



Faculty of Engineering & Technology  
Electrical & Computer Engineering Department

**ENCS3320**

**Project 1 Report**

**Socket Programming**

---

**Mohammed Buirat-1192896**

**Mohammad AbuJaber-1190298**

**Momen Bazzar-1192214**

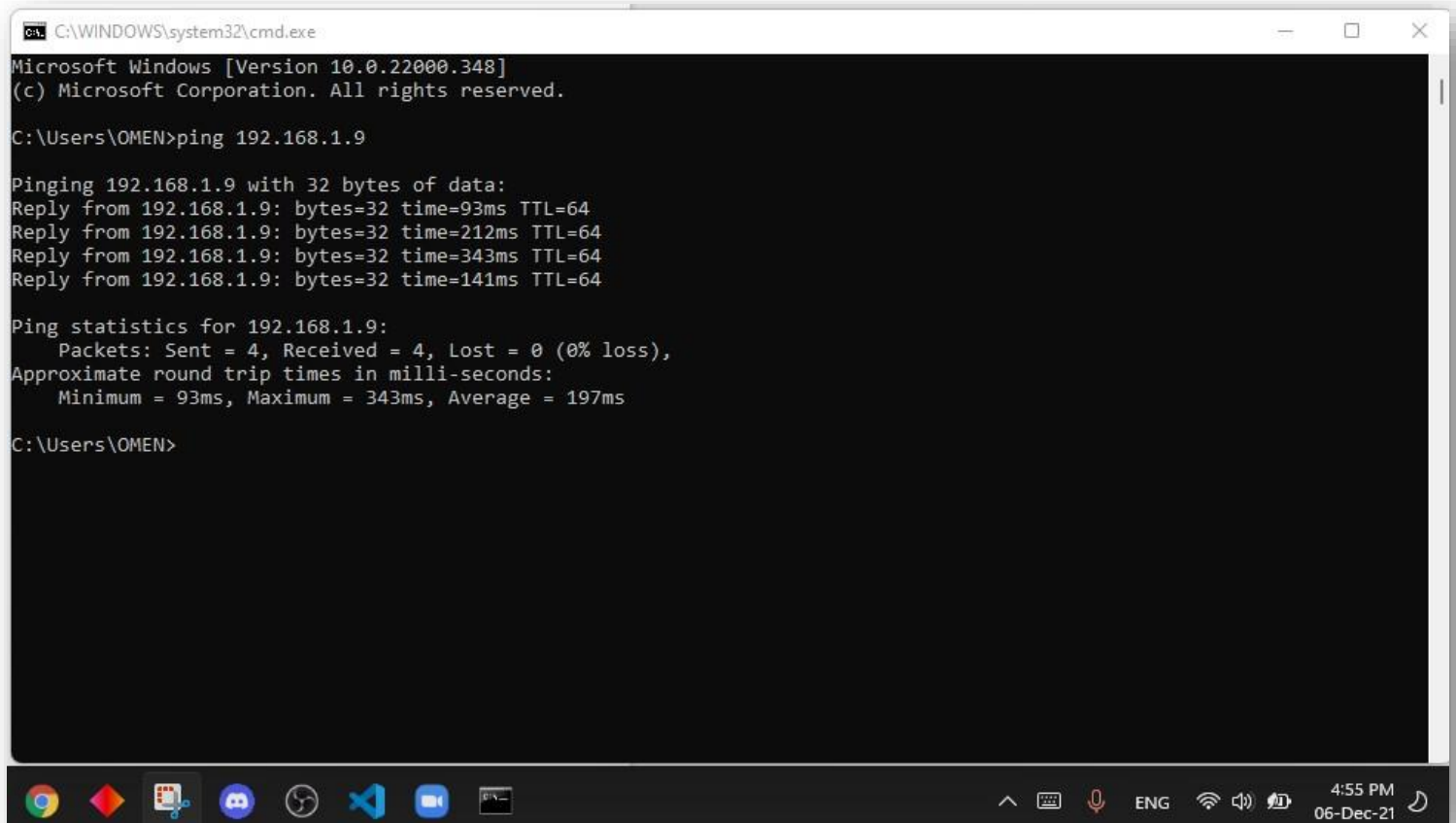
**10<sup>th</sup> Dec. 2021**

# Table of Contents

1. Part I:	3
1.1. Ping a device in the same network from a laptop to a smartphone	3
1.2. Ping www.birzeit.edu	4
1.3. Tracert www.stanford.edu	5
1.4. Nslookup www.cambridge.edu	6
2. Part II:	7
2.1. The code:	7
2.2. The output:	7
2.3. Code as text with comments:	8
3. Part III:	9
3.1. The code:	9
3.2. code as text whit comments:	10
3.3. The text file that we read the data from:	12
3.4. The HTTP response that we get when we open the browser and request localhost:6500	13
3.5. Screenshot from another computer: 192.168.1.109:6500/index.html	13
3.6. Localhost:6500 or Localhost:6500/index.html	14
3.7. Localhost:6500/x	16
3.8. /SortByName	17
3.9. /SortByPrice	17
3.10. /valid jpg request	18

## 1. Part I:

### 1.1. Ping a device in the same network from a laptop to a smartphone



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.22000.348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\OMEN>ping 192.168.1.9

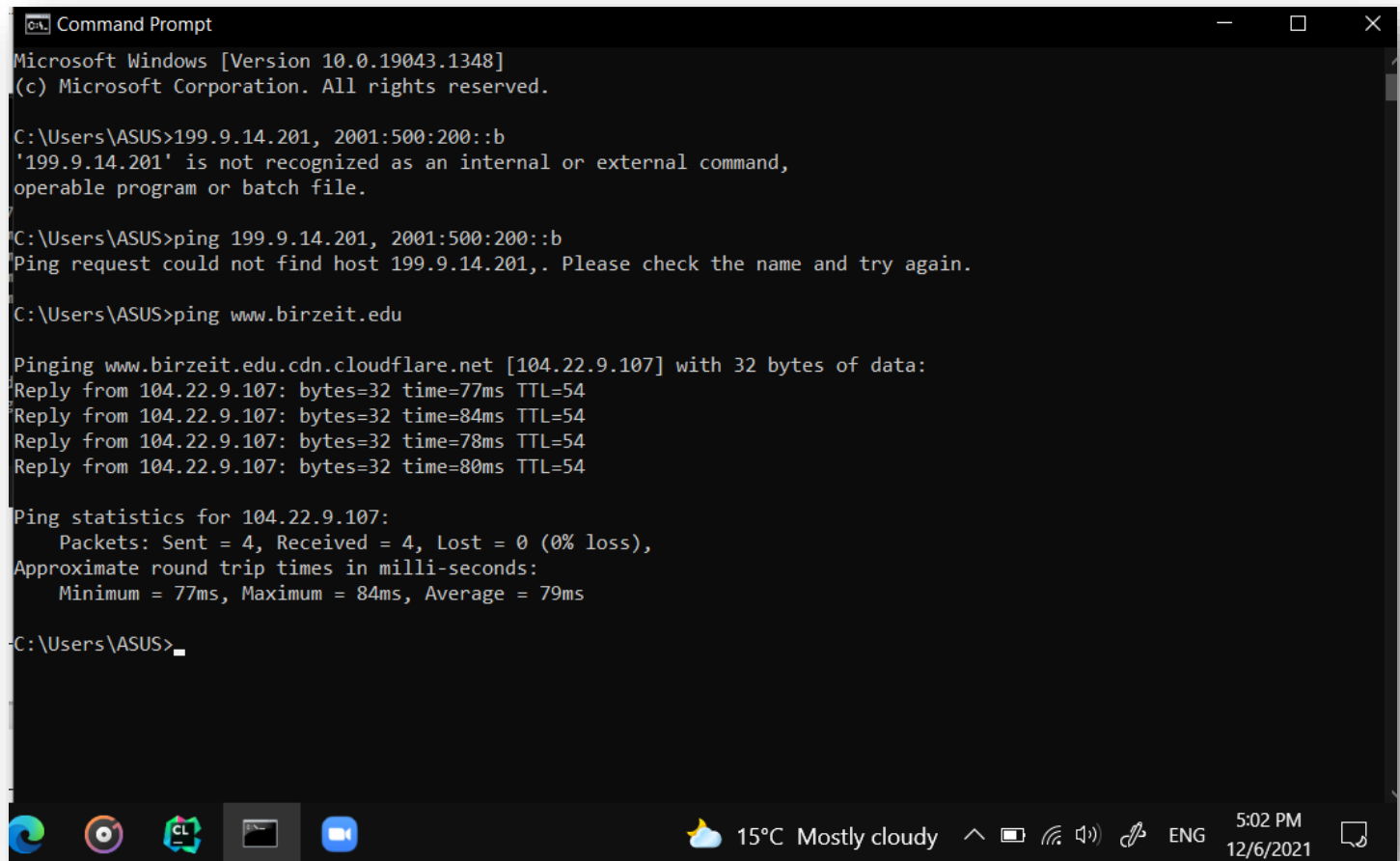
Pinging 192.168.1.9 with 32 bytes of data:
Reply from 192.168.1.9: bytes=32 time=93ms TTL=64
Reply from 192.168.1.9: bytes=32 time=212ms TTL=64
Reply from 192.168.1.9: bytes=32 time=343ms TTL=64
Reply from 192.168.1.9: bytes=32 time=141ms TTL=64

