BIRZEIT UNIVERSITY

Faculty of Engineering & Technology
Electrical & Computer Engineering Department

ENCS3320

Project 1 Report

Socket Programming

Mohammed Buirat-1192896

Mohammad AbuJaber-1190298

Momen Bazzar-1192214

10th **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

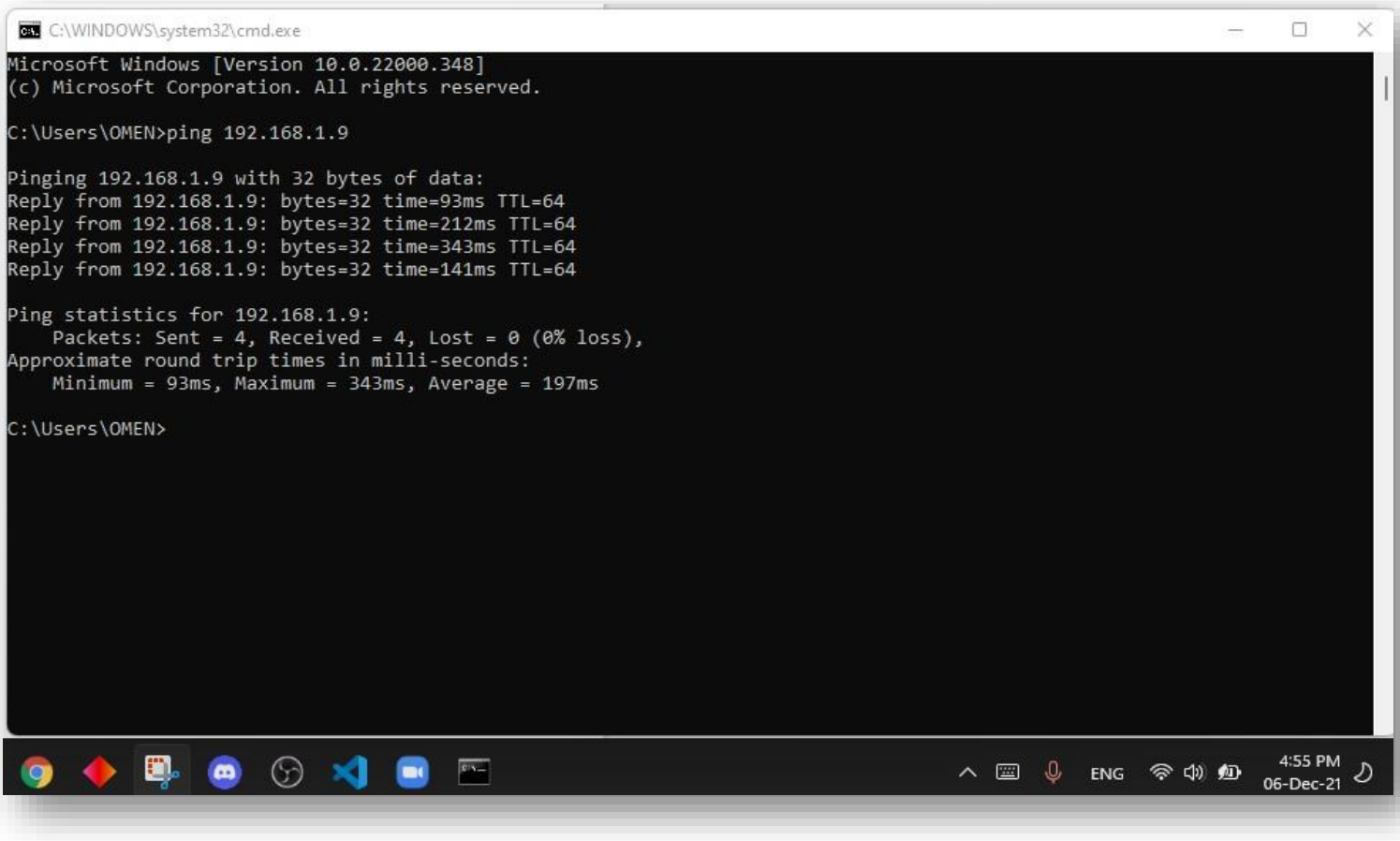
3.8. /SortByName 17

3.9. /SortByPrice 17

3.10. /valid jpg request 18

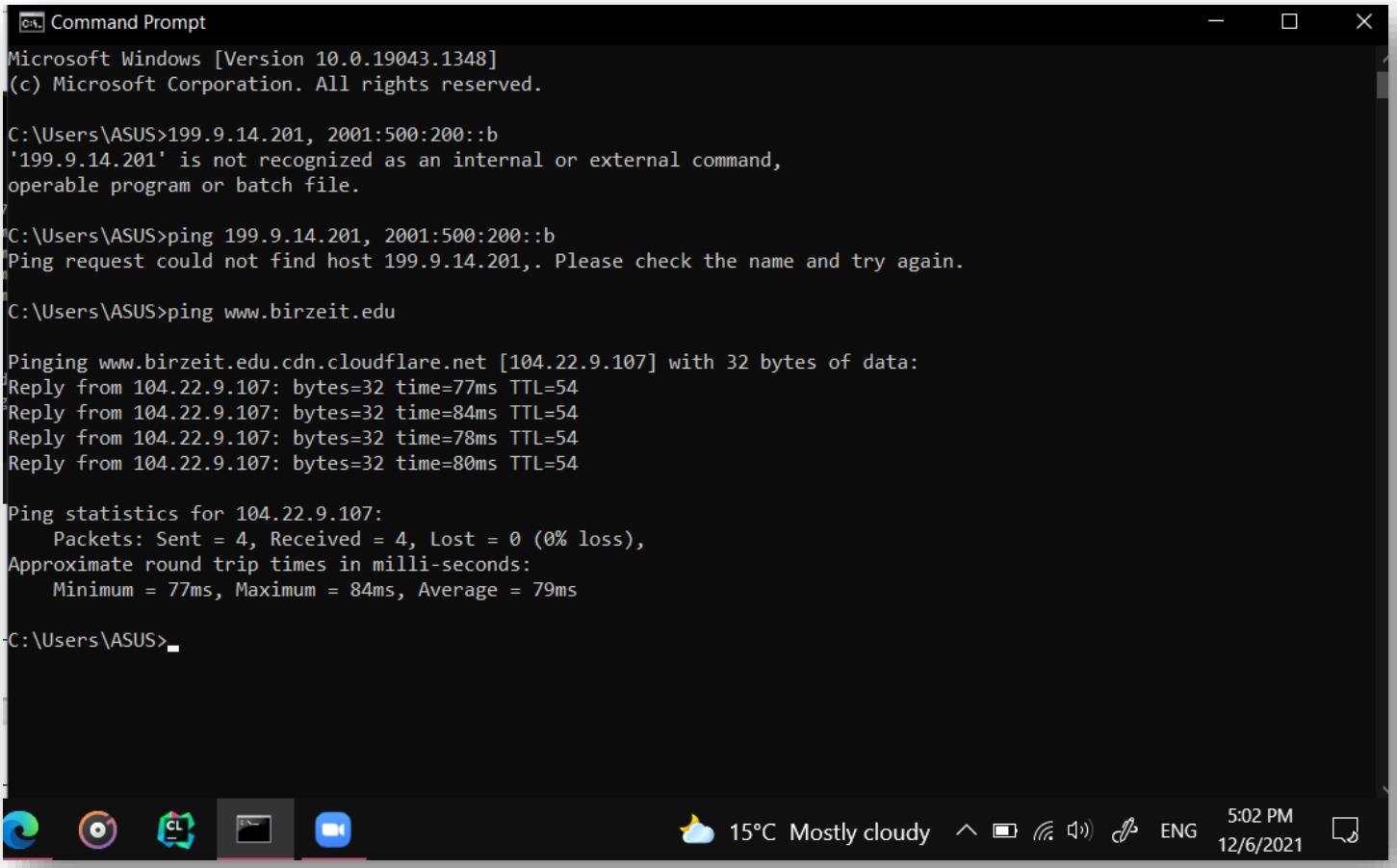
1. Part I:

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



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.

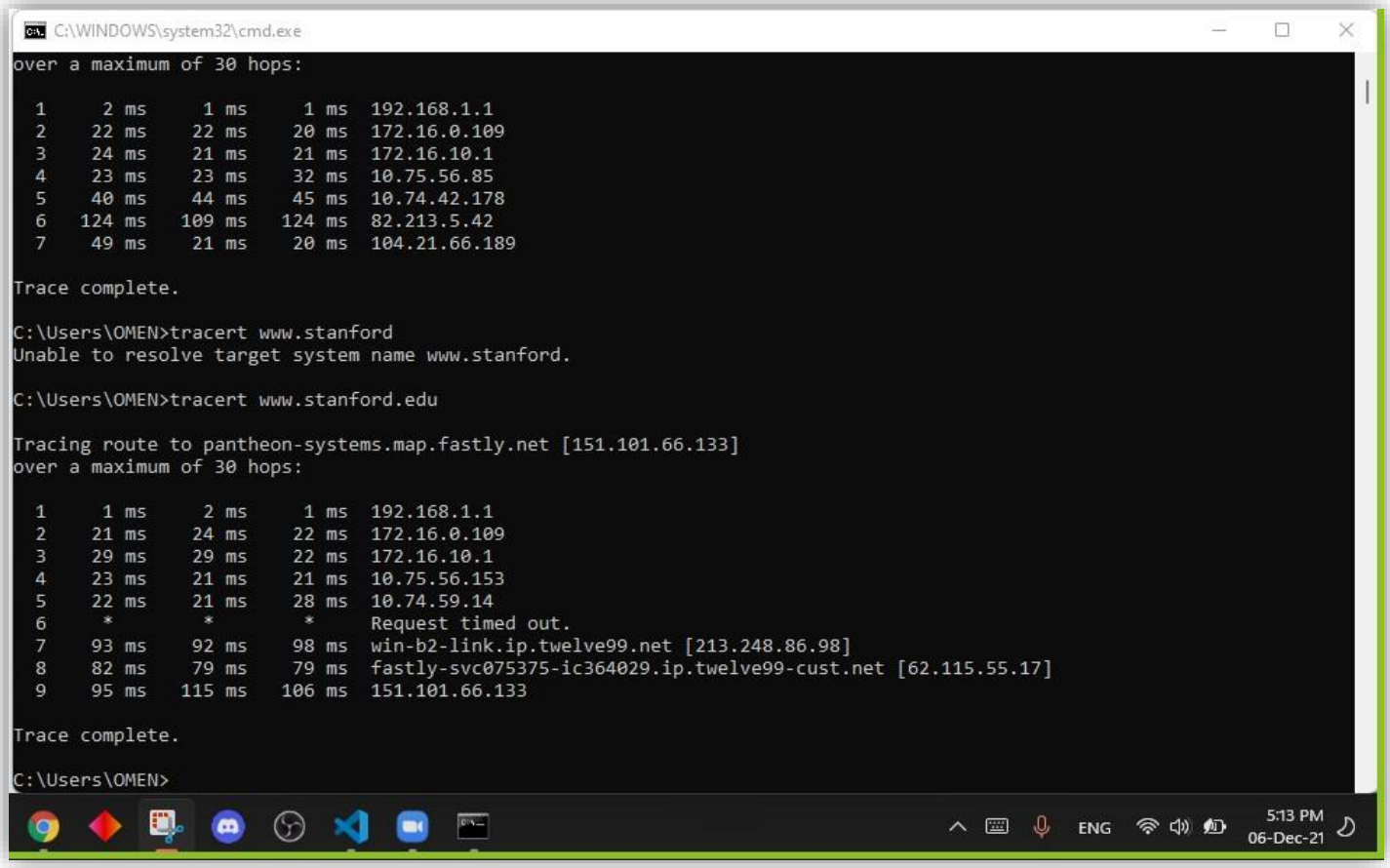
1.2. Ping www.birzeit.edu



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

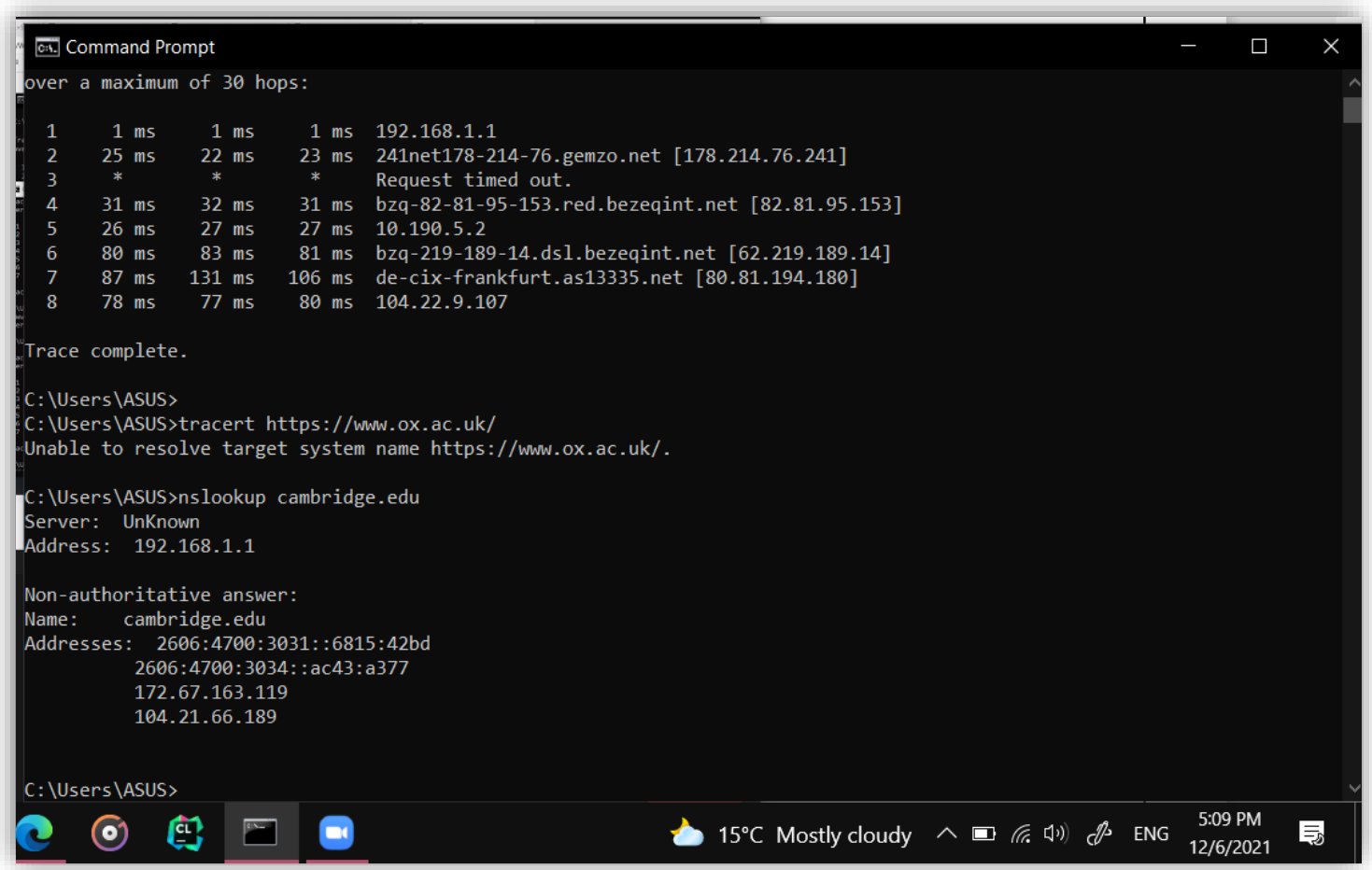
We sent 4 packets that all have the same time to live and we received them with deferent delays, with an average 79ms.

1.3. Tracert www.stanford.edu



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



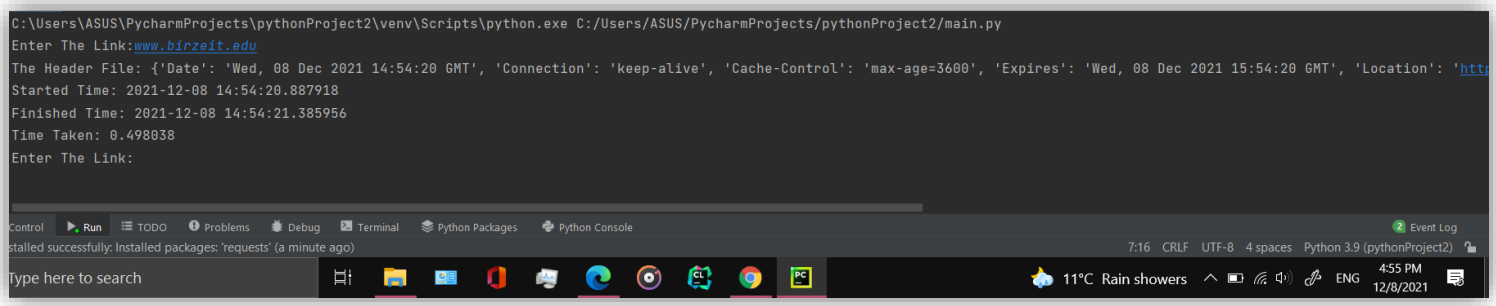
Will show the IP address of the device corresponding to the host which is the device that I am working on, and print the name and address of the host to which we sent the prop.

