests printed on command line – 2	20
Figure 30: text file that contains the names of the items	21
Figure 31: localhost:6500/SortByName browser window	21
Figure 32: SortByName HTTP requests printed on command line - 1.....	22
Figure 33: SortByName HTTP requests printed on command line - 2.....	22
Figure 34: localhost:6500/SortByPrice and localhost:6500/SortByName browser from phone .	23
Figure 35: localhost:6500/AAAA browser window	24
Figure 36: AAAA HTTP requests printed on command line - 1.....	24
Figure 37: AAAA HTTP requests printed on command line – 2.....	25
Figure 38: localhost:6500/AAAA browser window from phone	25

1. Part I:

1.1. Ping a device in the same network

As we can see the result in figure 1, it displays the total number of packets sent. As a result, the number of packets received is displayed (here we sent 4 packets where all packets have the same TTL, we received a response from 192.168.1.100), All packets are received with different delays. Also, we sent out 32 bytes of data and we got back 32 bytes and this is stable connection.



```
C:\Users\ALManar>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:
Reply from 192.168.1.100: bytes=32 time=32ms TTL=64
Reply from 192.168.1.100: bytes=32 time=43ms TTL=64
Reply from 192.168.1.100: bytes=32 time=56ms TTL=64
Reply from 192.168.1.100: bytes=32 time=13ms TTL=64

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 13ms, Maximum = 56ms, Average = 36ms

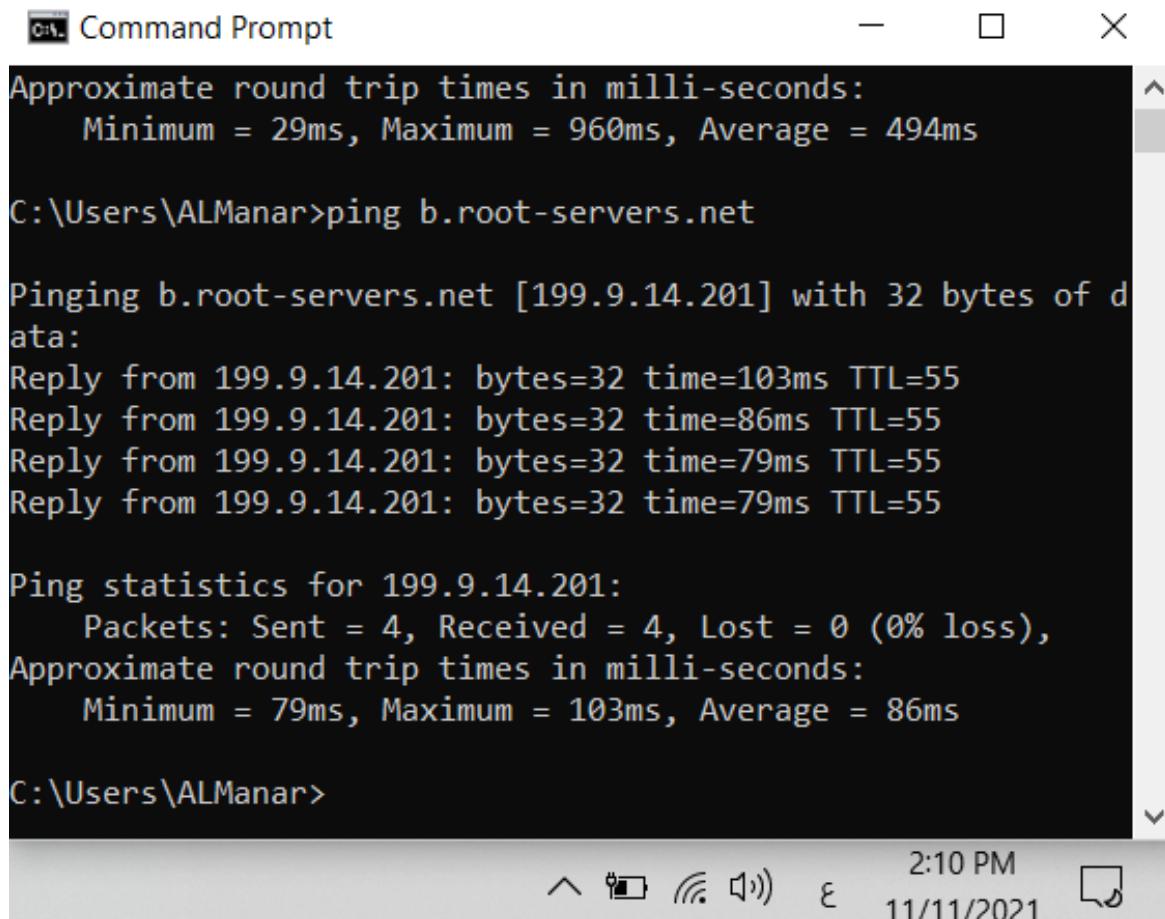
C:\Users\ALManar>
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command "ping 192.168.1.100" was entered, followed by its execution output. The output shows four replies from the target IP address, each with a different round-trip time (32ms, 43ms, 56ms, and 13ms). Below the replies, ping statistics are provided: 4 packets sent, 4 received, 0 lost (0% loss), with a minimum of 13ms, maximum of 56ms, and average of 36ms. The window has standard minimize, maximize, and close buttons at the top right. The taskbar at the bottom shows the date (11/11/2021) and time (2:22 PM).

Figure 1: ping Command - Device on the same Network

1.2. ping b.root-servers.net

As we can see the result in figure 2, it displays the total number of packets sent. As a result, the number of packets received is displayed (here we sent 4 packets where all packets have the same TTL, we sent to b.root-server.net four packets to the destination with IP 199.9.14.201 and the destination response back with the same four packets, all packets are received with different delays. Also, we sent out 32 bytes of data and we got back 32 bytes and this is stable connection.



```
Command Prompt
Approximate round trip times in milli-seconds:
    Minimum = 29ms, Maximum = 960ms, Average = 494ms

C:\Users\ALManar>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=103ms TTL=55
Reply from 199.9.14.201: bytes=32 time=86ms TTL=55
Reply from 199.9.14.201: bytes=32 time=79ms TTL=55
Reply from 199.9.14.201: bytes=32 time=79ms TTL=55

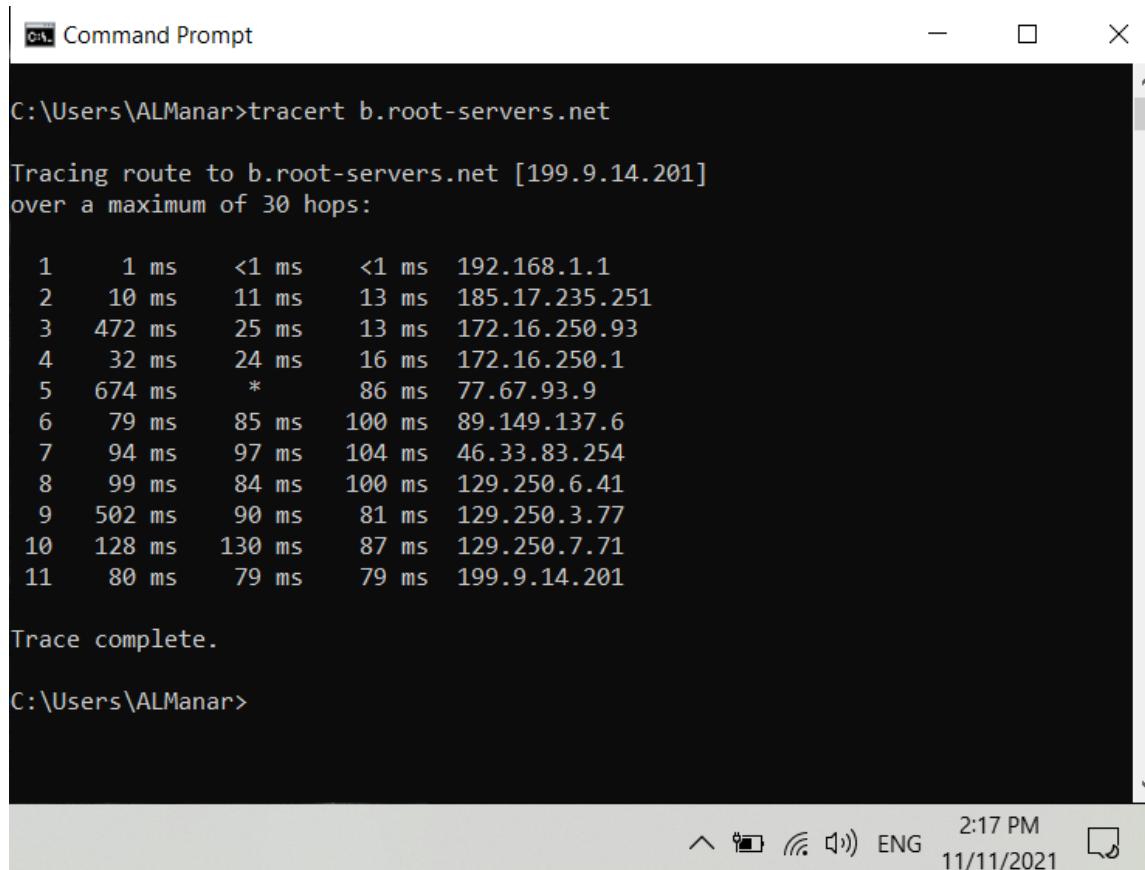
Ping statistics for 199.9.14.201:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 79ms, Maximum = 103ms, Average = 86ms

C:\Users\ALManar>
```

Figure 2: ping b.root-servers.net

1.3. tracert b.root-servers.net

The tracer command was used to display several details about a packet's path from the computer or device you're on to the destination you select. and this command sends 3 messages for every router and waits the response from the router, it continues in this process until it reaches the chosen IP. There are 5 columns in the end result, the number one is the hop number (TTL) and the time it takes for packets to make each hop is shown in the 3columns below (TTL), The last column is the server at the specified hop.



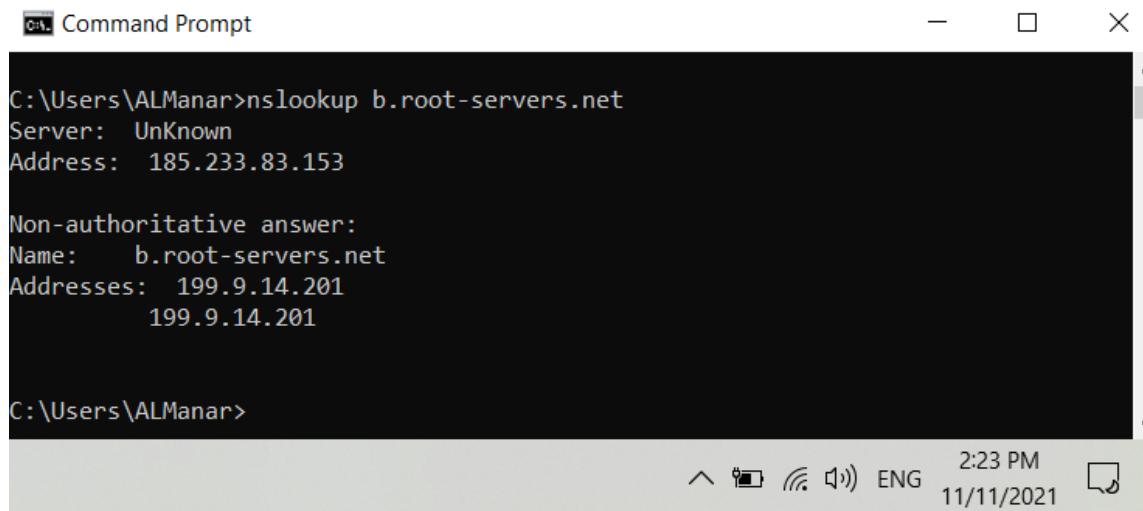
The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command "tracert b.root-servers.net" is entered, followed by its execution output. The output displays the tracing route to the destination IP [199.9.14.201] over a maximum of 30 hops. The results are presented in a table with five columns: Hop Number, Round Trip Time (ms), Minimum RTT (ms), Maximum RTT (ms), and Destination IP. The trace completes after 11 hops, reaching the final destination. The system tray at the bottom right shows the date and time as 11/11/2021 and 2:17 PM respectively.

Hop	T1	T2	T3	Server
1	1 ms	<1 ms	<1 ms	192.168.1.1
2	10 ms	11 ms	13 ms	185.17.235.251
3	472 ms	25 ms	13 ms	172.16.250.93
4	32 ms	24 ms	16 ms	172.16.250.1
5	674 ms	*	86 ms	77.67.93.9
6	79 ms	85 ms	100 ms	89.149.137.6
7	94 ms	97 ms	104 ms	46.33.83.254
8	99 ms	84 ms	100 ms	129.250.6.41
9	502 ms	90 ms	81 ms	129.250.3.77
10	128 ms	130 ms	87 ms	129.250.7.71
11	80 ms	79 ms	79 ms	199.9.14.201

Figure 3: tracert Command

1.4. nslookup b.root-servers.net

As we can see in finger 4, when we used nslookup that is used to diagnose DNS problems first it prints the server and the address Unknown server is my router with the address 192.168.1.1 and prints the name and the 3 addresses of the server which is the host that we sent.