Ping statistics for 192.168.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 93ms, Maximum = 343ms, Average = 197ms

C:\Users\OMEN>
```

We can see from the previous message that we received a response from **192.168.1.9** Where we sent 4 packets all of them have the same time to live (**TTL**) around 64 ms, with different delays where the avg is 197 ms.

## 1.2. Ping [www.birzeit.edu](http://www.birzeit.edu)



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>199.9.14.201, 2001:500:200::b
'199.9.14.201' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\ASUS>ping 199.9.14.201, 2001:500:200::b
Ping request could not find host 199.9.14.201,. Please check the name and try again.

C:\Users\ASUS>ping www.birzeit.edu

Pinging www.birzeit.edu.cdn.cloudflare.net [104.22.9.107] with 32 bytes of data:
Reply from 104.22.9.107: bytes=32 time=77ms TTL=54
Reply from 104.22.9.107: bytes=32 time=84ms TTL=54
Reply from 104.22.9.107: bytes=32 time=78ms TTL=54
Reply from 104.22.9.107: bytes=32 time=80ms TTL=54

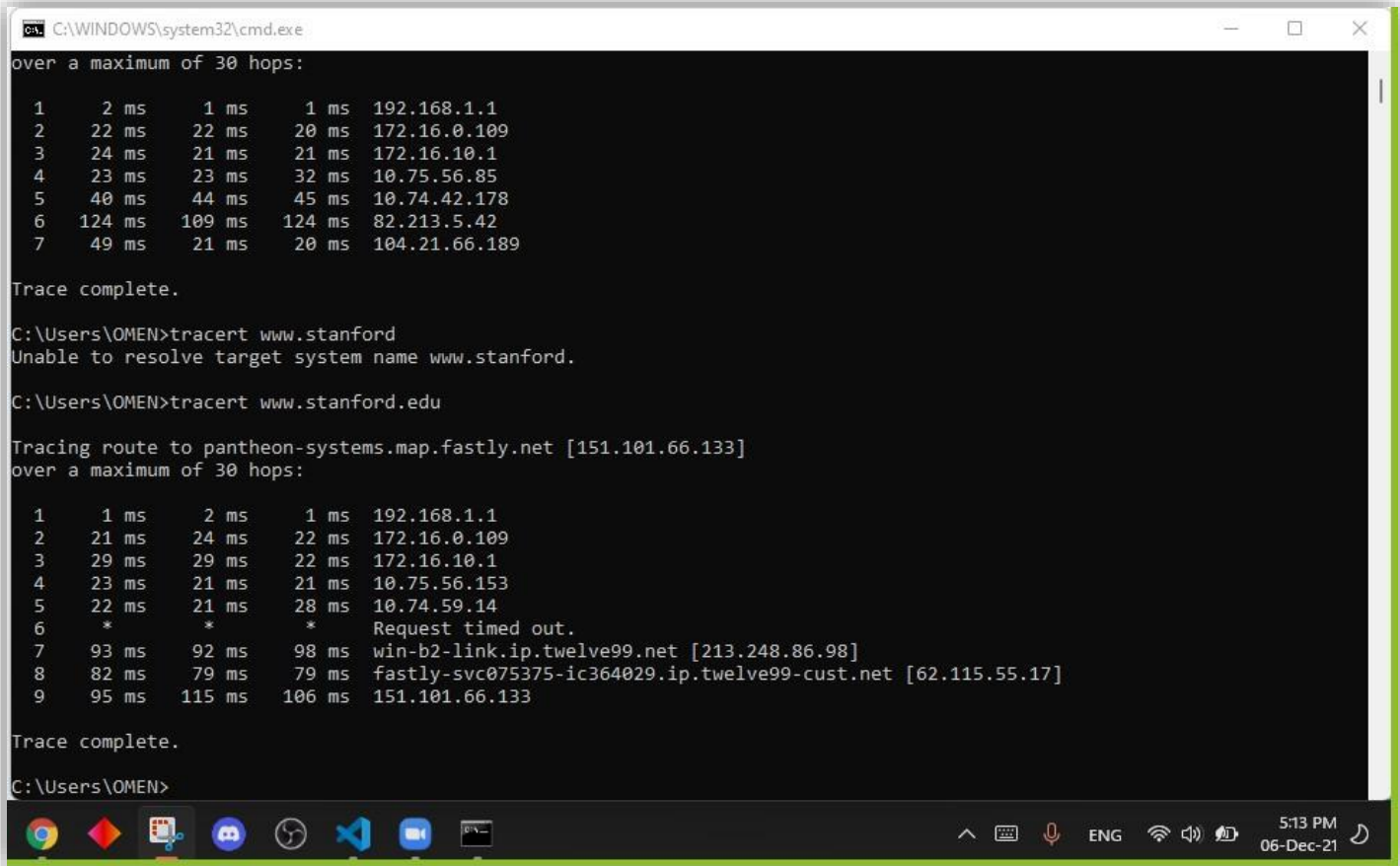
Ping statistics for 104.22.9.107:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 77ms, Maximum = 84ms, Average = 79ms

C:\Users\ASUS>
```

We can see in the previous image that we received a response from **104.22.9.107**

We sent 4 packets that all have the same time to live and we received them with different delays, with an average 79ms.

### 1.3. Tracert www.stanford.edu



```
C:\WINDOWS\system32\cmd.exe
over a maximum of 30 hops:

 1    2 ms    1 ms    1 ms  192.168.1.1
 2   22 ms   22 ms   20 ms  172.16.0.109
 3   24 ms   21 ms   21 ms  172.16.10.1
 4   23 ms   23 ms   32 ms  10.75.56.85
 5   40 ms   44 ms   45 ms  10.74.42.178
 6  124 ms  109 ms  124 ms  82.213.5.42
 7   49 ms   21 ms   20 ms  104.21.66.189

Trace complete.

C:\Users\OMEN>tracert www.stanford
Unable to resolve target system name www.stanford.

C:\Users\OMEN>tracert www.stanford.edu

Tracing route to pantheon-systems.map.fastly.net [151.101.66.133]
over a maximum of 30 hops:

 1    1 ms    2 ms    1 ms  192.168.1.1
 2   21 ms   24 ms   22 ms  172.16.0.109
 3   29 ms   29 ms   22 ms  172.16.10.1
 4   23 ms   21 ms   21 ms  10.75.56.153
 5   22 ms   21 ms   28 ms  10.74.59.14
 6    *      *      *      Request timed out.
 7   93 ms   92 ms   98 ms  win-b2-link.ip.twelve99.net [213.248.86.98]
 8   82 ms   79 ms   79 ms  fastly-svc075375-ic364029.ip.twelve99-cust.net [62.115.55.17]
 9   95 ms  115 ms  106 ms  151.101.66.133

Trace complete.

C:\Users\OMEN>
```

In the previous figure, the measurement increases since the router will have to go further which will take more time, while the star symbol (\*) represents that there is time out and the packet did not reach the destination since there is a problem in the location or the route is incorrect.