2. Part II:

2.1. The code:

```python
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>\t")  # 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
        # 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:



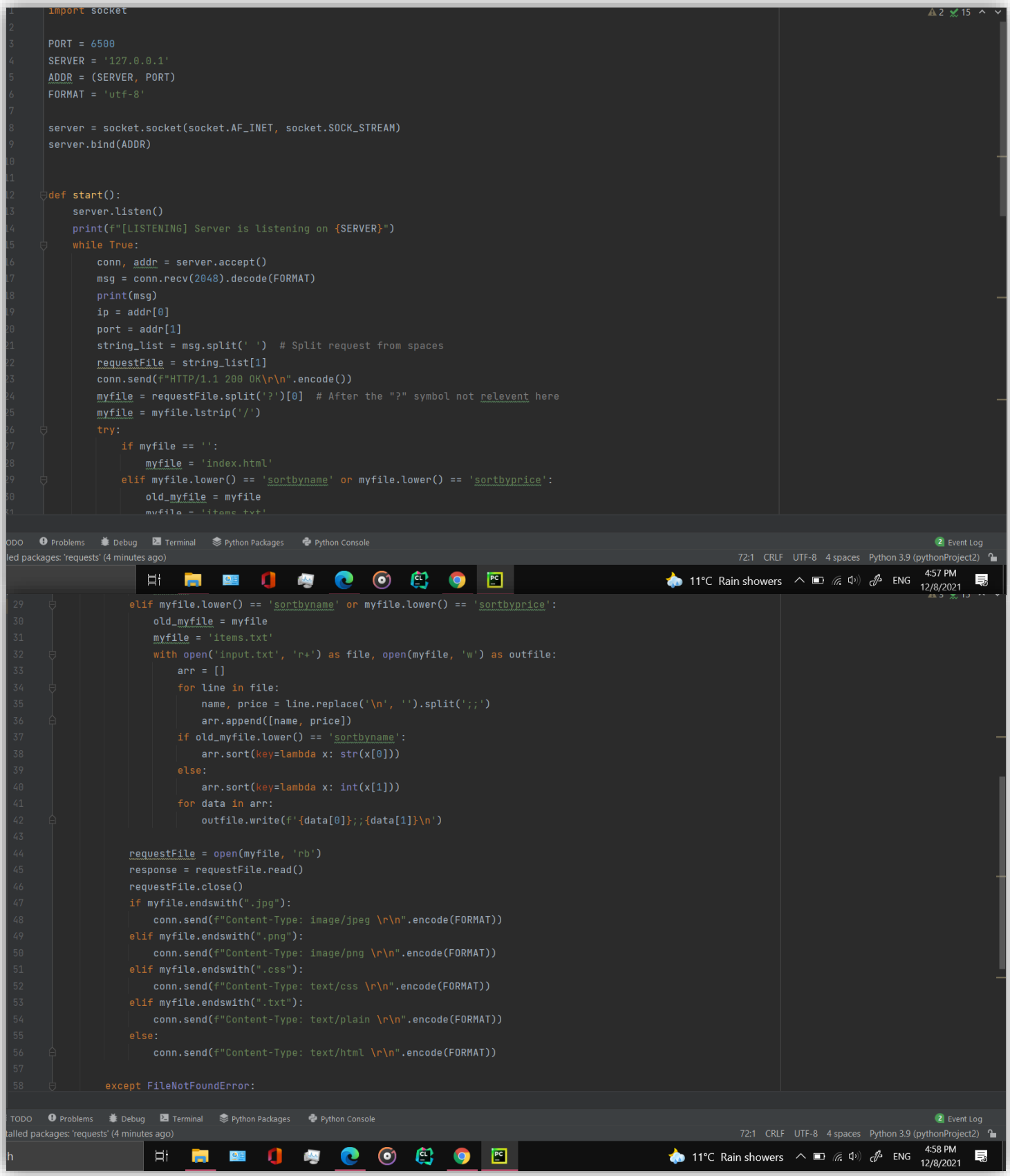
Showing the time of request, time of response the time interval, and the ability to enter new link (the program will exit when we enter -1).

2.3. Code as text with comments:

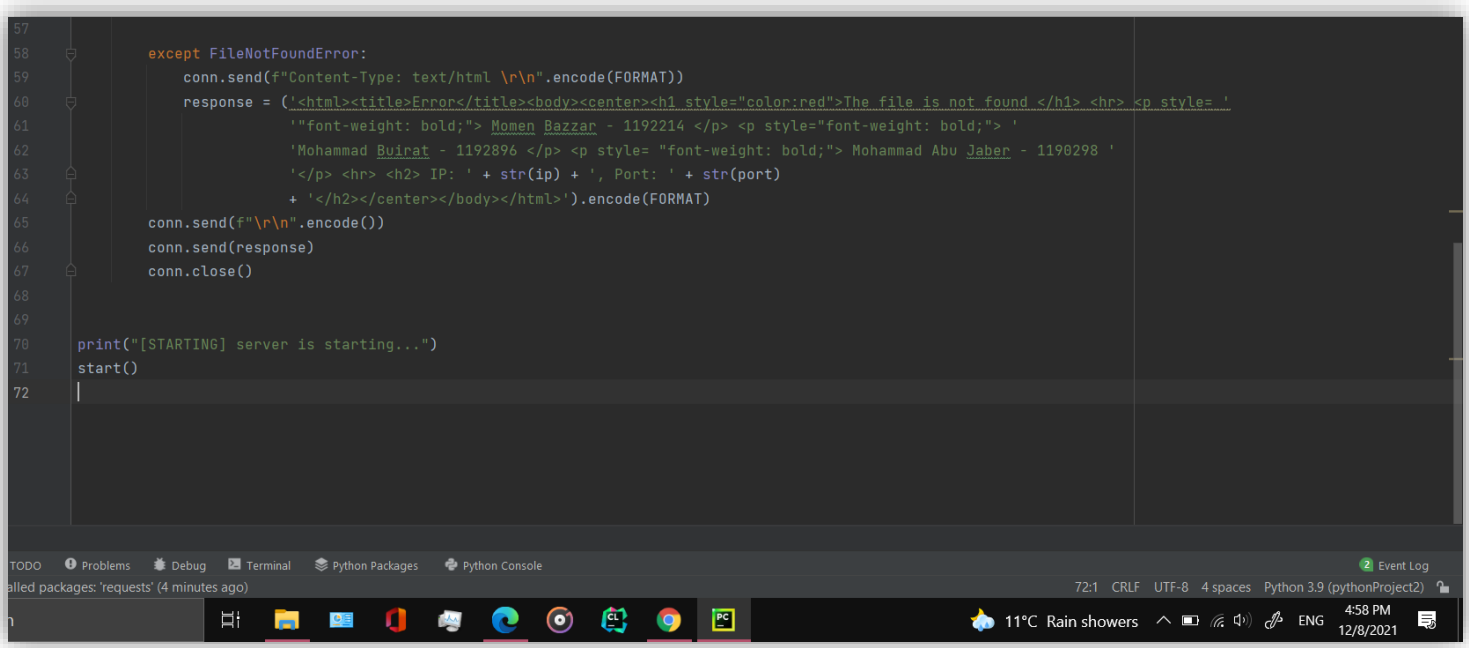
```
import requests #to import the library requests
import datetime #to import the library datetime which will help in calculating time taken from sending the request until we receive the response

while True: #infinite loop

    url = input("Enter The Link: <enter -1 to exit>\t") #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:



```
import socket

PORT = 6500
SERVER = '127.0.0.1'
ADDR = (SERVER, PORT)
FORMAT = 'utf-8'

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind(ADDR)


def start():
    server.listen()
    print(f"[LISTENING] Server is listening on {SERVER}")
    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 == '':