```
Command Prompt
C:\Users\ALManar>nslookup b.root-servers.net
Server: UnKnown
Address: 185.233.83.153

Non-authoritative answer:
Name: b.root-servers.net
Addresses: 199.9.14.201
          199.9.14.201

C:\Users\ALManar>
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The user has run the command "nslookup b.root-servers.net". The output indicates that the server is "UnKnown" and its address is "185.233.83.153". A "Non-authoritative answer" is provided, showing the name "b.root-servers.net" and two addresses: "199.9.14.201" and "199.9.14.201". The system tray at the bottom right shows the date and time as "11/11/2021 2:23 PM".

Figure 4: nslookup Command

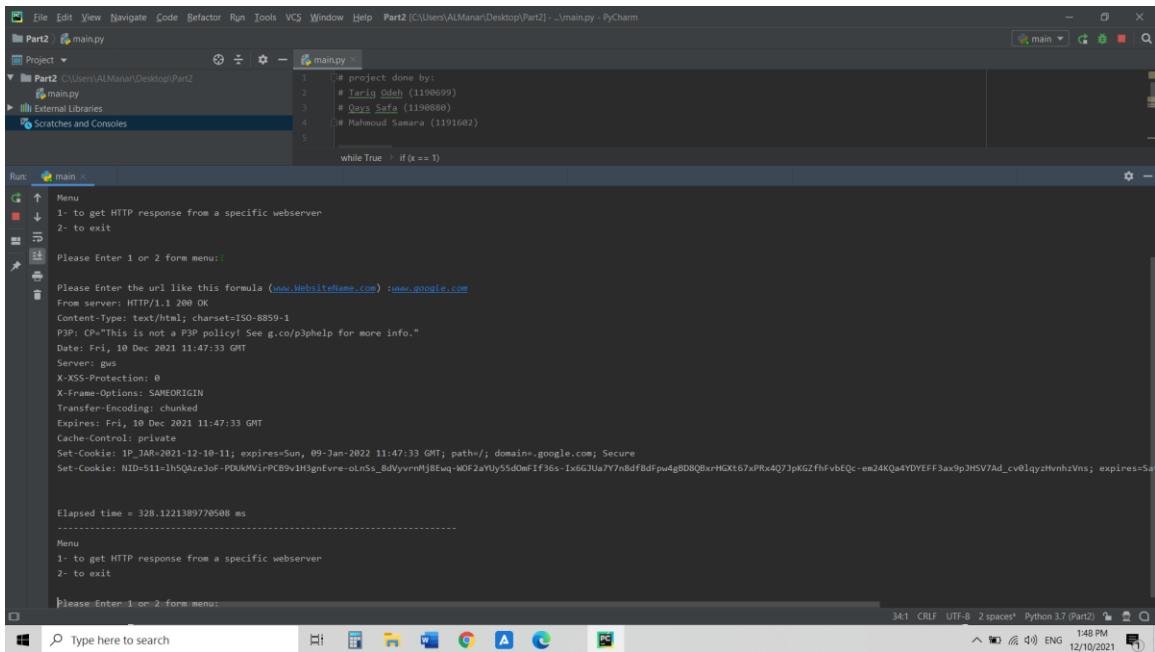
2. Part II

2.1. Explanation

In this part we are asking the user to enter the website name to get HTTP response from a specific webserver and calculate the http response time for this webserver. First, we will create a socket object then we sent data and started the time and received data to finish the time and finally we found the response time. In addition, we will display the response using HEAD method.

2.2. Response

The response information shows that HTTP response status codes is 200 OK (request succeeded, requested object later in this message), with content type (text/html) and response time = 328.12 ms.



The screenshot shows the PyCharm IDE interface. The project navigation bar at the top indicates the current file is 'main.py'. The code editor on the left contains a simple Python script:

```
# project done by:  
# Tarig Isheh (1198699)  
# Qays Safa (1190886)  
# Mahmoud Samara (1191602)  
  
while True : if (x == 1):  
  
Run: main >  
Menu  
1- to get HTTP response from a specific webserver  
2- to exit  
  
Please Enter 1 or 2 form menu:  
  
Please Enter the url like this formula (www.WebsiteName.com) :www.google.com  
From server: HTTP/1.1 200 OK  
Content-Type: text/html; charset=ISO-8859-1  
P3P: ("This is not a P3P policy! See g.co/p3phelp for more info."  
Date: Fri, 10 Dec 2021 11:47:33 GMT  
Server: gws  
X-XSS-Protection: 0  
X-Frame-Options: SAMEORIGIN  
Transfer-Encoding: chunked  
Expires: Fri, 10 Dec 2021 11:47:33 GMT  
Cache-Control: private  
Set-Cookie: IP_3AR=2021-12-10-11; expires=Sun, 09-Jan-2022 11:47:33 GMT; path=/; domain=.google.com; Secure  
Set-Cookie: NID=511=lh5QAzelof-POUkMvixPCB9vIH3gnEvre-oLnSs_BdVyyvrnRj8Ewq-WOF2aYUy55D0mFf36s-1x6GJUa7Y7n8df8dFpw4g8D8QbxrHGxt67xPRx4d7TpKGZfFvxEQc-em24KQa4YDYEFF3axOpjHSV7Ad_cv0lqyzHvnhzVns; expires=Sat, 10-Dec-2022 11:47:33 GMT; path=/; domain=.google.com; Secure  
  
Elapsed time = 328.1221389770508 ms  
-----  
Menu  
1- to get HTTP response from a specific webserver  
2- to exit  
  
Please Enter 1 or 2 form menu:
```

The terminal window below the code editor displays the output of the script. It shows the user entering '1' to get the response from 'www.google.com'. The response includes standard HTTP headers like 'Content-Type', 'P3P', and 'Set-Cookie', along with the elapsed time of 328.12 ms. The PyCharm interface also shows the system tray with icons for battery, signal, and clock.

Figure 5: HTTP response from www.google.com webserve

2.3. Full Code with comments

3. Part III

In this part we will use socket programming, implement a web server in python that is listening on port 6500.

3.1. Main Page

<http://localhost:6500> or <http://localhost:6500/index.html> or
<http://localhost:6500/main.html>

In the main page we used html language to design it and to put names, numbers and information about each student we used css language to arrange the boxes and the full design.

Main Page in the browser window:

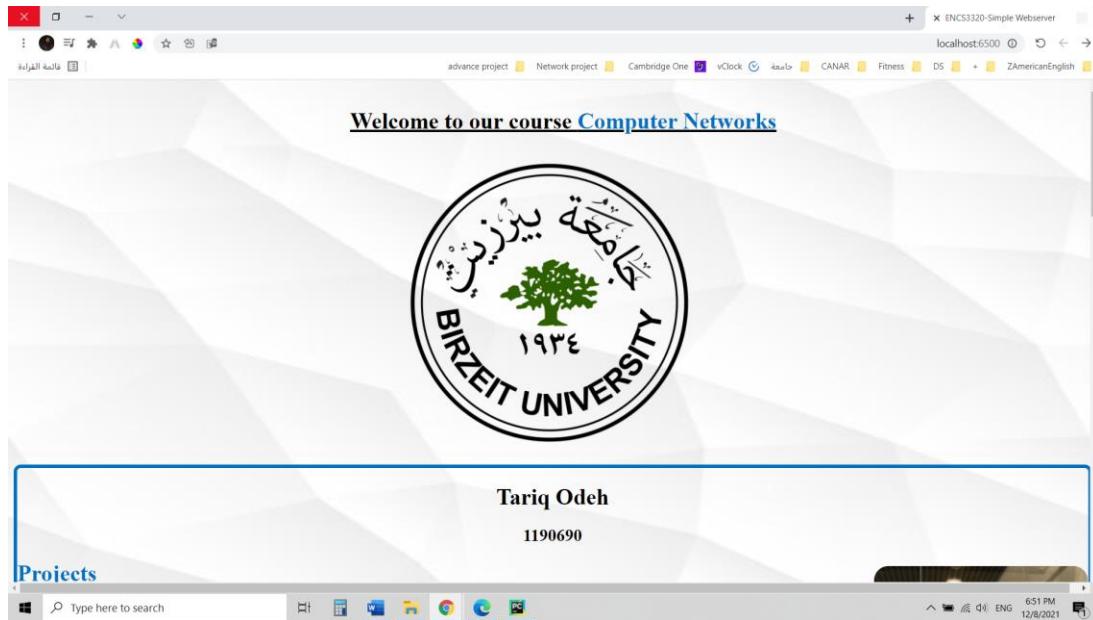


Figure 6: localhost:6500 browser window – 1

Tariq Odeh
1190690

Projects

- An 8-bit Comparator for signed 2s complement representation numbers.
- An educational application that uses augmented reality technology.
- System to manage patients data in a hospital.

Skills

- quick mathematical and physical analysis.
- scientific research.
- Planing.

Hobbies

- Reading.
- Football.
- Squash.

Figure 7: localhost:6500 browser window – 2

Qays Safa
1190880

Projects

- A Java program for managing patient information in a hospital.
- Establishing a company specializing in health food products.
- Making a simple calculator in 8086 program in Orga course.

Skills

- Marketing.
- Teamwork.
- Problem Solving.

Hobbies

- Programming.
- Volleyball
- cycling.

Figure 8: localhost:6500 browser window – 3

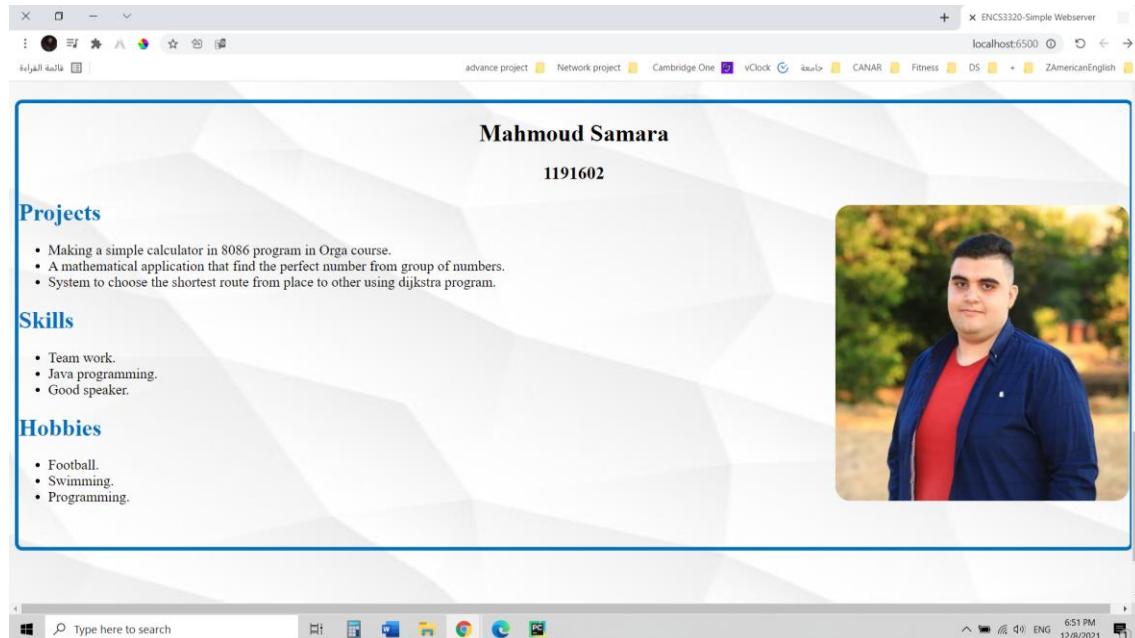


Figure 9: localhost:6500 browser window – 4

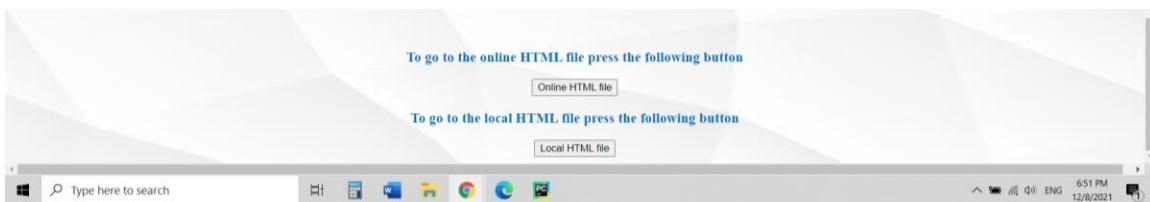


Figure 10: localhost:6500 browser window - 5

In the following figures, it will give the same results as the previous figures, but it will be using localhost:6500/index.html and localhost:6500/main.html in the browser.