## 1.4. Nslookup [www.cambridge.edu](http://www.cambridge.edu)

```
Command Prompt
over a maximum of 30 hops:
  1    1 ms    1 ms    1 ms    192.168.1.1
  2   25 ms   22 ms   23 ms   241net178-214-76.gemzo.net [178.214.76.241]
  3    *      *      *      Request timed out.
  4   31 ms   32 ms   31 ms   bzq-82-81-95-153.red.bezeqint.net [82.81.95.153]
  5   26 ms   27 ms   27 ms   10.190.5.2
  6   80 ms   83 ms   81 ms   bzq-219-189-14.dsl.bezeqint.net [62.219.189.14]
  7   87 ms  131 ms  106 ms  de-cix-frankfurt.as13335.net [80.81.194.180]
  8   78 ms   77 ms   80 ms  104.22.9.107

Trace complete.

C:\Users\ASUS>
C:\Users\ASUS>tracert https://www.ox.ac.uk/
Unable to resolve target system name https://www.ox.ac.uk/.

C:\Users\ASUS>nslookup cambridge.edu
Server: UnKnown
Address: 192.168.1.1

Non-authoritative answer:
Name:   cambridge.edu
Addresses:  2606:4700:3031::6815:42bd
            2606:4700:3034::ac43:a377
            172.67.163.119
            104.21.66.189

C:\Users\ASUS>
```

Will show the IP address of the device corresponding to the host which is the device that I am working on, and print the name and address of the host to which we sent the prop.

## 2. Part II:

### 2.1. The code:

```
import datetime
# to import the library datetime which will help in calculating time
# taken from sending the request until we receive the response
import requests # to import the library requests

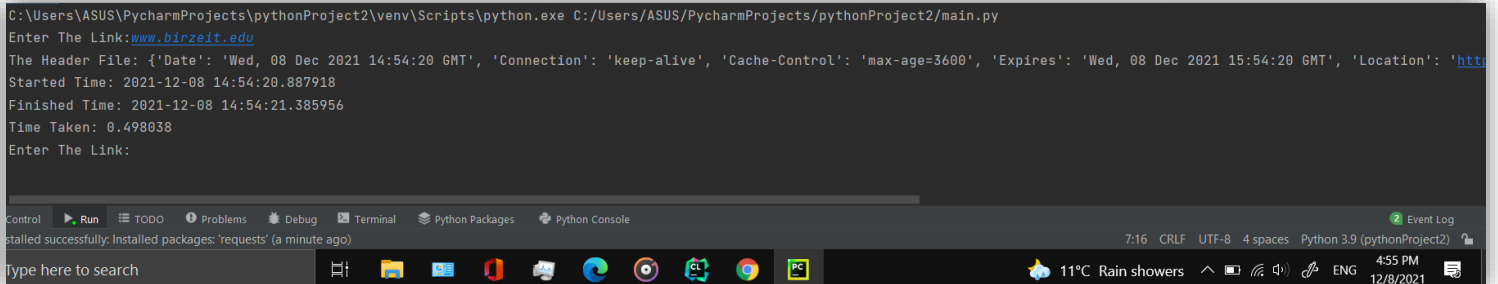
while True: # infinite loop
    url = input("Enter The Link: <enter -1 to exit>\n") # to print a statement to help the user know what should he do
    if url == "-1": # end the loop when the user enters -1
        break
    if not url.startswith('http'): # this will check if the entered link starts with http or not
        # if it did not start with http, it will add <http://> at the beginning of the link to avoid problems
        url = 'http://' + url
        # for example==>      http://amazon.com/ or amazon.com

    dt_started = datetime.datetime.utcnow() # started time will be the time we send the request at

    url = requests.head(url)
    # using head method to request the headers that would be returned if
    # the head request's url was instead requested with the http get method

    dt_ended = datetime.datetime.utcnow() # ended time will be the time receive the response at
    # We used <print(f"{}")> to print statements and variables values at the same line with the same print <statement>
    print(f"The Header File: {url.headers}")
    print(f"Started Time: {dt_started}")
    print(f"Finished Time: {dt_ended}")
    print(f"Time Taken: {(dt_ended - dt_started).total_seconds()}")
```

### 2.2. The output:



```
C:\Users\ASUS\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/ASUS/PycharmProjects/pythonProject2/main.py
Enter The Link:www.birzeit.edu
The Header File: {'Date': 'Wed, 08 Dec 2021 14:54:20 GMT', 'Connection': 'keep-alive', 'Cache-Control': 'max-age=3600', 'Expires': 'Wed, 08 Dec 2021 15:54:20 GMT', 'Location': 'http://www.birzeit.edu'}
Started Time: 2021-12-08 14:54:20.887918
Finished Time: 2021-12-08 14:54:21.385956
Time Taken: 0.498038
Enter The Link:
```

Showing the time of request, time of response the time interval, and the ability to enter new link (the program will exit when we enter -1).

## 2.3. Code as text with comments:

```
import requests #to import the library requests
import datetime #to import the library datetime which will help in calculating
time taken from sending the request until we receive the response

while True: #infinite loop

    url = input("Enter The Link: <enter -1 to exit>\n") #to print a statment
to help the user know what should he do
    if url == "-1": #end the loop when the user enters -1
        break
    if not url.startswith('http'): #this will check if the entered link starts
with http or not
        url = 'http://' + url #if it did not start with http, it will add
<http://> at the beginning of the link to avoid problems
        #for example:==> http://amazon.com/ or amazon.com

    dt_started = datetime.datetime.utcnow() #started time will be the time
we send the request at

    url = requests.head(url) #using head method to request the headers that
would be returned if the head request's url was instead requested with the
http get method

    dt_ended = datetime.datetime.utcnow() #ended time will be the time receive
the response at
    #We used <print(f"{}")> to print statements and variables values at the
same line with the same print <statement>
    print(f"The Header File: {url.headers}")
    print(f"Started Time: {dt_started}")
    print(f"Finished Time: {dt_ended}")
    print(f"Time Taken: {(dt_ended - dt_started).t
```



## 3. Part III:

### 3.1. The code:

```
1 import socket
2
3 PORT = 6500
4 SERVER = '127.0.0.1'
5 ADDR = (SERVER, PORT)
6 FORMAT = 'utf-8'
7
8 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9 server.bind(ADDR)
10
11
12 def start():
13     server.listen()
14     print(f"[LISTENING] Server is listening on {SERVER}")
15     while True:
16         conn, addr = server.accept()
17         msg = conn.recv(2048).decode(FORMAT)
18         print(msg)
19         ip = addr[0]
20         port = addr[1]
21         string_list = msg.split(' ') # Split request from spaces
22         requestFile = string_list[1]
23         conn.send(f"HTTP/1.1 200 OK\r\n".encode())
24         myfile = requestFile.split('?')[0] # After the "?" symbol not relevant here
25         myfile = myfile.lstrip('/')
26         try:
27             if myfile == '':
28                 myfile = 'index.html'
29             elif myfile.lower() == 'sortbyname' or myfile.lower() == 'sortbyprice':
30                 old_myfile = myfile
31                 myfile = 'items.txt'
32             else:
33                 pass
```

```
ODO Problems Debug Terminal Python Packages Python Console Event Log
Installed packages: 'requests' (4 minutes ago) 72:1 CRLF UTF-8 4 spaces Python 3.9 (pythonProject2) 4:57 PM 12/8/2021
29 elif myfile.lower() == 'sortbyname' or myfile.lower() == 'sortbyprice':
30     old_myfile = myfile
31     myfile = 'items.txt'
32     with open('input.txt', 'r+') as file, open(myfile, 'w') as outfile:
33         arr = []
34         for line in file:
35             name, price = line.replace('\n', '').split(';;')
36             arr.append([name, price])
37         if old_myfile.lower() == 'sortbyname':
38             arr.sort(key=lambda x: str(x[0]))
39         else:
40             arr.sort(key=lambda x: int(x[1]))
41         for data in arr:
42             outfile.write(f'{data[0]};;{data[1]}\n')
43
44     requestFile = open(myfile, 'rb')
45     response = requestFile.read()
46     requestFile.close()
47     if myfile.endswith(".jpg"):
48         conn.send(f"Content-Type: image/jpeg \r\n".encode(FORMAT))
49     elif myfile.endswith(".png"):
50         conn.send(f"Content-Type: image/png \r\n".encode(FORMAT))
51     elif myfile.endswith(".css"):
52         conn.send(f"Content-Type: text/css \r\n".encode(FORMAT))
53     elif myfile.endswith(".txt"):
54         conn.send(f"Content-Type: text/plain \r\n".encode(FORMAT))
55     else:
56         conn.send(f"Content-Type: text/html \r\n".encode(FORMAT))
57
58 except FileNotFoundError:
```

```
h TODO Problems Debug Terminal Python Packages Python Console Event Log
Installed packages: 'requests' (4 minutes ago) 72:1 CRLF UTF-8 4 spaces Python 3.9 (pythonProject2) 4:58 PM 12/8/2021
```

```

57
58     except FileNotFoundError:
59         conn.send(f"Content-Type: text/html \r\n".encode(FORMAT))
60         response = ('<html><title>Error</title><body><center><h1 style="color:red">The file is not found </h1> <hr> <p style=
61             '"font-weight: bold;"> Momen Bazzar - 1192214 </p> <p style="font-weight: bold;"> '
62             'Mohammad Buirat - 1192896 </p> <p style="font-weight: bold;"> Mohammad Abu Jaber - 1190298 '
63             '</p> <hr> <h2> IP: ' + str(ip) + ', Port: ' + str(port)
64             + '</h2></center></body></html>').encode(FORMAT)
65         conn.send(f"\r\n".encode())
66         conn.send(response)
67         conn.close()
68
69
70     print("[STARTING] server is starting...")
71     start()
72