                myfile = 'index.html'
            elif myfile.lower() == 'sortbyname' or myfile.lower() == 'sortbyprice':
                old_myfile = myfile
                myfile = 'items.txt'
                with open('input.txt', 'r+') as file, open(myfile, 'w') as outfile:
                    arr = []
                    for line in file:
                        name, price = line.replace('\n', '').split(';;')
                        arr.append([name, price])
                    if old_myfile.lower() == 'sortbyname':
                        arr.sort(key=lambda x: str(x[0]))
                    else:
                        arr.sort(key=lambda x: int(x[1]))
                    for data in arr:
                        outfile.write(f'{data[0]};;{data[1]}\n')

            requestFile = open(myfile, 'rb')
            response = requestFile.read()
            requestFile.close()
            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:
```



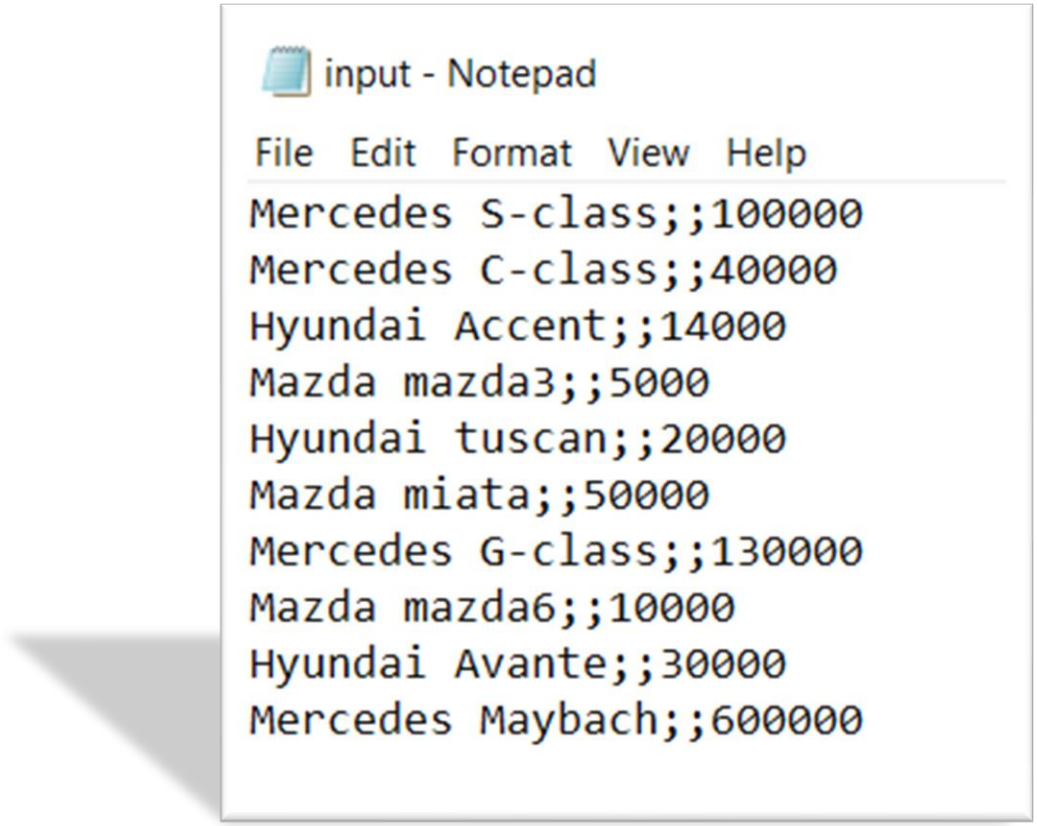
3.2. code as text whit comments:

```
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('/')
```