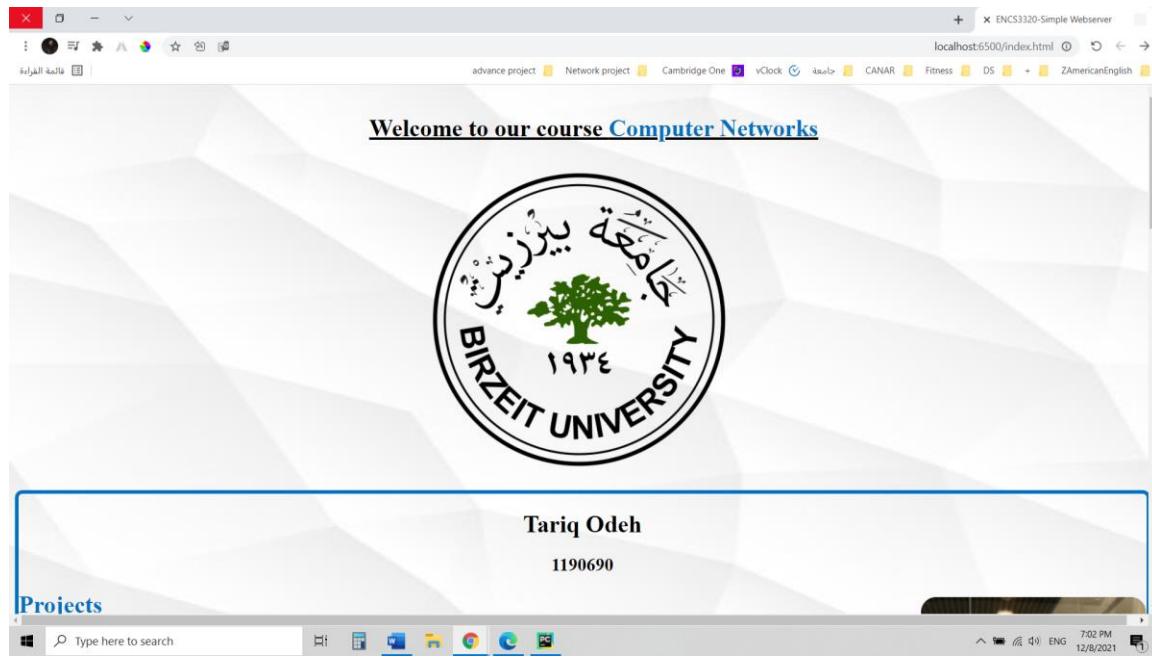


Figure 11: localhost:6500/index.html browser window

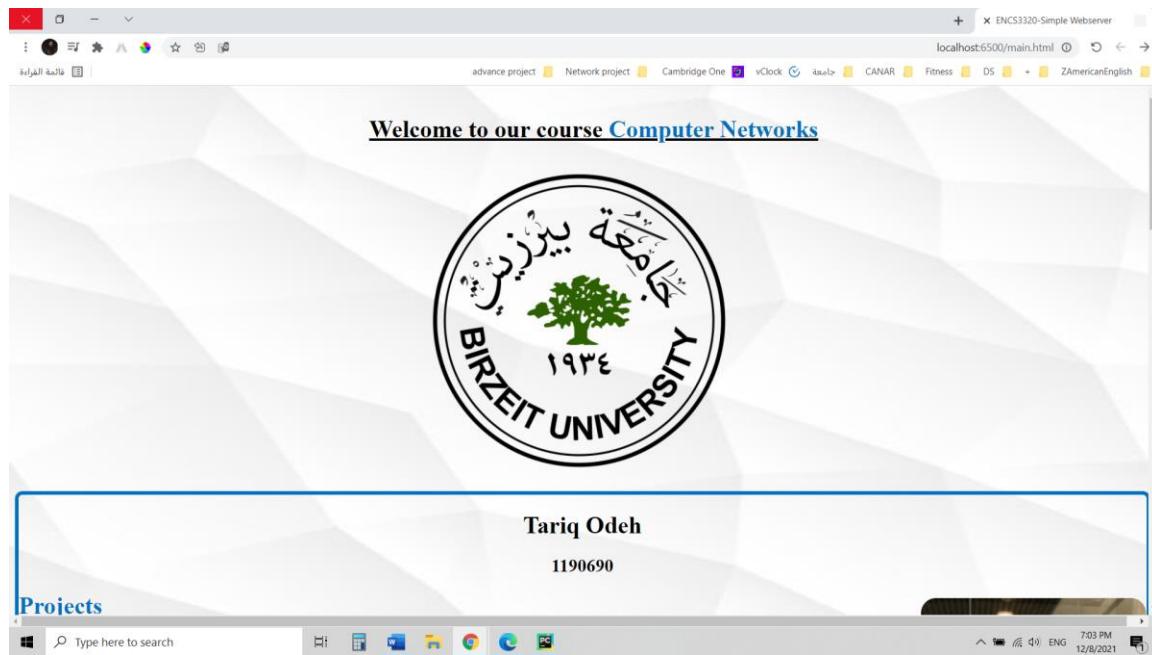


Figure 12: localhost:6500/main.html browser window

In the following figures we will see where we will go when we press online html file button and same thing for local html file button.

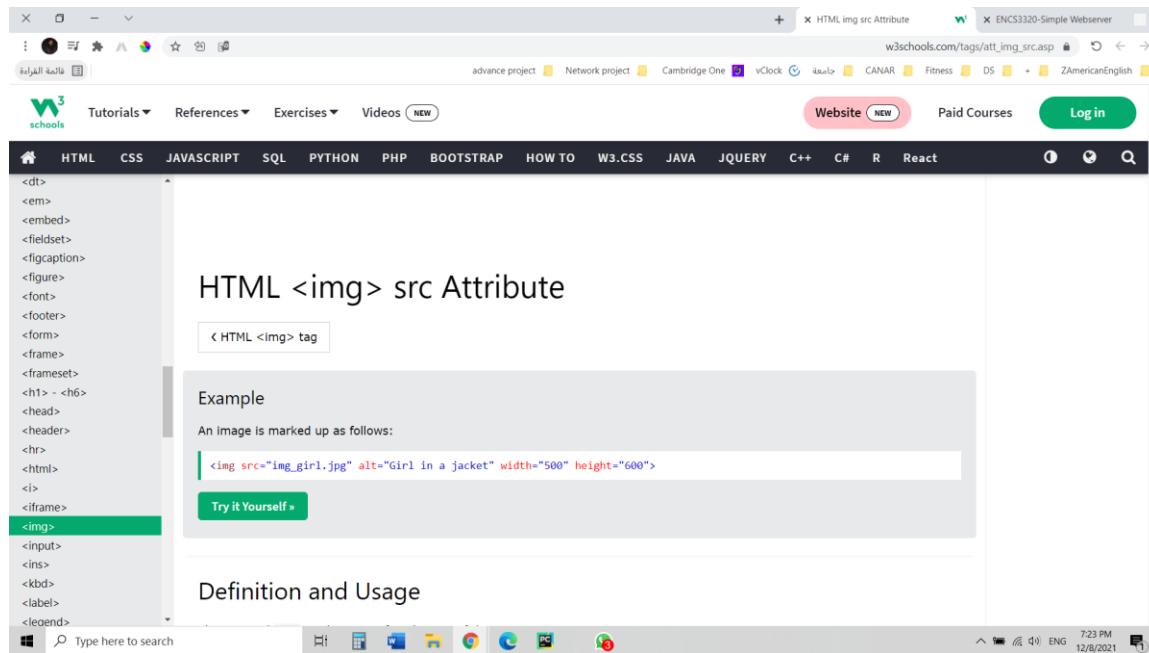


Figure 13: Online HTML file browser window (button)

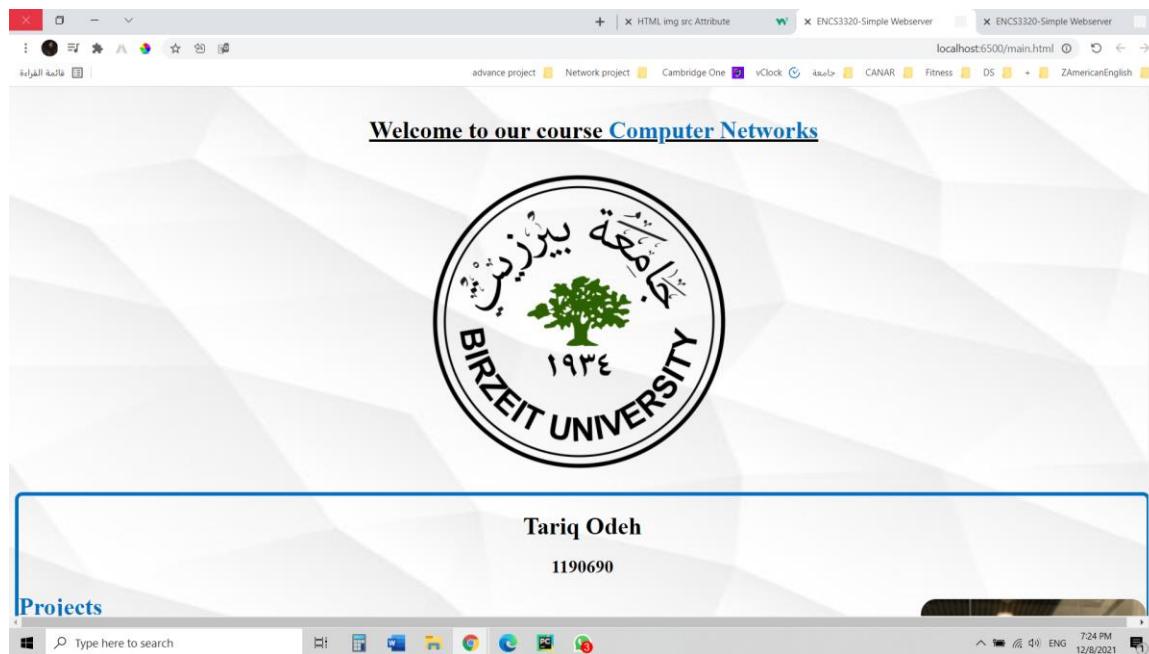
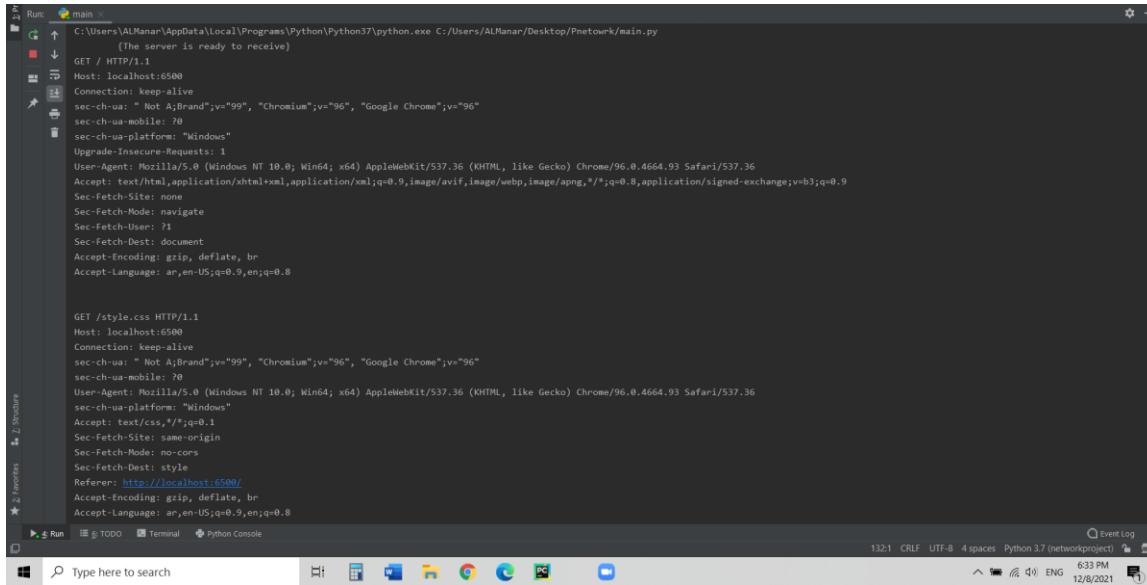