```

### 3.2. code as text whit comments:

```
import socket #To include Python's socket library
```

```
PORT = 6500 #Declaring unused port
SERVER = '127.0.0.1' #Giving our server an ip address
ADDR = (SERVER, PORT)
FORMAT = 'utf-8'
```

```
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #Create a
welcoming socket
server.bind(ADDR)
```

```
def start(): #Declaring the function start
    server.listen() #To listen and wait for incoming requests
    print(f"[LISTENING] Server is listening on {SERVER}") #Printing on
terminal
```

```
while True:
    conn, addr = server.accept()
    msg = conn.recv(2048).decode(FORMAT)
    print(msg)
    ip = addr[0]
    port = addr[1]
    string_list = msg.split(' ') # Split request from spaces
    requestFile = string_list[1]
    conn.send(f"HTTP/1.1 200 OK\r\n".encode())
    myfile = requestFile.split('?')[0] # After the "?" symbol not
relevent here
    myfile = myfile.lstrip('/')

```

```

try:
    if myfile == '': #If nothing was send with the request, the
default      is      the      main      html      file
        myfile = 'index.html'
    elif myfile.lower() == 'sortbyname' or myfile.lower() ==
'sortbyprice': #It will convert the request into lower case
        old_myfile = myfile
        myfile = 'items.txt'
    with open('input.txt', 'r+') as file, open(myfile, 'w') as
outfile: #It will deal with input file as an input file, and items file as
an      output      file
        arr = []
        for line in file:
            name, price = line.replace('\n', '').split(';;') #It
will put the first information before ;; inside the <name> variable and the
data      after      ;;      will      be      the      price
            arr.append([name, price]) #To create array of names
and      prices
        if old_myfile.lower() == 'sortbyname':
            arr.sort(key=lambda x: str(x[0])) #We used lambda
function which helps us compare based on the names located at index 0 of the
columns      <before      >;;>
        else:
            arr.sort(key=lambda x: int(x[1])) #Using lambda
function to compare based on the prices located at index 1 of the columns
<after      >;;>
        for data in arr: #For loop to print the sorted data into
a      new      file      <items.txt>
            outfile.write(f'{data[0]};;{data[1]}\n')

```

```

requestFile = open(myfile, 'rb')
response = requestFile.read()
requestFile.close()
#To check the requested order and send the appropriate data
if myfile.endswith(".jpg"):
    conn.send(f"Content-Type: image/jpeg \r\n".encode(FORMAT))
elif myfile.endswith(".png"):
    conn.send(f"Content-Type: image/png \r\n".encode(FORMAT))
elif myfile.endswith(".css"):
    conn.send(f"Content-Type: text/css \r\n".encode(FORMAT))
elif myfile.endswith(".txt"):
    conn.send(f"Content-Type: text/plain \r\n".encode(FORMAT))
else:
    conn.send(f"Content-Type: text/html \r\n".encode(FORMAT))

```

```

except FileNotFoundError: #The exception that will be sent if the

```

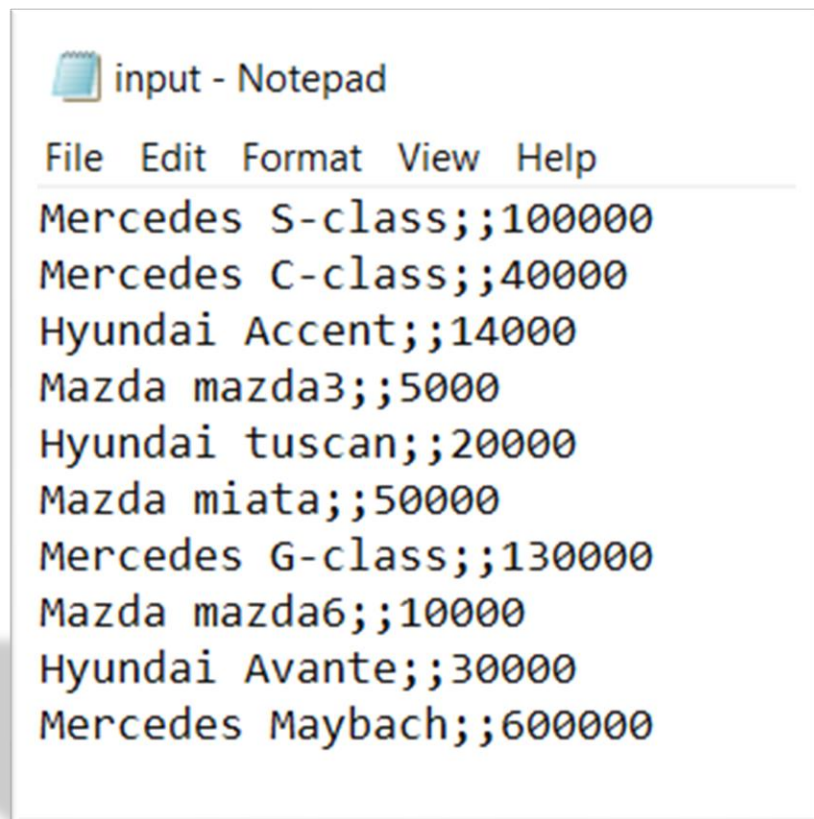
```

requested order was not found
conn.send(f"Content-Type: text/html \r\n".encode(FORMAT))
#HTML code for a simple page with the 404 error
response = ('<html><title>Error 404</title><body><center><h1
style="color:red">The file is not found </h1> <hr> <p style= '
'"font-weight: bold;"> Momen Bazzar - 1192214 </p>
<p style="font-weight: bold;">
'Mohammad Buirat - 1192896 </p> <p style= "font-
weight: bold;"> Mohammad AbuJaber - 1190298 '
'</p> <hr> <h2> IP: ' + str(ip) + ', Port: ' +
str(port)
+ '</h2></center></body></html>').encode(FORMAT)
conn.send(f"\r\n".encode())
conn.send(response)
conn.close()

print("[STARTING] server is starting...")
start()