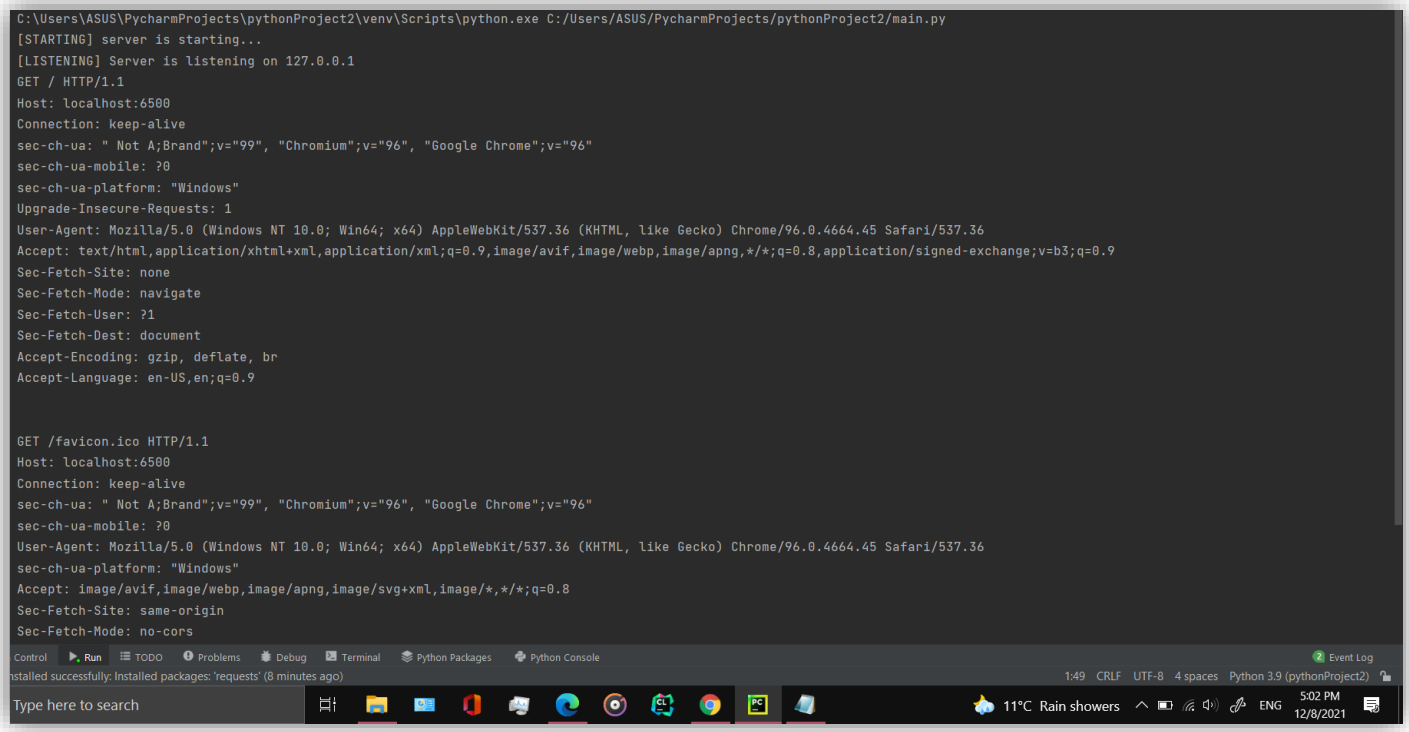
```
        try:
            if myfile == '': #If nothing was send with the request, the default is the main html file
                myfile = 'index.html'
            elif myfile.lower() == 'sortbyname' or myfile.lower() == 'sortbyprice': #It will convert the request into lower case
                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()
```