Figure 14: Local HTML file browser window (button)

Requests:

As we can see in the figures below that shown the http request for localhost:6500. We can see that http request is OK and everything is right, and keepalive means persistent, after that it specified all the contents in localhost with accepted content type. As we can see there are many responses like mahmoud or tariq image also the css file (style.css), and the method that we use is GETmethod.

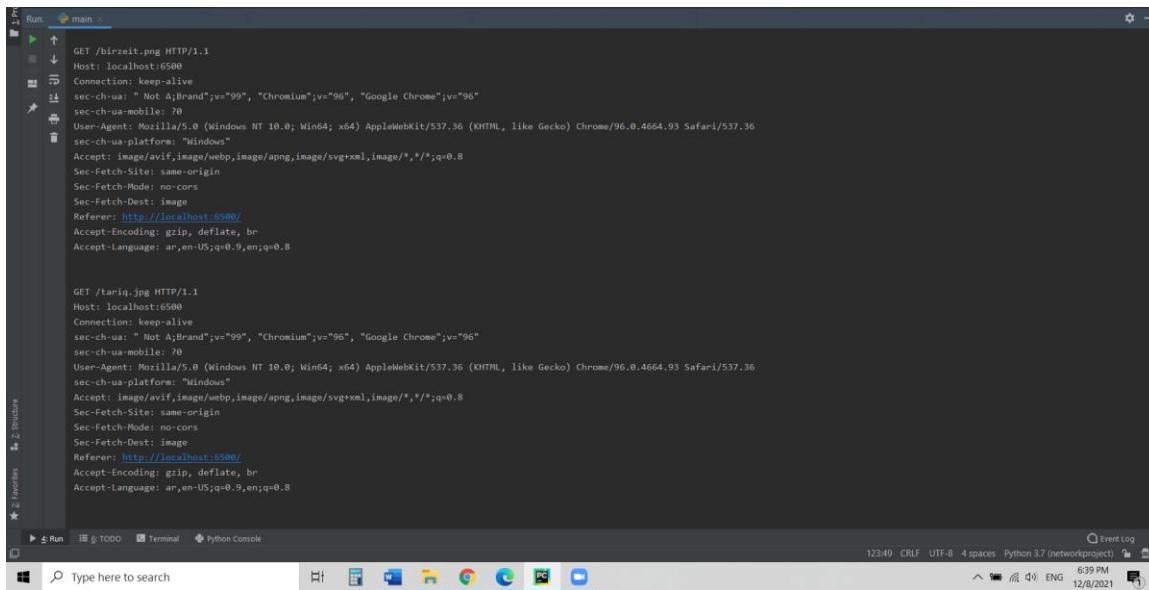


```
C:\Users\AlManar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/AlManar/Desktop/Pnetwork/main.py
  (The server is ready to receive)
  GET / HTTP/1.1
  Host: localhost:6500
  Connection: keep-alive
  sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
  sec-ch-ua-mobile: 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/96.0.4664.93 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: ar,en-US;q=0.9,en;q=0.8

  GET /style.css HTTP/1.1
  Host: localhost:6500
  Connection: keep-alive
  sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
  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/96.0.4664.93 Safari/537.36
  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-US;q=0.9,en;q=0.8

  GET /birzeit.png HTTP/1.1
  Host: localhost:6500
  Connection: keep-alive
  sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
  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/96.0.4664.93 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:6500/
  Accept-Encoding: gzip, deflate, br
  Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 15: Main Page HTTP requests printed on command Line - 1



```
C:\Users\AlManar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/AlManar/Desktop/Pnetwork/main.py
  (The server is ready to receive)
  GET /birzeit.png HTTP/1.1
  Host: localhost:6500
  Connection: keep-alive
  sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
  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/96.0.4664.93 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:6500/
  Accept-Encoding: gzip, deflate, br
  Accept-Language: ar,en-US;q=0.9,en;q=0.8

  GET /tariq.jpg HTTP/1.1
  Host: localhost:6500
  Connection: keep-alive
  sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
  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/96.0.4664.93 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:6500/
  Accept-Encoding: gzip, deflate, br
  Accept-Language: ar,en-US;q=0.9,en;q=0.8

  GET / HTTP/1.1
  Host: localhost:6500
  Connection: keep-alive
  sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
  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/96.0.4664.93 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: ar,en-US;q=0.9,en;q=0.8
```

Figure 16: Main Page HTTP requests printed on command Line - 2

```
Run: main ->
GET /qays.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8

GET /mahmoud.png HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8

GET /back.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 17: Main Page HTTP requests printed on command Line - 3

```
Run: main ->
GET /qays.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8

GET /mahmoud.png HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8

GET /back.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 18: Main Page HTTP requests printed on command Line - 4

Screenshot from another device (phone):

To test our program from other divide: first we must know ipv4 for the device we work on it (origin device), then we make run for the code and note that both origin device and other device are on the same network. Finally, we used the following IP address to open the project: 192.168.1.1.158:6500.

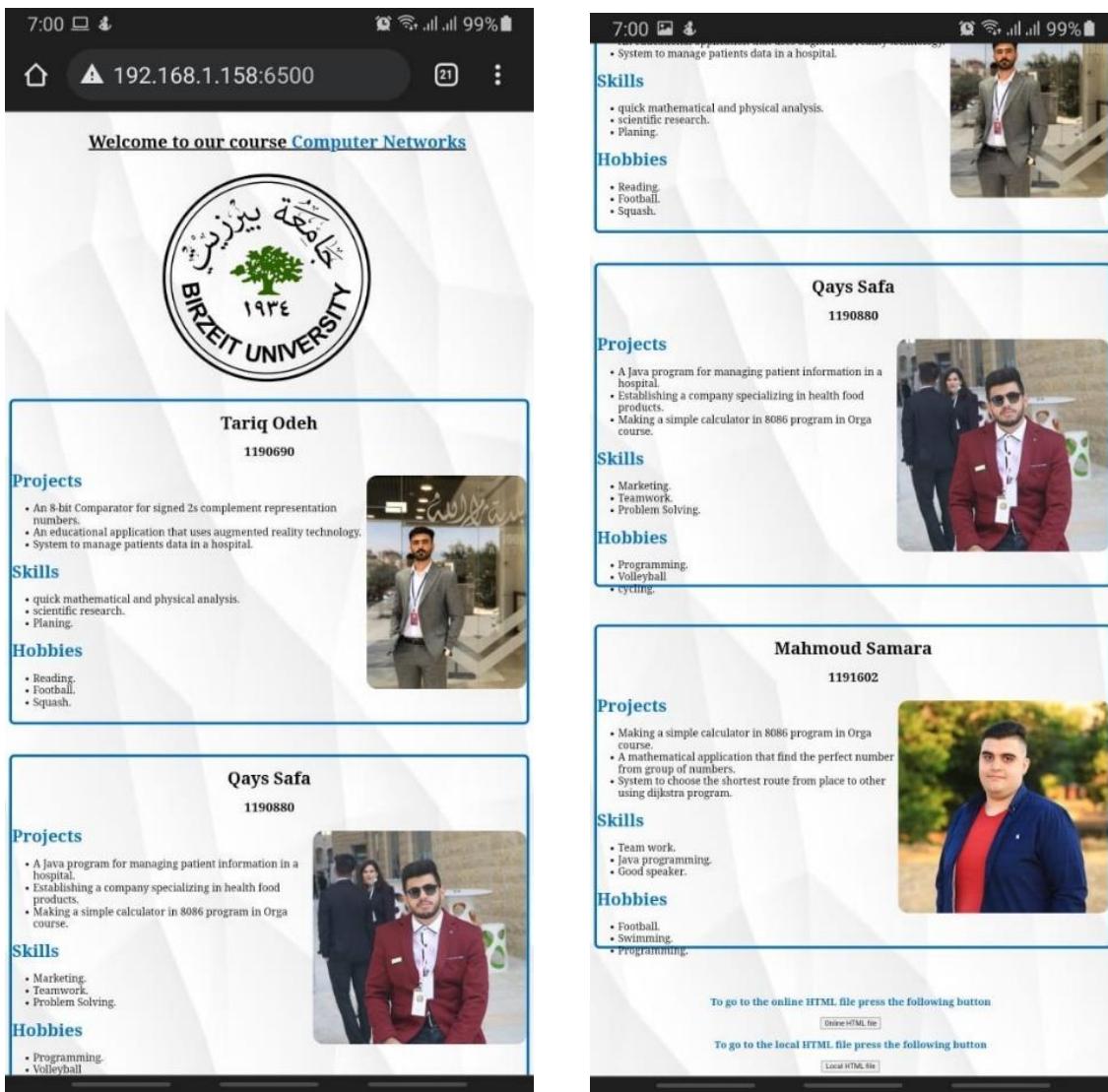


Figure 19: Localhost:6500 or Localhost:6500/index.html From phone

Welcome to our course Computer Networks

BIRZEIT UNIVERSITY

Tariq Odeh
1190690

Projects

- An 8-bit Comparator for signed 2s complement representation numbers.
- An educational application that uses augmented reality technology.
- System to manage patients data in a hospital.

Skills

- quick mathematical and physical analysis.
- Scientific research.
- Planing.

Hobbies

- Reading.
- Football.
- Squash.

Qays Safa
1190880

Projects

- A Java program for managing patient information in a hospital.
- Establishing a company specializing in health food products.
- Making a simple calculator in 8086 program in Orga course.

Skills

- Marketing.
- Teamwork.
- Problem Solving.

HTML src Attribute

< HTML tag

Example

An image is marked up as follows:

```

```

Try it Yourself »

Definition and Usage

Figure 20: Local host:6500 and Online host (button) From phone

3.2. PNG Image

<http://localhost:6500/mahmoud.png>

Main Page in the browser window:

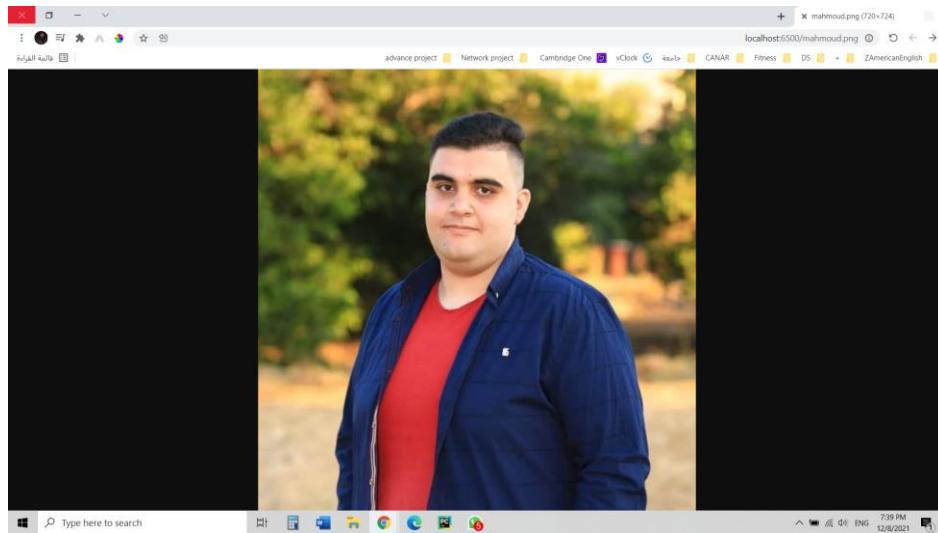


Figure 21: localhost:6500/mahmoud.png browser window

Requests:

As we can see in the figure below that shown the http request for an image with type (png). We can see that http request is OK and everything is right, and keepalive means persistent, after that it specified all the contents in localhost with accepted content type. As we can see there are a response for mahmoud image and in the website, it appears only the image as what we asked.