```

3.3. The text file that we read the data from:

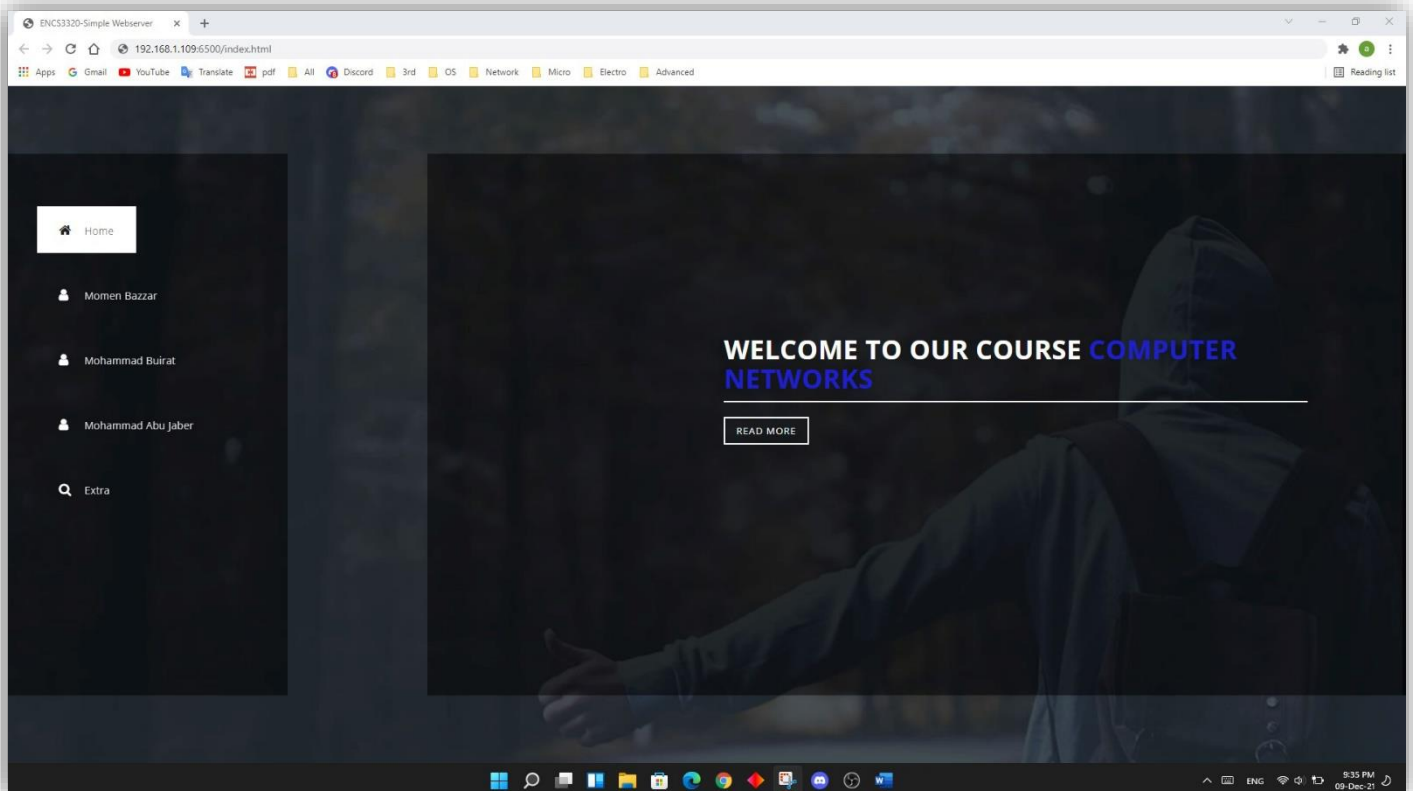


### 3.4. The HTTP response that we get when we open the browser and request localhost:6500

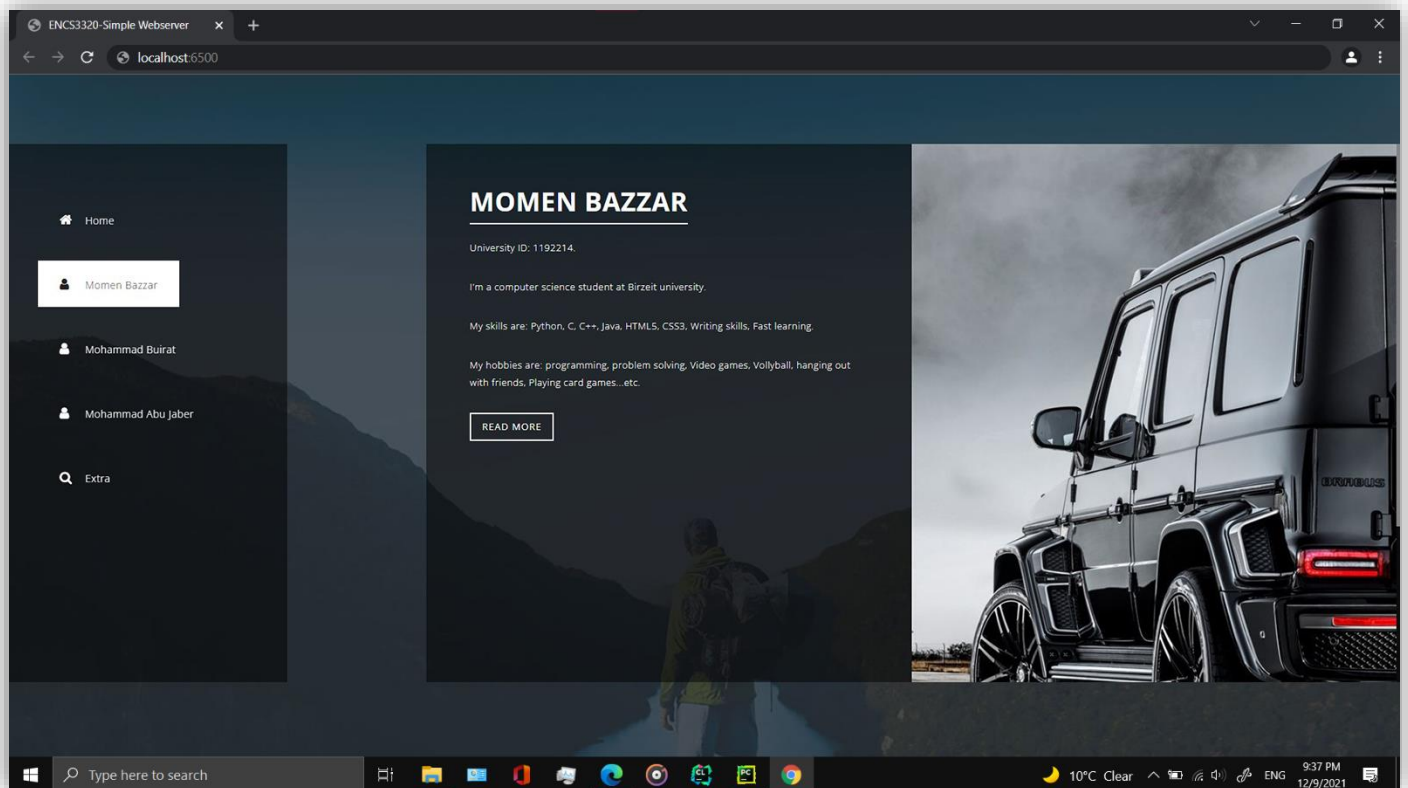
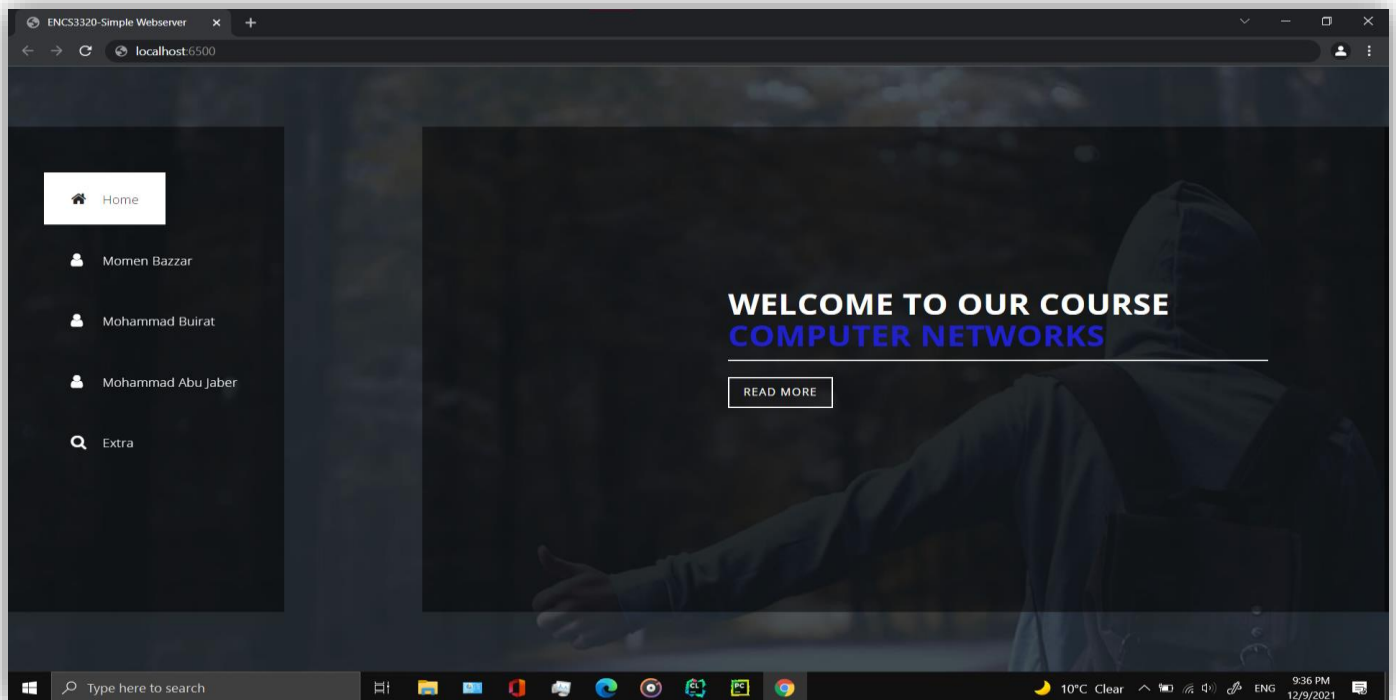
```
C:\Users\ASUS\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/ASUS/PycharmProjects/pythonProject2/main.py
[STARTING] server is starting...
[LISTENING] Server is listening on 127.0.0.1
GET / HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "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/96.0.4664.45 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

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/96.0.4664.45 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
```