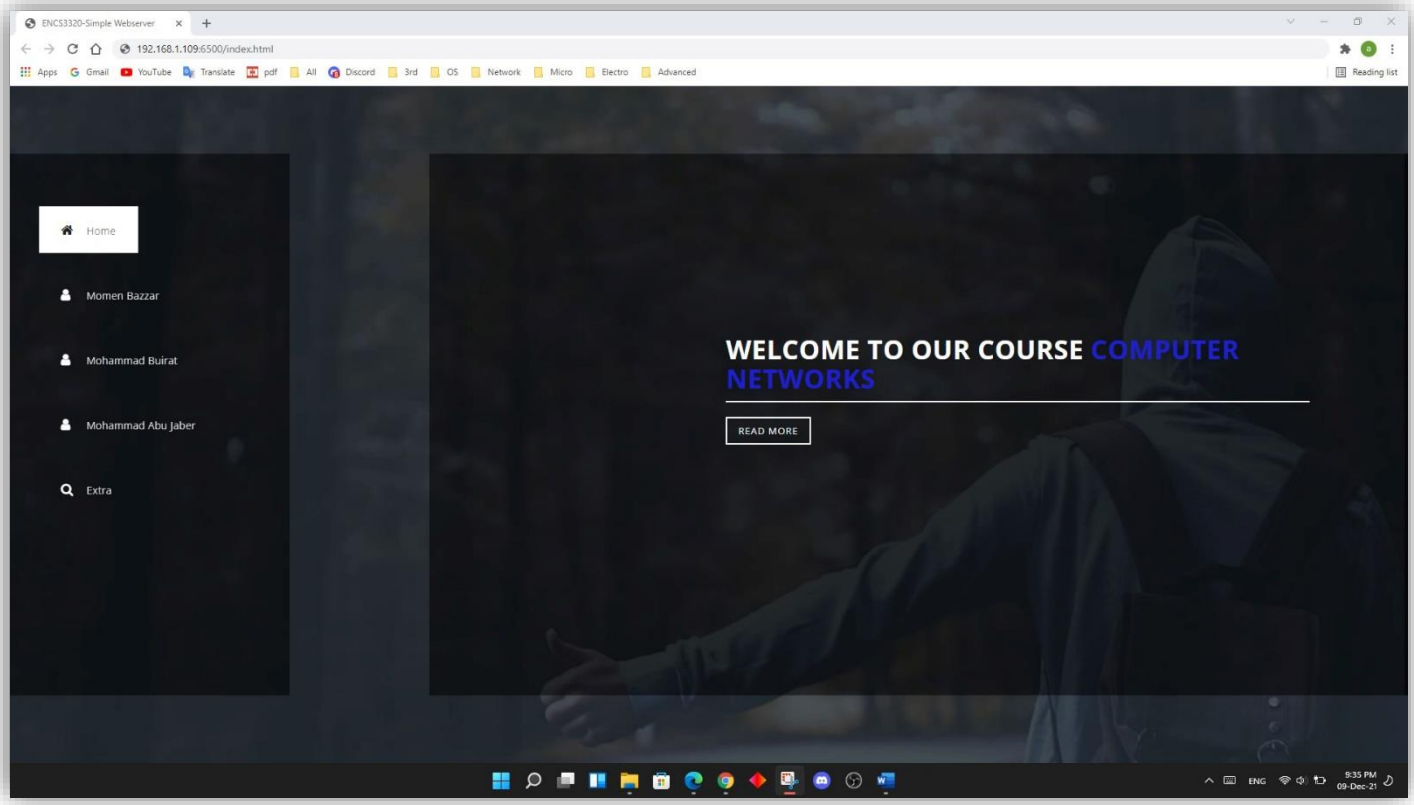
3.3. The text file that we read the data from:



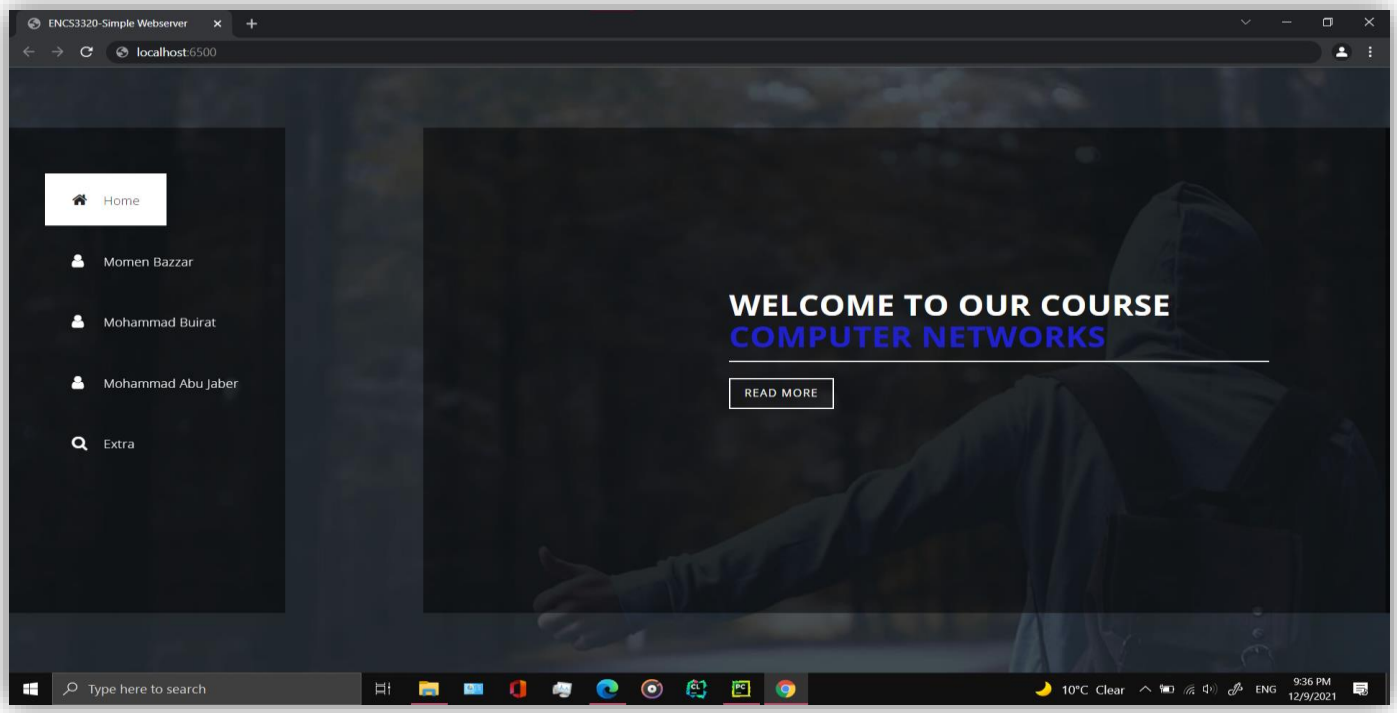
3.4. The HTTP response that we get when we open the browser and request localhost:6500