```
File Edit View Navigate Code Behaivor Run Tools VCS Window Help main.py [C:\Users\AlManar\Desktop\Pnetowrk] ...main.py - PyCharm
Pnetowrk > main.py
Project < main >
Run: GET /mahmoud.png HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8

GET /favicon.ico HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/mahmoud.png
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 22: localhost:6500/mahmoud.png HTTP requests printed on command line

3.3. JPG Image

<http://localhost:6500/qays.jpg>

Main Page in the browser window:

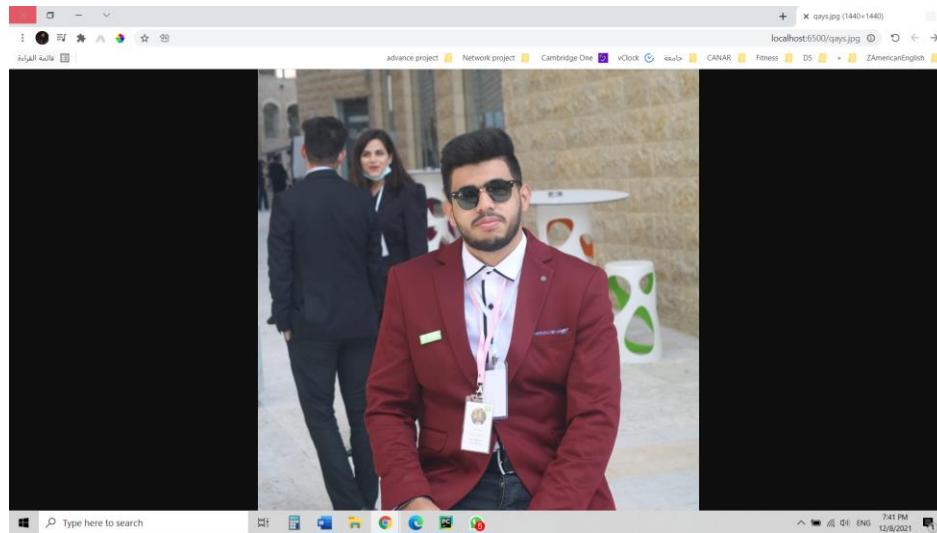


Figure 23: localhost:6500/qays.jpg browser window

Requests:

As we can see in the figure below that shown the http request for an image with type (jpg). We can see that http request is OK and everything is right, and keepalive means persistent, after that it specified all the contents in localhost with accepted content type. As we can see there are a response for qays image and in the website, it appears only the image as what we asked.

```
File Edit View Navigate Code Behaivor Run Tools VCS Window Help main.py [C:\Users\ALManar\Desktop\Pnetwerk] -->main.py - PyCharm
Project main.py
Run main
C:\Users\ALManar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/ALManar/Desktop/Pnetwerk/main.py
  (The server is ready to receive)
GET /favicon.ico HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
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/96.0.4664.93 Safari/537.36
sec-ch-ua-device: "mobile"
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: https://localhost:6500/qays.jpg
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 24: localhost:6500/qays.jpg HTTP requests printed on command line

Screenshot from another device (phone):

We used the following IP address to open the project:

192.168.1.1.158:6500/qays.jpg

192.168.1.1.158:6500/mahmoud.png

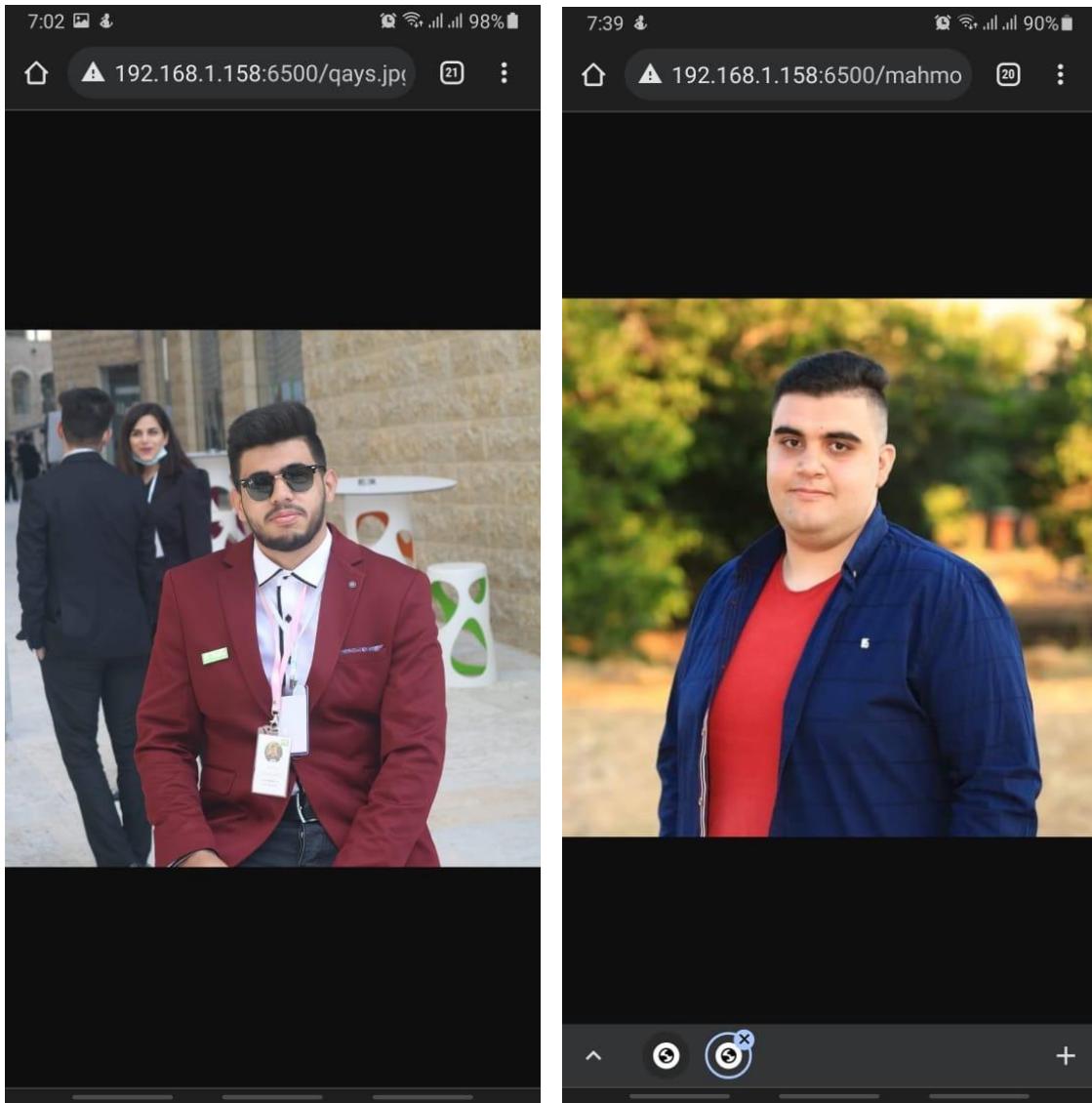
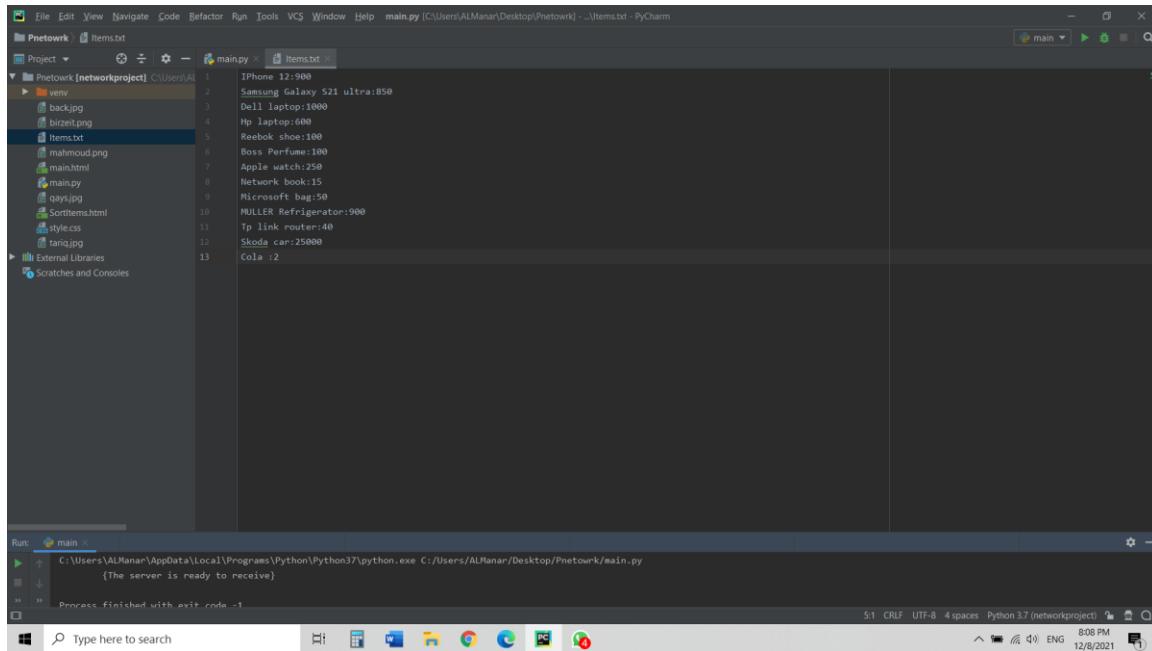


Figure 25: localhost:6500/qays.jpg and localhost:6500/ mahmoud.png browser from phone

3.4. Sort By Price

<http://localhost:6500/SortByPrice>

Text file:



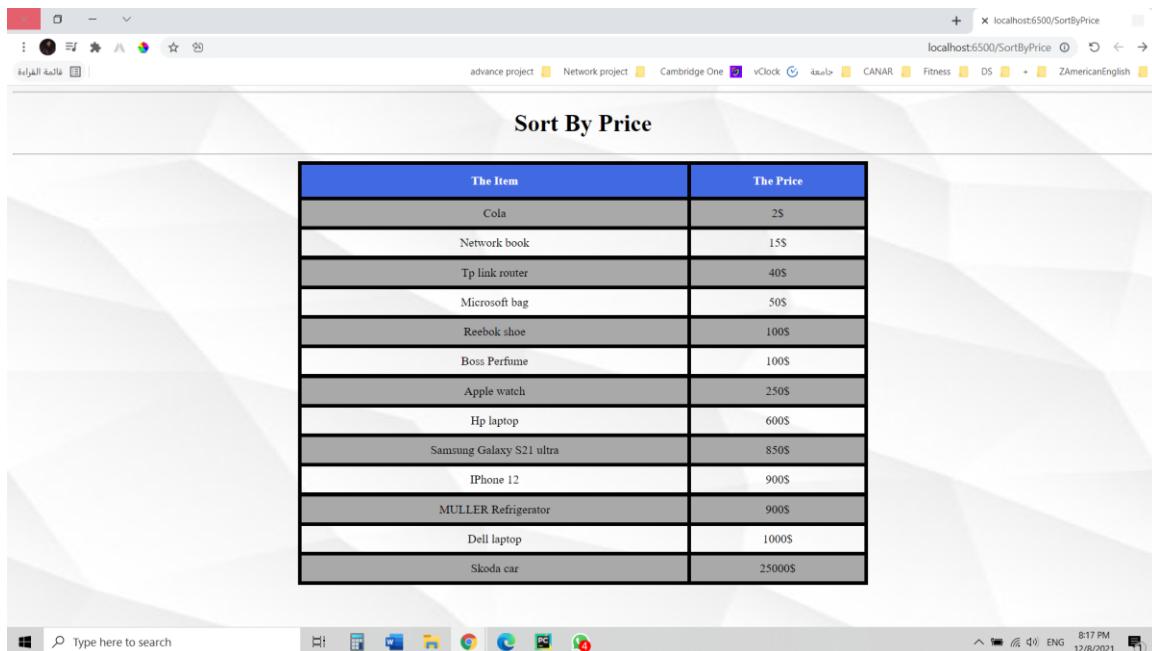
The screenshot shows the PyCharm IDE interface. The left sidebar displays a project structure for 'Pnetwrk [networkproject]' containing files like 'main.py', 'Items.txt', 'venv', 'backup.jpg', 'birzeit.png', 'mahmoud.png', 'main.html', 'main.py', 'qays.png', 'SortItems.html', 'style.css', and 'tariq.jpg'. The right pane shows the content of 'Items.txt':