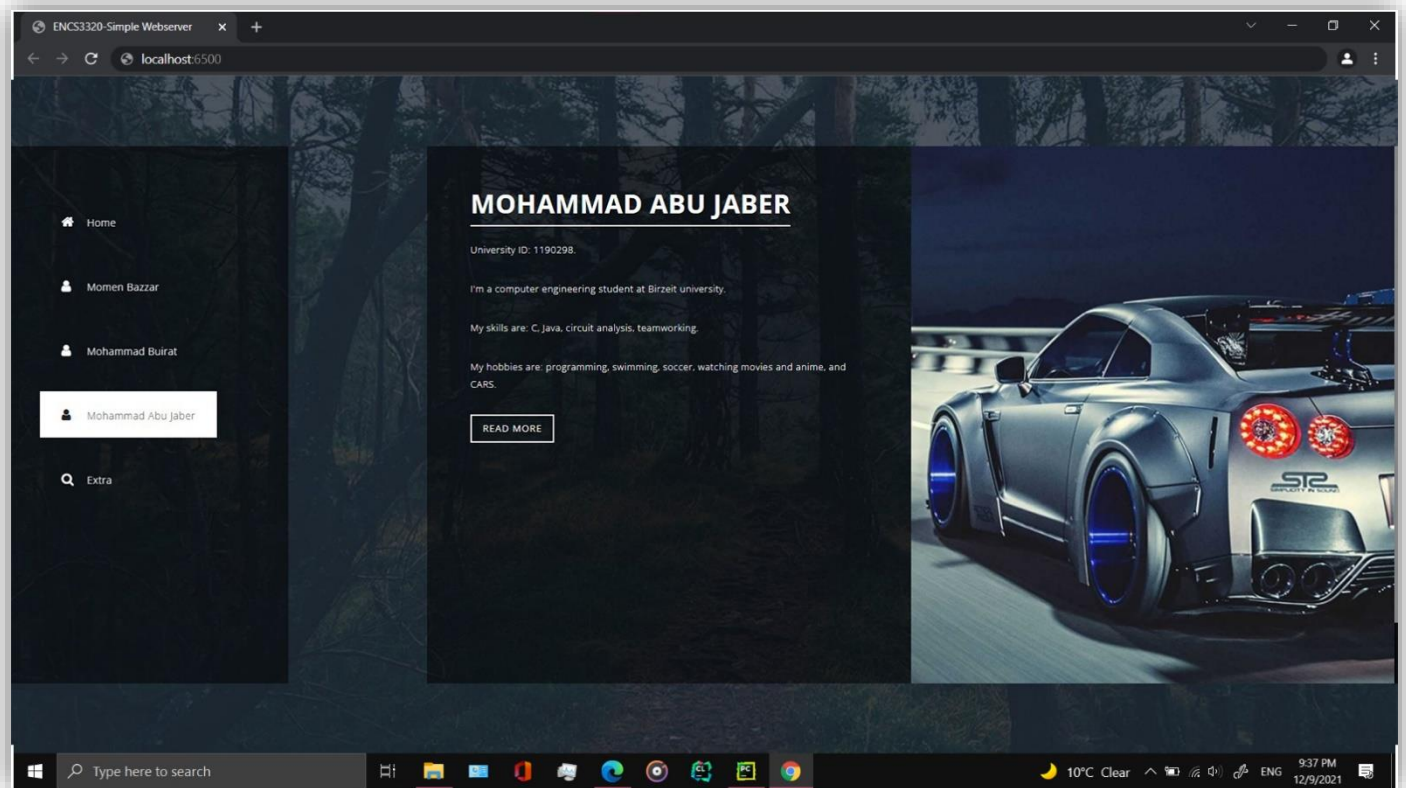
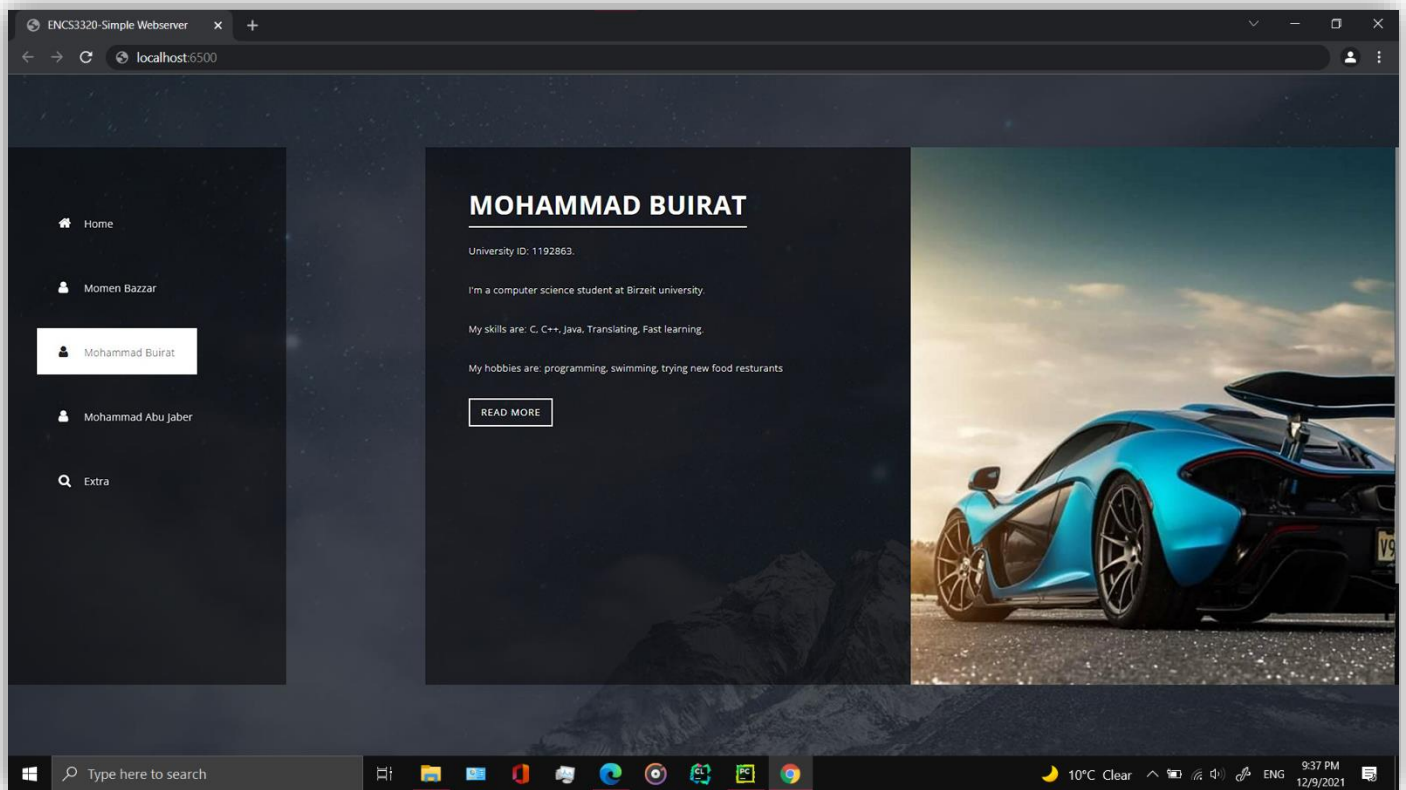
### 3.5. Screenshot from another computer: 192.168.1.109:6500/index.html