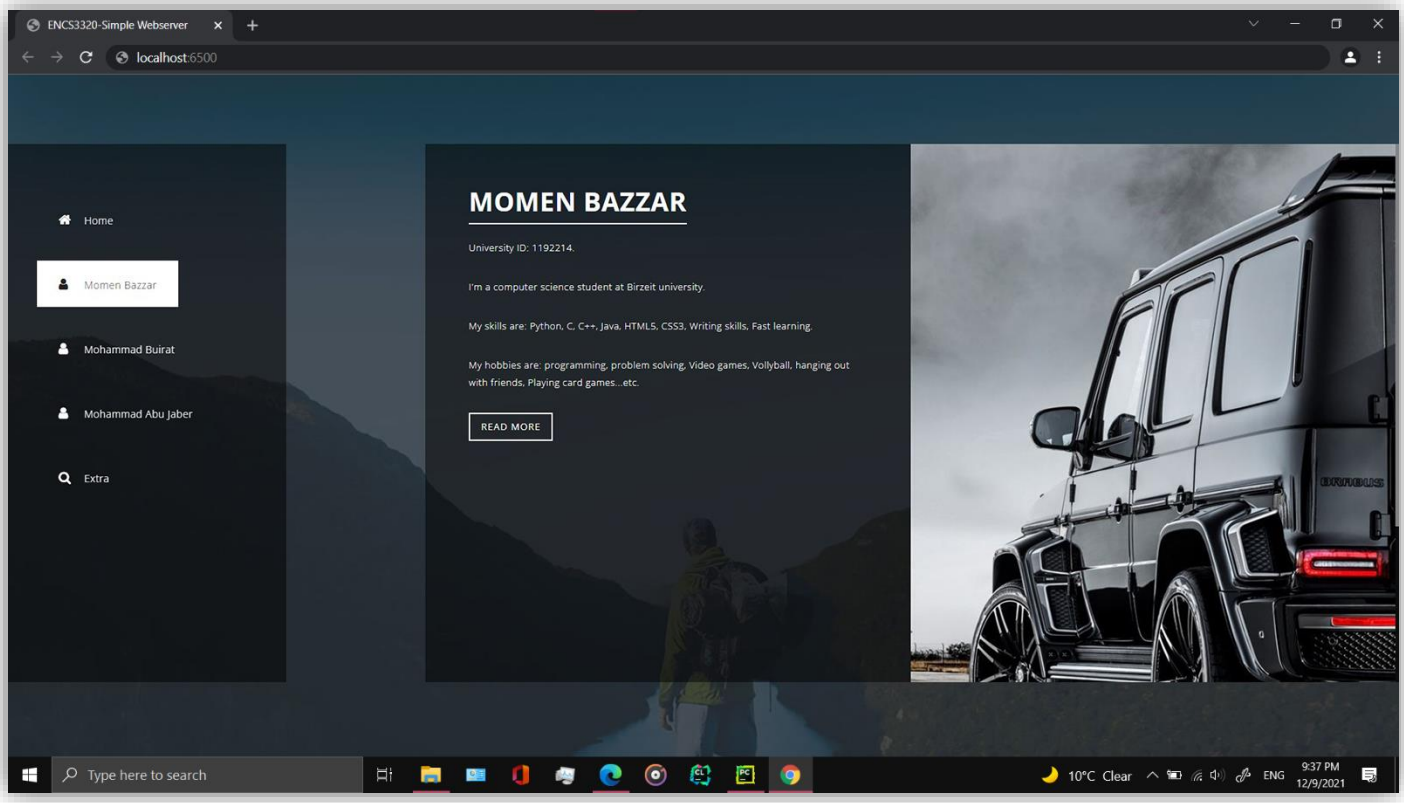


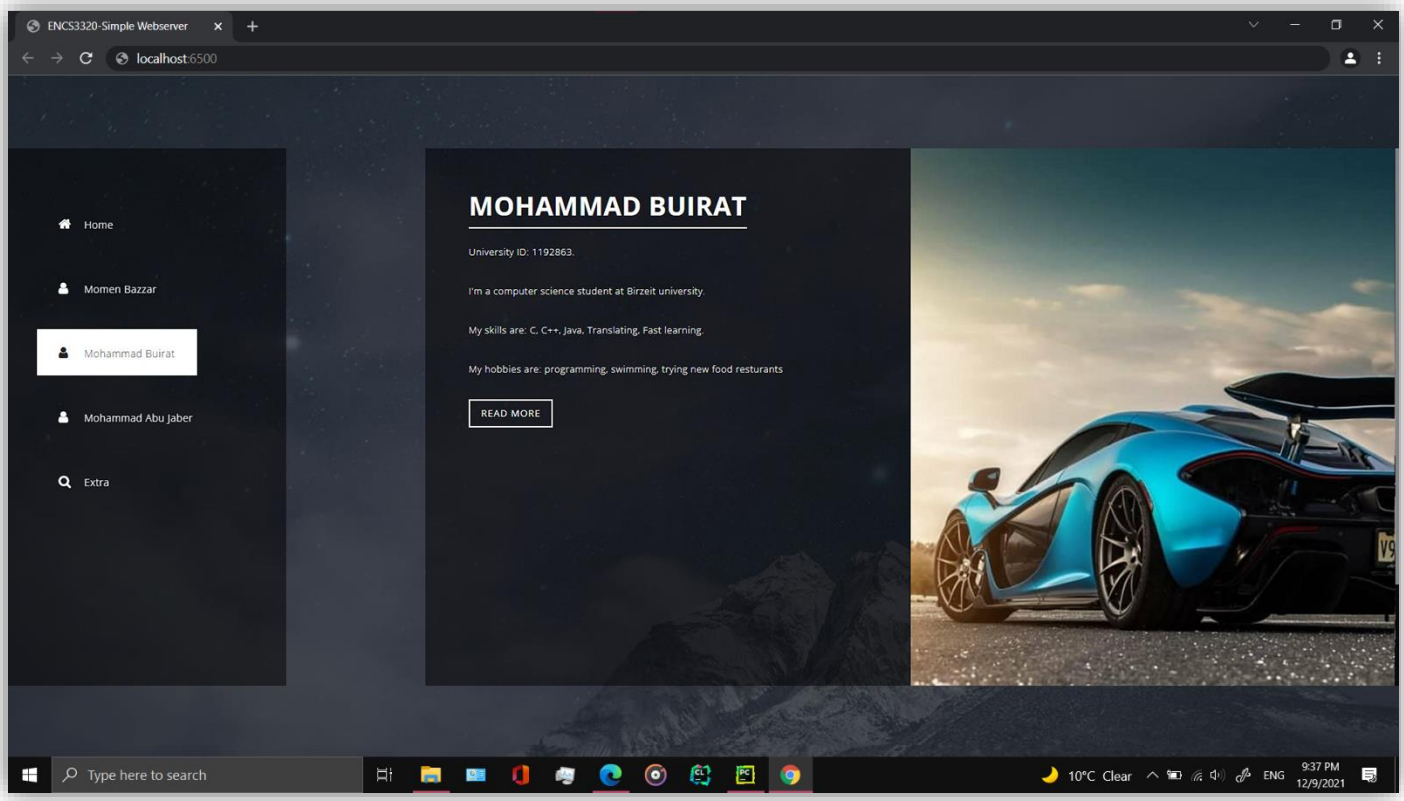
3.5. Screenshot from another computer: 192.168.1.109:6500/index.html