```
1 iPhone 12:900
2 venv
3 Samsung Galaxy S21 ultra:850
4 Dell laptop:1000
5 Hp laptop:600
6 Reebok shoe:100
7 Boss Perfume:100
8 Apple watch:250
9 Network book:15
10 Microsoft bag:50
11 MULLER Refrigerator:900
12 Tp link router:40
13 Skoda car:25000
14 Cola :2
```

The bottom status bar indicates the run configuration is 'main' and the command is 'C:\Users\ALManar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/ALManar/Desktop/Pnetwrk/main.py'. The terminal output shows '(The server is ready to receive)' and 'Process finished with exit code 0'.

Figure 26: text file that contains the names of the items

Main Page in the browser window:



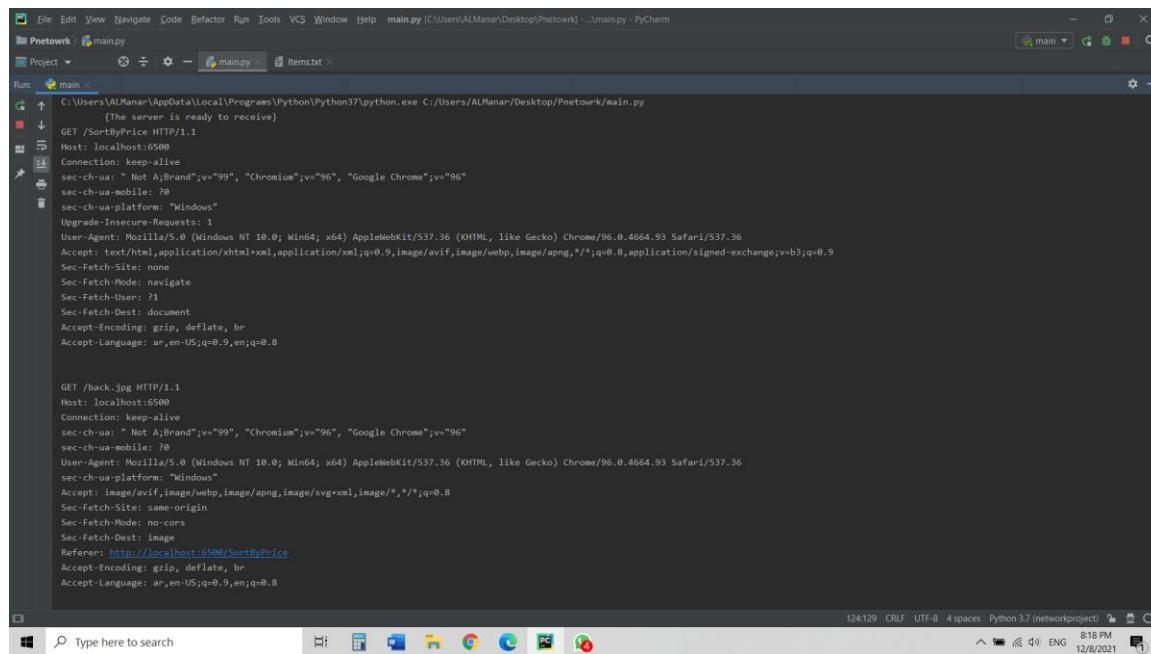
The screenshot shows a web browser window with the URL 'localhost:6500/SortByPrice'. The page title is 'Sort By Price'. Below it is a table with two columns: 'The Item' and 'The Price'.

The Item	The Price
Cola	25
Network book	15\$
Tp link router	40\$
Microsoft bag	50\$
Reebok shoe	100\$
Boss Perfume	100\$
Apple watch	250\$
Hp laptop	600\$
Samsung Galaxy S21 ultra	850\$
iPhone 12	900\$
MULLER Refrigerator	900\$
Dell laptop	1000\$
Skoda car	25000\$

Figure 27: localhost:6500/SortByPrice browser window

Requests:

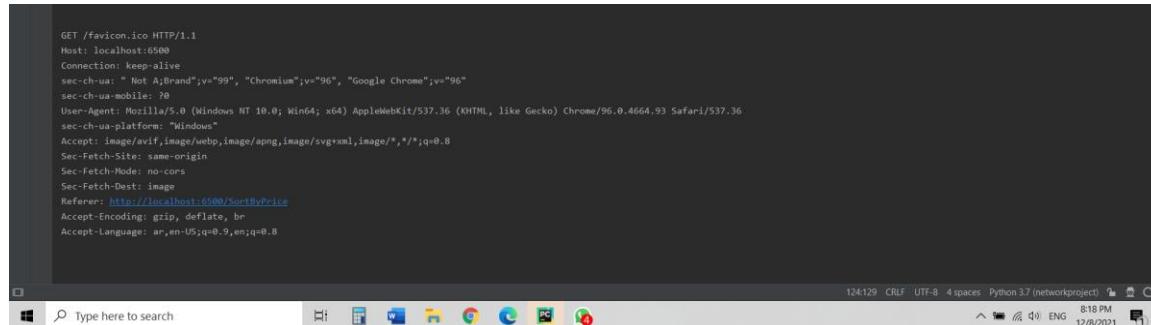
In the figure below we can see the http response for sort by price it accepted content type of text/plain, and the design of the page was arranged using html code and it was put in the main python. Note that the items were read from a text file in the python program.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help main.py [C:\Users\ALManar\Desktop\Pnetowrk] ...main.py - PyCharm
Pnetowrk main.py
Project main.py Item.txt
Run: main.x
C:\Users\ALManar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/ALManar/Desktop/Pnetowrk/main.py
(The server is ready to receive)
GET /SortByPrice HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
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/96.0.4664.93 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: ar,en-US;q=0.9,en;q=0.8

GET /back.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
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/96.0.4664.93 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:6500/SortByPrice
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 28: SortByPrice HTTP requests printed on command line - 1



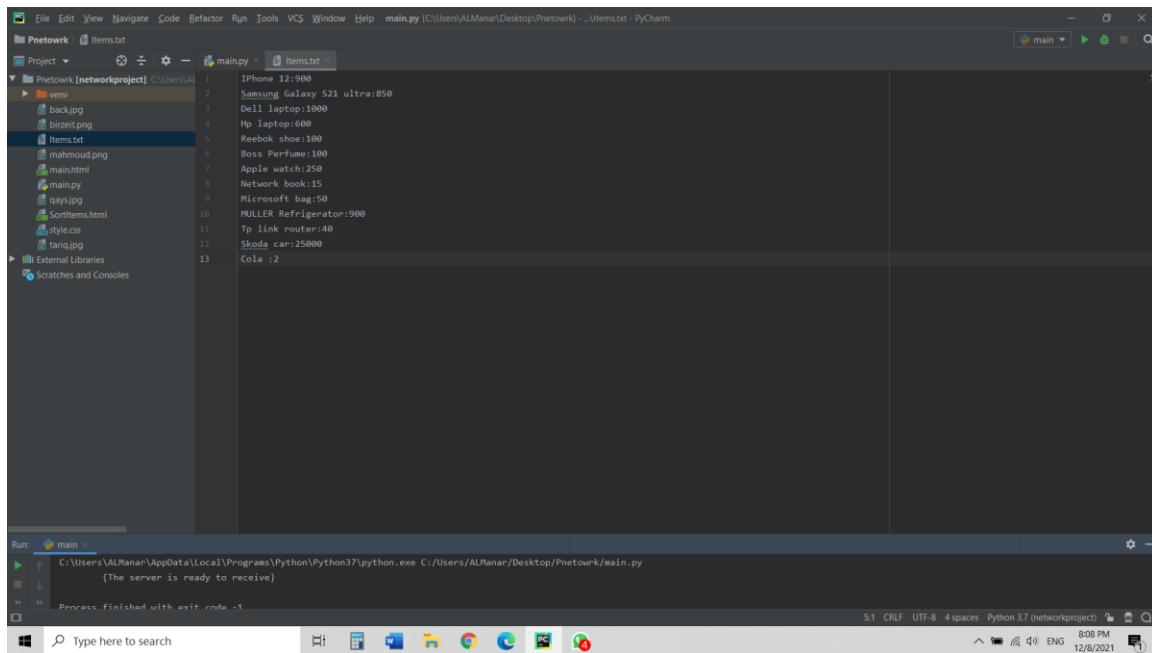
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help main.py [C:\Users\ALManar\Desktop\Pnetowrk] ...main.py - PyCharm
Pnetowrk main.py Item.txt
Run: main.x
C:\Users\ALManar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/ALManar/Desktop/Pnetowrk/main.py
(GET /favicon.ico HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
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/96.0.4664.93 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:6500/SortByPrice
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8)
```

Figure 29: SortByPrice HTTP requests printed on command line – 2

3.5. Sort By Name

<http://localhost:6500/SortByName>

Text file:



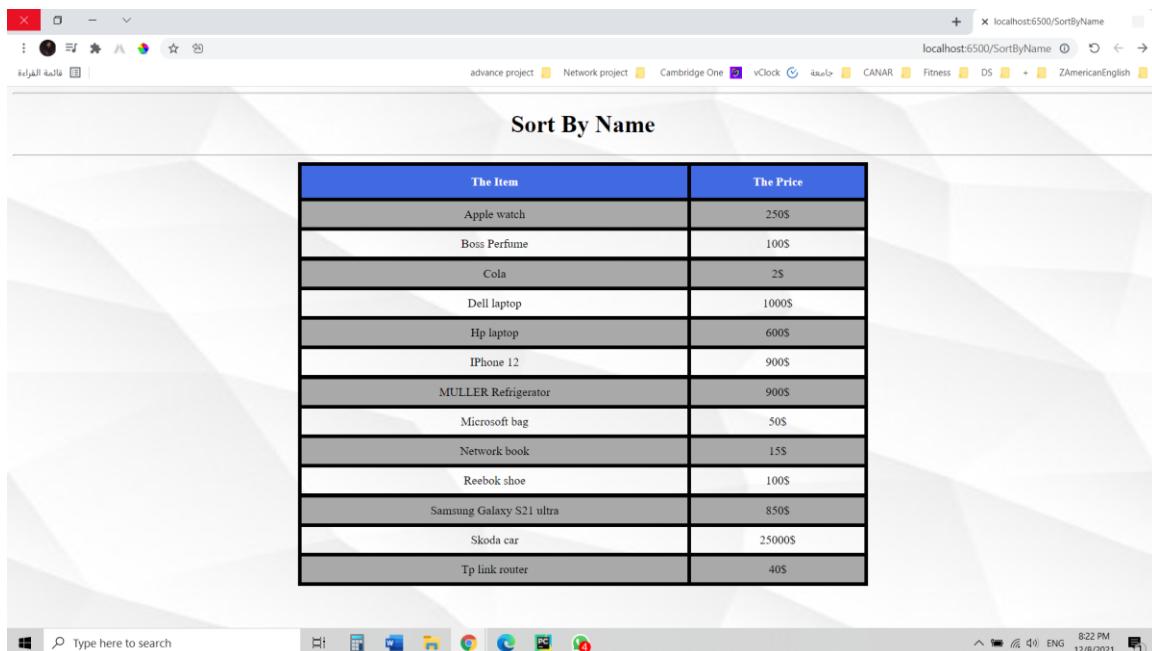
The screenshot shows a PyCharm interface with a project named "Pnetwrk". The "Items.txt" file is open in the editor, displaying a list of items and their prices. The content of the file is as follows:

```
1 iPhone 12:900
2 venv
3 Samsung Galaxy S21 ultra:850
4 Dell laptop:1000
5 Hp laptop:600
6 Reebok shoe:100
7 Boss Perfume:100
8 Apple watch:250
9 Network book:15
10 Microsoft bag:50
11 MULLER Refrigerator:900
12 Tp link router:40
13 Skoda car:25000
14 Cola :2
```

The "Run" tab at the bottom shows the command: C:\Users\ALManar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/ALManar/Desktop/Pnetwrk/main.py. The output says: "The server is ready to receive". The status bar at the bottom right indicates: 5.1 CRLF UTF-8 4 spaces Python 3.7 (networkproject) 8:00 PM 12/6/2021.

Figure 30: text file that contains the names of the items

Main Page in the browser window:



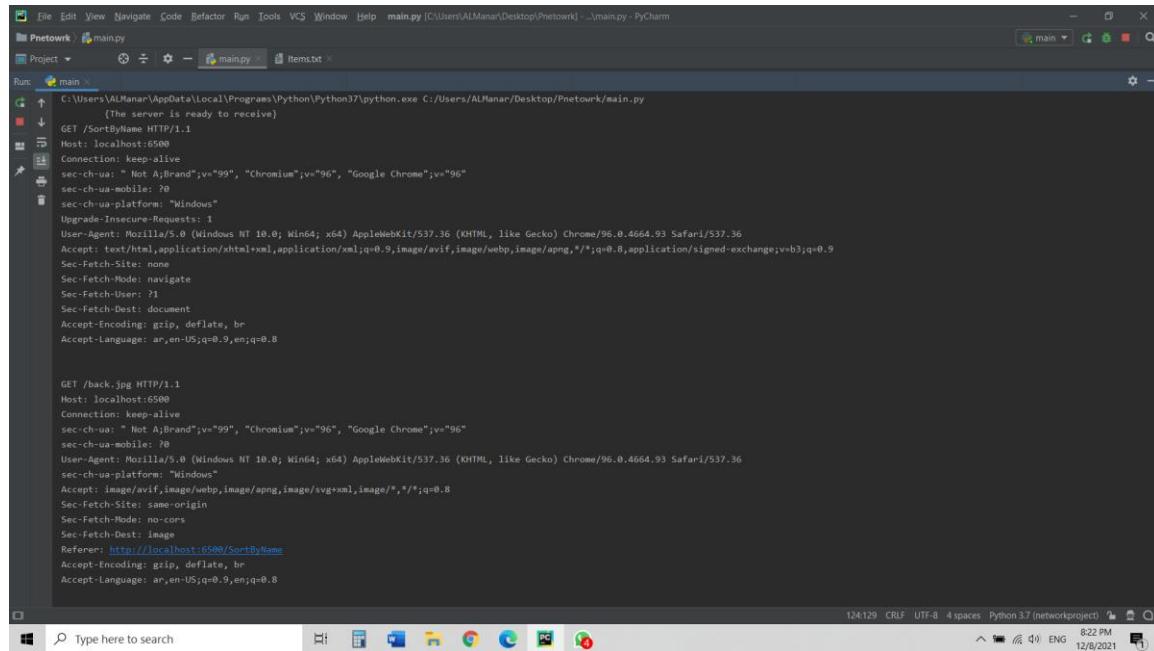
The screenshot shows a browser window with the URL localhost:6500/SortByName. The title of the page is "Sort By Name". Below the title is a table with two columns: "The Item" and "The Price". The table lists the same 14 items from the text file, sorted by name. The table has alternating row colors (light gray and white). The status bar at the bottom right indicates: 8:22 PM 12/6/2021.

The Item	The Price
Apple watch	250\$
Boss Perfume	100\$
Cola	2\$
Dell laptop	1000\$
Hp laptop	600\$
iPhone 12	900\$
MULLER Refrigerator	900\$
Microsoft bag	50\$
Network book	15\$
Reebok shoe	100\$
Samsung Galaxy S21 ultra	850\$
Skoda car	25000\$
Tp link router	40\$

Figure 31: localhost:6500/SortByName browser window

Requests:

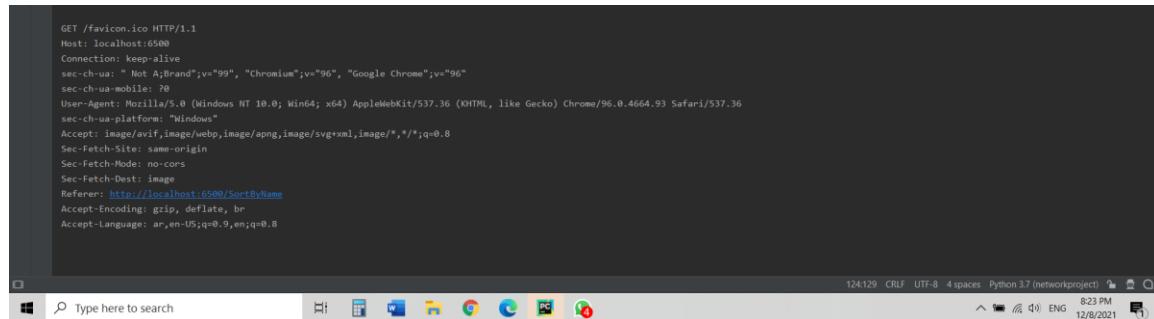
In the figure below we can see the http response for sort by name it accepted content type of text/plain, and the design of the page was arranged using html code and it was put in the main python. Note that the items were read from a text file in the python program.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help main.py [C:\Users\ALManar\Desktop\Pnetwrk] -...\\main.py - PyCharm
Project main
Run: main
C:\Users\ALManar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/ALManar/Desktop/Pnetwrk/main.py
(The server is ready to receive)
GET /SortByName HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
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/96.0.4664.93 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: ar,en-US;q=0.9,en;q=0.8

GET /back.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/SortByName
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 32: SortByName HTTP requests printed on command line - 1



```
GET /favicon.ico HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/SortByName
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 33: SortByName HTTP requests printed on command line - 2

Screenshot from another device (phone):

We used the following IP address to open the project:

192.168.1.1.158:6500/SortByPrice

192.168.1.1.158:6500/ SortByName

The image shows two side-by-side screenshots of a mobile browser on an Android device. Both screenshots display a table with two columns: 'The Item' and 'The Price'. The left screenshot is titled 'Sort By Price' and shows items ordered by price from lowest to highest. The right screenshot is titled 'Sort By Name' and shows items ordered alphabetically by name.

Sort By Price

The Item	The Price
Network book	15\$
Tp link router	40\$
Microsoft bag	50\$
Boss Perfume	100\$
Reebok shoe	100\$
Apple watch	250\$
Hp laptop	600\$
Samsung Galaxy S21 ultra	850\$
iPhone 12	900\$
MULLER Refrigerator	900\$
Dell laptop	1000\$
Skoda car	25000\$

Sort By Name

The Item	The Price
Apple watch	250\$
Boss Perfume	100\$
Dell laptop	1000\$
Hp laptop	600\$
iPhone 12	900\$
MULLER Refrigerator	900\$
Microsoft bag	50\$
Network book	15\$
Reebok shoe	100\$
Samsung Galaxy S21 ultra	850\$
Skoda car	25000\$
Tp link router	40\$

Figure 34: localhost:6500/SortByPrice and localhost:6500/SortByName browser from phone

3.6. Error 404

http://localhost:6500/AAAA

Main Page in the browser window:

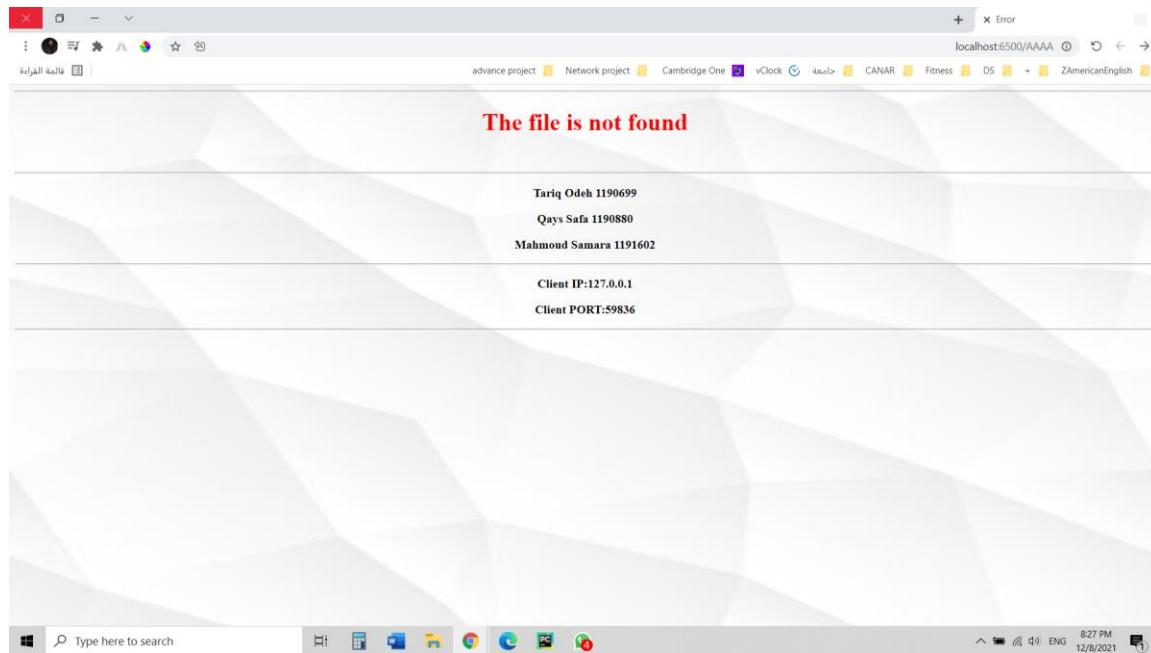


Figure 35: localhost:6500/AAAA browser window

Requests:

```
C:\Users\ALManar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/ALManar/Desktop/Pnetwrk/main.py
[Running] main.py
[Terminal]
Run: main.py
C:\Users\ALManar\Desktop\Pnetwrk>main.py
[The server is ready to receive]
GET /AAAA HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
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/96.0.4664.93 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;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-US;q=0.9,en;q=0.8

GET /back.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
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/96.0.4664.93 Safari/537.36
Accept: image/avif,image/webp,image/png,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/AAAA
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 36: AAAA HTTP requests printed on command line - 1

```
GET /favicon.ico HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Google Chrome";v="96"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 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:6500/AAAA
Accept-Encoding: gzip, deflate, br
Accept-Language: ar,en-US;q=0.9,en;q=0.8
```

Figure 37: AAAA HTTP requests printed on command line – 2

Screenshot from another device (phone):