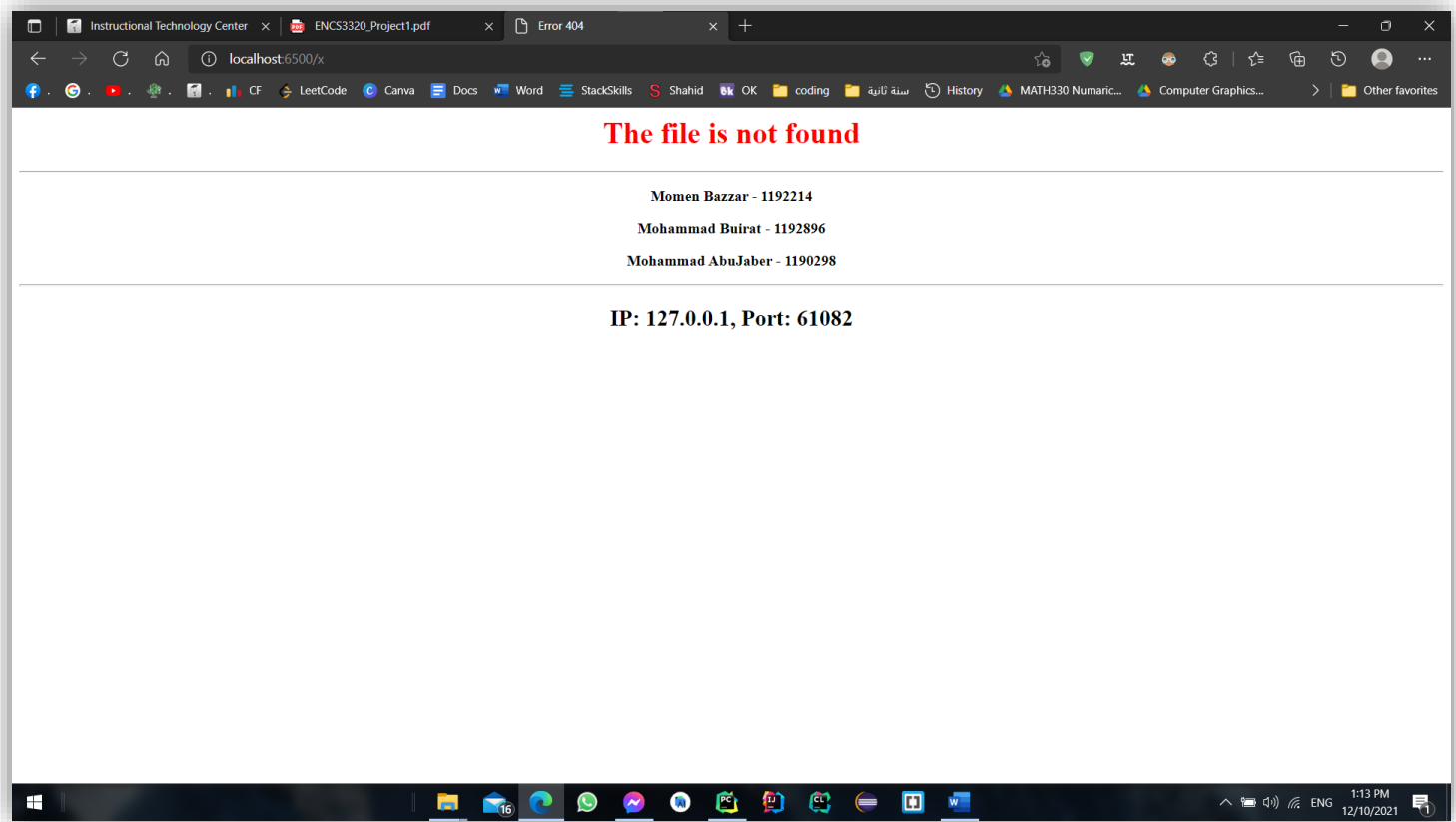
### 3.6. Localhost:6500 or Localhost:6500/index.html







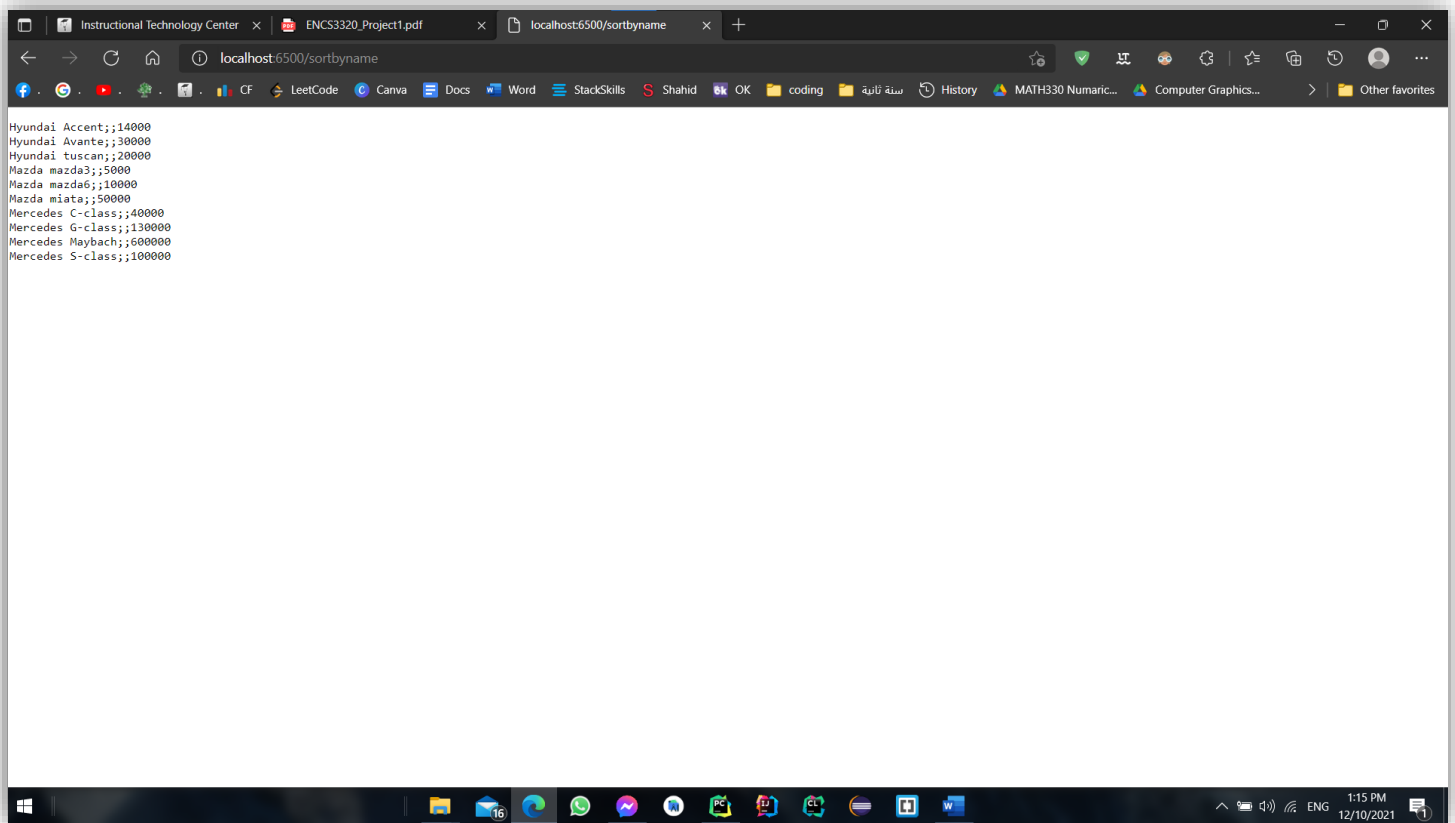
### 3.7. Localhost:6500/x



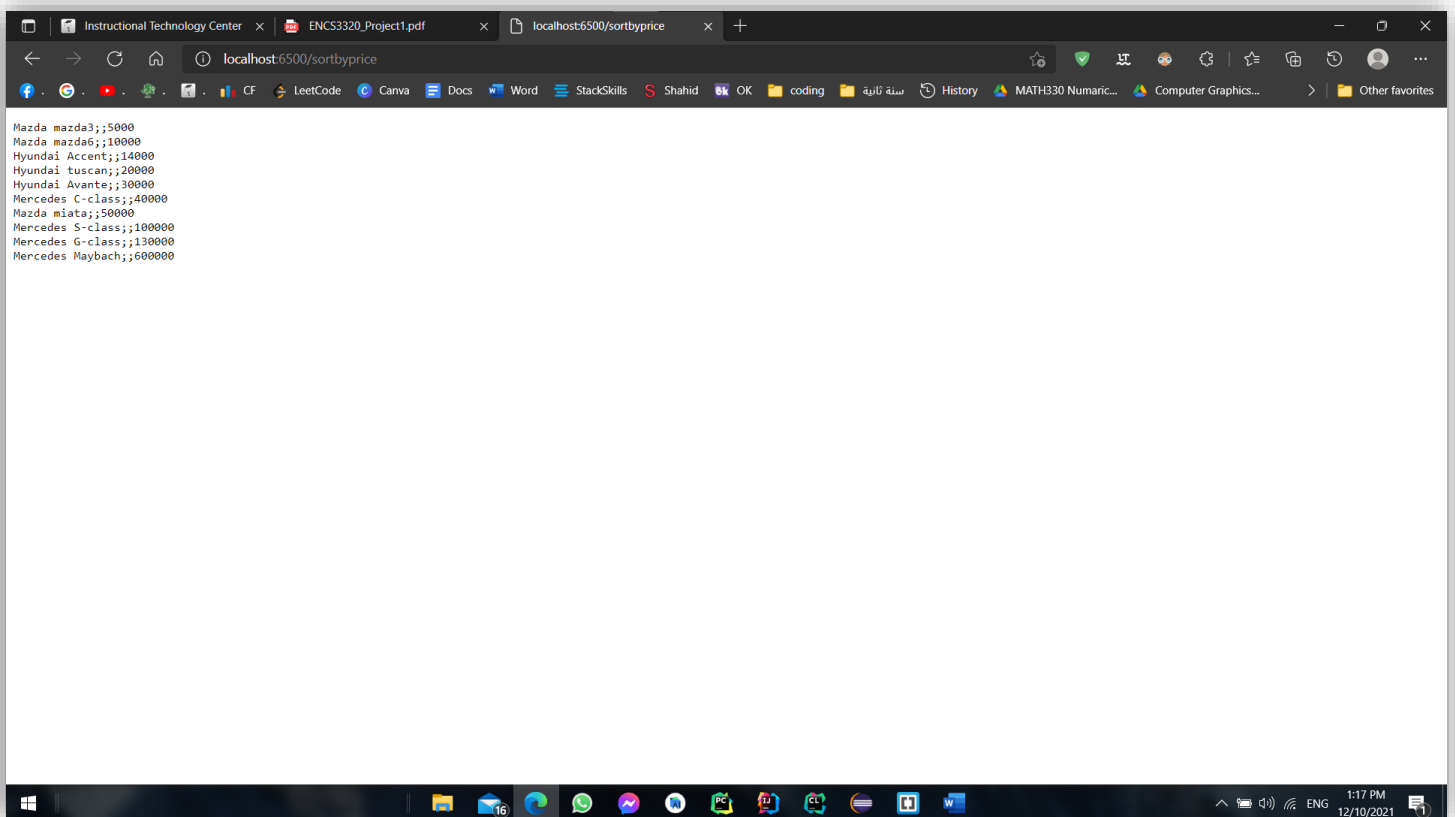
/X return the error page because it's not a valid request or file



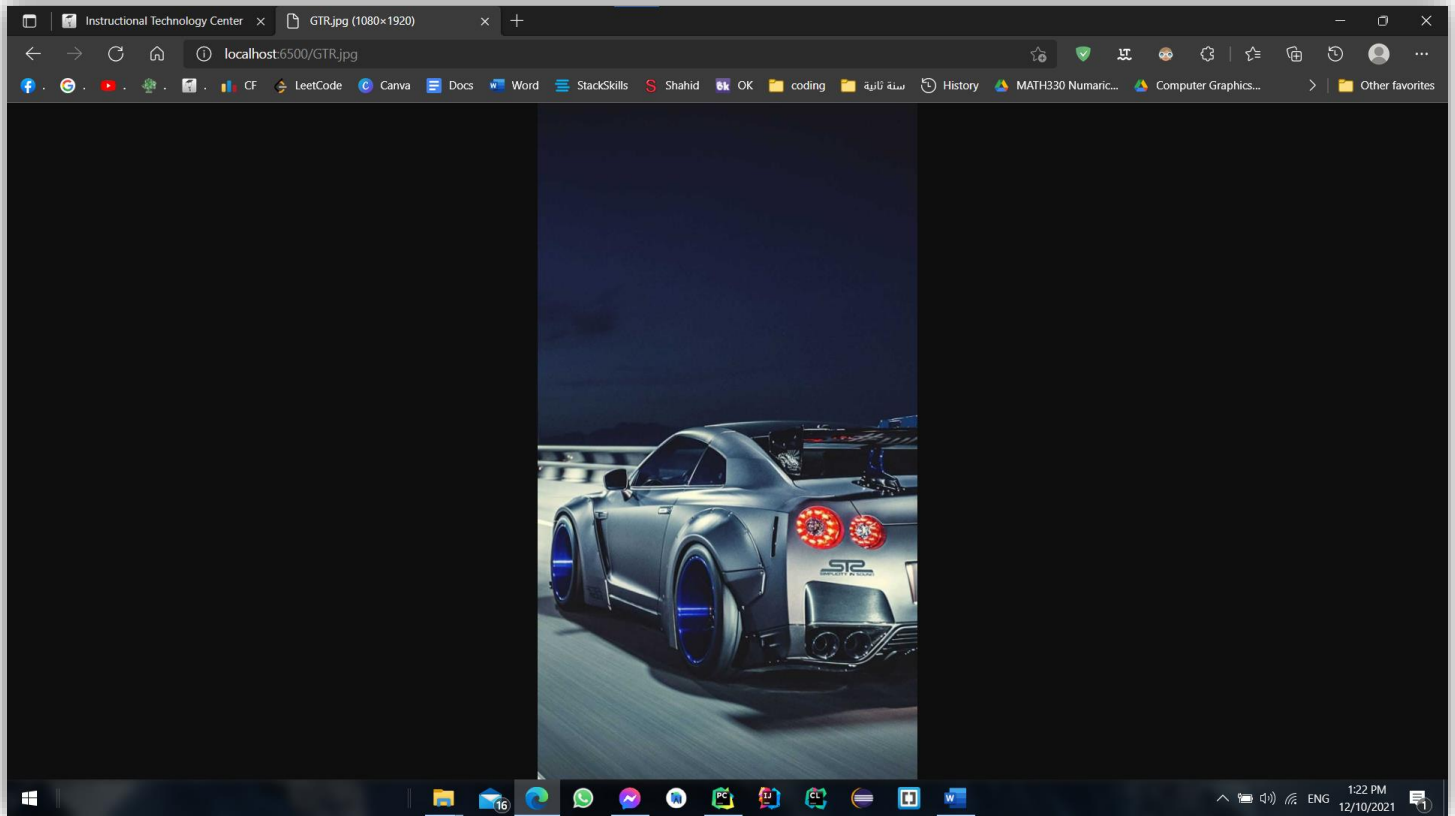
### 3.8. /SortByName



### 3.9. /SortByPrice



### 3.10. /valid.jpg request



```
GET /GTR.jpg HTTP/1.1
Host: localhost:6500
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96", "Microsoft Edge";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/96.0.4664.55 Safari/537.36 Edg/96.0.1054.43
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ar;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", "Microsoft Edge";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/96.0.4664.55 Safari/537.36 Edg/96.0.1054.43
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:6500/GTR.jpg
```

TODO Problems Terminal Python Packages Python Console  
Completed in 48 sec. Shared indexes were applied to 685 of 5,187 files (13%) (today 12:15 PM)

78.8 4 spaces Python 3.9 (Project1) Event Log  
1:23 PM 12/10/2021