Figure 38: localhost:6500/AAAA browser window from phone

3.7. Full Code with comments

```
# project done by:  
# Tariq Odeh (1190699)  
# Qays Safa (1190880)  
# Mahmoud Samara (1191602)  
  
from socket import *  
# Include Python's socket library.  
  
items = []  
# Initialise an array to put all items in it.  
PORT = 6500  
# Listening on port 6500.  
serverSocket = socket(AF_INET, SOCK_STREAM)  
# Create TCP socket for server, remote port 6500.  
serverSocket.bind(("", PORT))  
serverSocket.listen(1)  
# Server begins listening for incoming TCP requests.  
print("\t\t{The server is ready to receive}")  
  
# function to read the items file and cut it to items and prices  
  
def readfile(filename):  
  
    with open(filename) as f:  
# Create inputfile to read the data in items.txt .  
        item = f.readlines()  
# Read the data in Items.txt line.  
        for sentences in item:  
# Split the data from the file and append it to a new list, then cut the data  
based on :) .  
            line = sentences.split(":")  
            line[1] = str(line[1]).replace("\n", "")  
            line[1] = int(line[1])  
            items.append(line)  
# Add item in items array.  
readfile('items.txt')  
  
while True:  
  
    connectionSocket, address = serverSocket.accept()  
# Server waits on accept() for incoming requests, new socket created on return  
.  
    sentence = connectionSocket.recv(1024).decode()  
# Read bytes from socket.  
    requestFile = sentence.split(' ')[1]  
  
    printedfile = requestFile.lstrip('/')  
# Removing the first( / )to get the requested file name .  
    connectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())  
  
    if printedfile == '' or printedfile == 'index.html':  
        printedfile = 'main.html'  
# Load main.html file as default so if the request is / or /index.html then
```

```

the server should send main.html file.

try:

# Accepting different file formats
    if printedfile.endswith(".html"):
# If the request is a .html then the server should send the html file with
Content-Type: text/html.
        requestedType = 'text/html'

    elif printedfile.endswith(".css"):
# If the request is a .css then the server should send the css file with
Content-Type: text/css.
        requestedType = 'text/css'

    elif printedfile.endswith(".png"):
# If the request is a .jpg then the server should send the png image with
Content-Type: image/png.
        requestedType = 'image/png'

    elif printedfile.endswith(".jpg"):
# If the request is a .jpg then the server should send the jpg image with
Content-Type: image/jpg.
        requestedType = 'image/jpeg'

    elif printedfile == "SortByName" or printedfile == "SortByPrice":
# If the request is SortByName or SortByPrice then the server should send text
page with Content-Type: text/plain.
        requestedType = 'text/plain'

    else:
# Else then server should send text page with Content-Type: text/html.
        requestedType = 'text/html'

    if printedfile == 'SortByName' or printedfile == 'SortByPrice':

# If the user requests to sort either by name or by price for the items, it
will enter this IF condition

# to know to show the sort price page or sortname page depend on what the user
request.

        if printedfile == 'SortByName':
# To sort the items depending on the names.
            items.sort()
            ST = '<!DOCTYPE html><html><head><style>body {background-
image: url(' \
                           '"back.jpg");background-repeat: no-repeat;background-
attachment: fixed; background-size: ' \
                           '100% 100%;}</style></head><head><style>#Items {font-
family: Times new roman, ' \
                           'sans-serif;text-align:center;border-collapse:
collapse;width: 50%;} #Items td,' \
                           '#Items th {border: 5px solid #000000;padding: 8px;}'
#Items tr:nth-child(even){' \
                           'background-color: darkgrey;} #Items tr:hover
{background-color: darkgrey;}#Items th {' \
                           'padding-top: 12px;padding-bottom: 12px;text-align:
left;text-align:center;color: ' \

```

```

        'white;}></style></head><body><hr><center><h1>Sort By
Name</h1><table id="Items"><hr><tr ' \
        'style="background-color: royalblue;"><th>The
Item</th><th>The Price</th></tr> '


    else:

# To sort the items depending on the pricess.
        items.sort(key=lambda items: items[1])
        ST = '<!DOCTYPE html><html><head><style>body {background-
image: url(' \
            '"back.jpg");background-repeat: no-repeat;background-
attachment: fixed; background-size: ' \
            '100% 100%;}</style></head><head><style>#Items {font-
family: Times new roman, ' \
            'sans-serif;text-align:center;border-collapse:
collapse;width: 50%;} #Items td,' \
            '#Items th {border: 5px solid #000000;padding: 8px;}'
#Items tr:nth-child(even){' \
            'background-color: darkgrey;} #Items tr:hover
{background-color: darkgrey;}#Items th {' \
            'padding-top: 12px;padding-bottom: 12px;text-align:
left;text-align:center;color: ' \
            'white;}</style></head><body><hr><center><h1>Sort By
Price</h1><table id="Items"><hr><tr ' \
        'style="background-color: royalblue;"><th>The
Item</th><th>The Price</th></tr> '


        for OurItems in items:
# To fill the table with items.

            ST += '<td>' + OurItems[0] + '</td><td>' + str(OurItems[1]) +
'$</td></tr>'
            ST += "</table></center></body></html>"
# End of html code.

            printedfile = 'SortItems.html'
# Set the requested file name is SortItems.html.
            Sortfile = open("SortItems.html","w")
# Create SortItems.html to write the html code after added sorted item.
            Sortfile.write(ST)
            Sortfile.close()

            requestFile = open(printedfile,'rb') # Open and read the requested
file in byte format.
            ST = requestFile.read()
            requestFile.close()

            header = 'Content-Type: ' + str(requestedType) + '\r\n\r\n'

        except Exception as e:

# If the request is wrong or the file doesn't exist the server should return a
simple HTML webpage with our

# names and IDs and IP and port number of the client
            header = 'HTTP/1.1 404 Not Found\r\n\r\n'
            ST = ('<!DOCTYPE html><head><title>Error</title><style>
```

```
type="text/css">h1 {text-align: center;}li {font-weight: 'bold;}</style></head><head><style type="text/css">p {text-align: 'center;}</style></head><head><style> body {background-image: url("back.jpg"); 'background-repeat: no-repeat;background-attachment: fixed; background-size: 100% ' '100%;}</style></head><hr><body><h1 style="color:red">The file is not found</h1><br><hr><p ' 'style="color:black"><b>Tariq Odeh 1190699</b></p><p style="color:black"><b>Qays Safa ' '1190880</b></p><p style="color:black"><b>Mahmoud Samara 1191602</b></p><hr><p><b>Client ' 'IP:' +str(address[0])+ '</b></p><p><b>Client PORT:' +str(address[1])+ '</b></p><hr> '</body></html>').encode('utf-8')

connectionSocket.send(f"\r\n".encode())
connectionSocket.send(ST)
# Send the final response with all parts of header.
connectionSocket.close()
# To closes a connectionSocket socket.
print(sentence)
#print the HTTP request on the terminal window.
```

3.8. HTML Code

```
<!DOCTYPE html>
<html>

<head>
    <title>ENCS3320-Simple Webserver</title>
    <link rel="stylesheet" href="style.css" type="text/css">
</head>

<head>
    <style>
        body {
            background-image: url('back.jpg');
            background-repeat: no-repeat;
            background-attachment: fixed;
            background-size: 100% 100%;
        }
    </style>
</head>

<body>
    <div class="header">
        <h1>
            <ins> Welcome to our course <span style="color: #0070C0">Computer Networks </ins>
            </span>
        </h1>
    </div>

    
    <br><br>
```

```
<div class="informationBox">

<div>
    <h1>Tariq Odeh</h1>
    <h2>1190690</h2>
    

    <div align = "left">
        <h1 style="color: #0070C0" >Projects </h1>
        <ul>
            <li>An 8-bit Comparator for signed 2s complement representation numbers.</li>
            <li>An educational application that uses augmented reality technology.</li>
            <li>System to manage patients data in a hospital.</li>
        </ul>

        <h1 style="color: #0070C0" >Skills</h1>
        <ul>
            <li>quick mathematical and physical analysis.</li>
            <li>scientific research.</li>
            <li>Planing.</li>
        </ul>

        <h1 style="color: #0070C0" >Hobbies</h1>
        <ul>
            <li>Reading.</li>
            <li>Football.</li>
            <li>Squash.</li>
        </ul>
    </div>
</div>
</div>
```

```
<br><br><br><br>

<div class="informationBox">

    <div>
        <h1>Qays Safa</h1>
        <h2>1190880</h2>
        

        <div align = "left">
            <h1 style="color: #0070C0" >Projects </h1>
            <ul>
                <li>A Java program for managing patient information in a hospital.</li>
                <li>Establishing a company specializing in health food products.</li>
                <li>Making a simple calculator in 8086 program in Orga course.</li>
            </ul>

            <h1 style="color: #0070C0" >Skills</h1>
            <ul>
                <li>Marketing.</li>
                <li>Teamwork.</li>
                <li>Problem Solving.</li>
            </ul>

            <h1 style="color: #0070C0" >Hobbies</h1>
            <ul>
                <li>Programming.</li>
                <li>Volleyball</li>
                <li>cycling.</li>
            </ul>
        </div>
    </div>
```

```
</div>  
</div>  
  
<br><br><br><br>  
  
<div class="informationBox">  
    <div>  
        <h1>Mahmoud Samara</h1>  
        <h2>1191602</h2>  
        
```

```
    <div align = "left">  
        <h1 style="color: #0070C0" >Projects </h1>  
        <ul>
```

```
            <li>Making a simple calculator in 8086 program in Orga course.</li>  
            <li>A mathematical application that find the perfect number from group of  
numbers.</li>  
            <li>System to choose the shortest route from place to other using dijkstra  
program.</li>
```

```
        </ul>
```

```
<h1 style="color: #0070C0" >Skills</h1>
```

```
<ul>
```

```
    <li>Team work.</li>  
    <li>Java programming.</li>  
    <li>Good speaker.</li>
```

```
</ul>
```

```
<h1 style="color: #0070C0" >Hobbies</h1>
```

```
<ul>
```

```
<li>Football.</li>
<li>Swimming.</li>
<li>Programming.</li>
</ul>
</div>
</div>

<br><br><br><br>

<div align = "center">

<h3 style="color: #0070C0" >To go to the online HTML file press the following button </h3>
<form action="https://www.w3schools.com/tags/att_img_src.asp" target="_blank"
method="post">
    <input type="submit" value="Online HTML file">
</form>

<h3 style="color: #0070C0" >To go to the local HTML file press the following button </h3>
<form action="main.html" target="_blank" method="post">
    <input type="submit" value="Local HTML file">
</form>
</div>

</body>

</html>
```

3.9. CSS Code

```
.header}{  
    text-align: center;  
    height: 85px;  
    width: 100%;  
    padding: 10px 0 0 20px;  
    border: 0px;  
    border-radius: 0px;  
{
```

```
h1}{  
    color: black;  
{
```

```
li}{  
    font-size: 120%;  
{
```

```
.informationBox}{  
    text-align: center ;  
    height: 600px;  
    width: 100%;  
    border: 5px solid #0070C0;  
    border-radius: 10px;  
    display: inline-block ;  
{
```

```
.image}{  
    align : right;  
    height: 100%;  
}  
  
{
```

```
img}{  
    height: 400px;  
    position: relative;  
    top: 10px;  
    border-radius: 20px;  
    align : right;  
}  
  
{
```

```
.b-img}{  
    width: 50;  
    height: 50;  
    display: block;  
    margin-left: auto;  
    margin-right: auto ;  
}
```

4. References

- [1] CSS tutorial. (2019, April 8). Retrieved December 9, 2021, from <https://www.w3schools.com/css/>.
- [2] HTML tutorial. (2020, May 7). Retrieved December 9, 2021, from <https://www.w3schools.com/html/>.
- [3] *Python get current time*. Programiz. (2019, May 4). Retrieved December 9, 2021, from <https://www.programiz.com/python-programming/datetime/current-time>.
- [4] Team, P. (2019, May 7). *Send get request python socket - pretag*. Pretag development team. Retrieved December 9, 2021, from https://pretagteam.com/question/send-get-request-python-socket?fbclid=IwAR1P5xcT_GNiONtJiv2ak2SEeKMn7t1hkibXK0R-PfMtU59m_wbKlIR8IY.