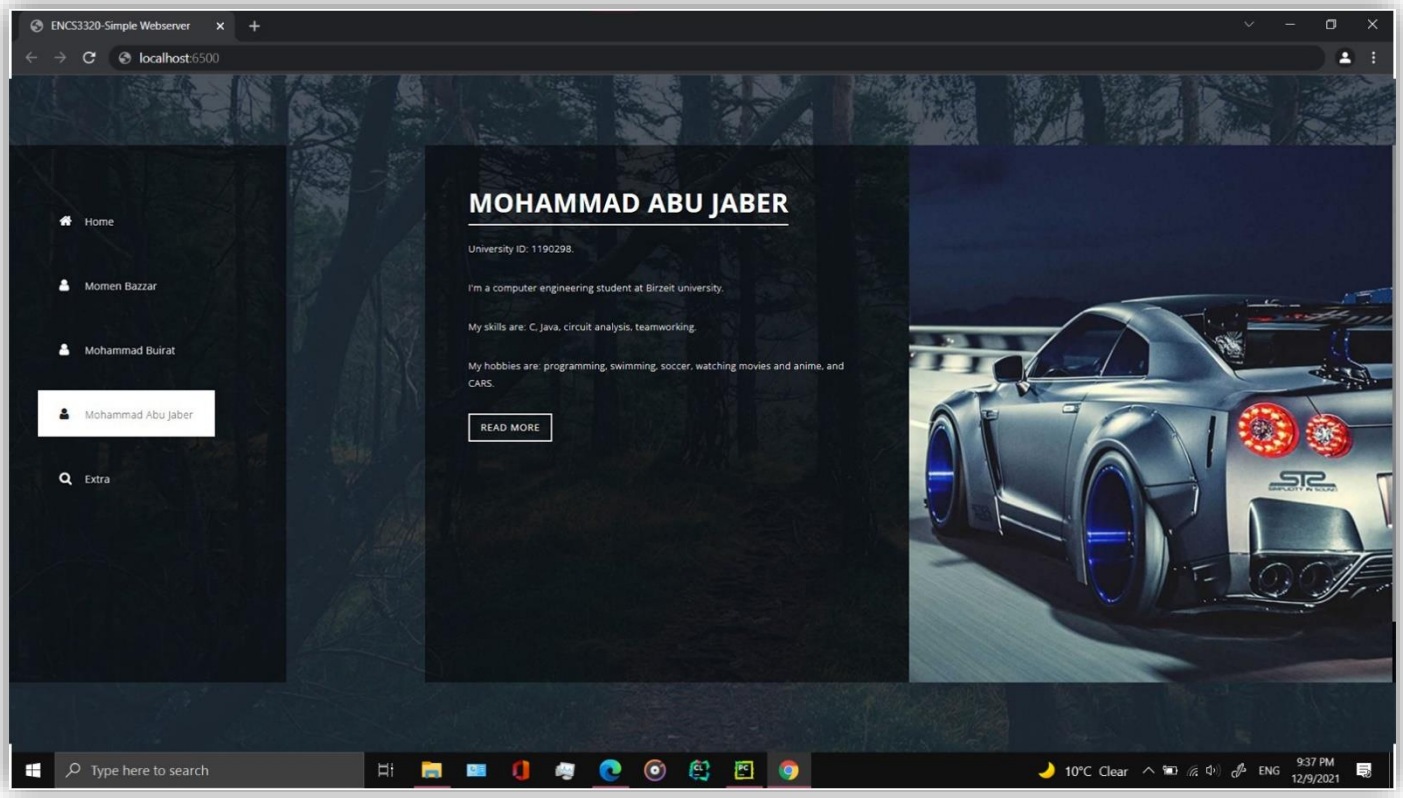


3.6. Localhost:6500 or Localhost:6500/index.html

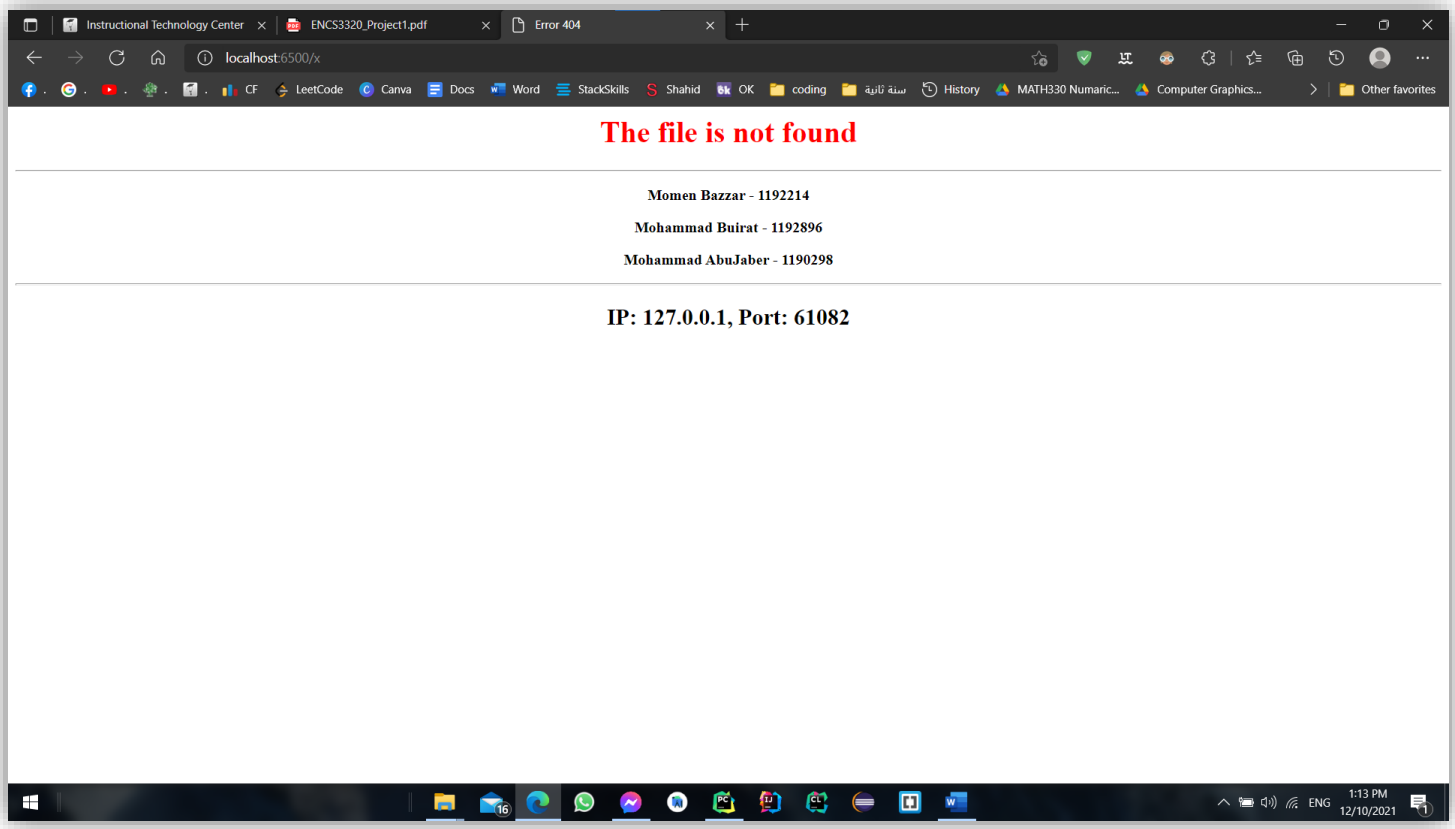





ENCS3320-Simple Webserver
localhost:6500
Home
Momen Bazzar
Mohammad Buirat
Mohammad Abu Jaber
Extra
MOHAMMAD BUIRAT
University ID: 1192863.
I'm a computer science student at Birzeit university.
My skills are: C, C++, Java, Translating, Fast learning.
My hobbies are: programming, swimming, trying new food resturants
READ MORE
Type here to search
10°C Clear
ENG
9:37 PM
12/9/2021

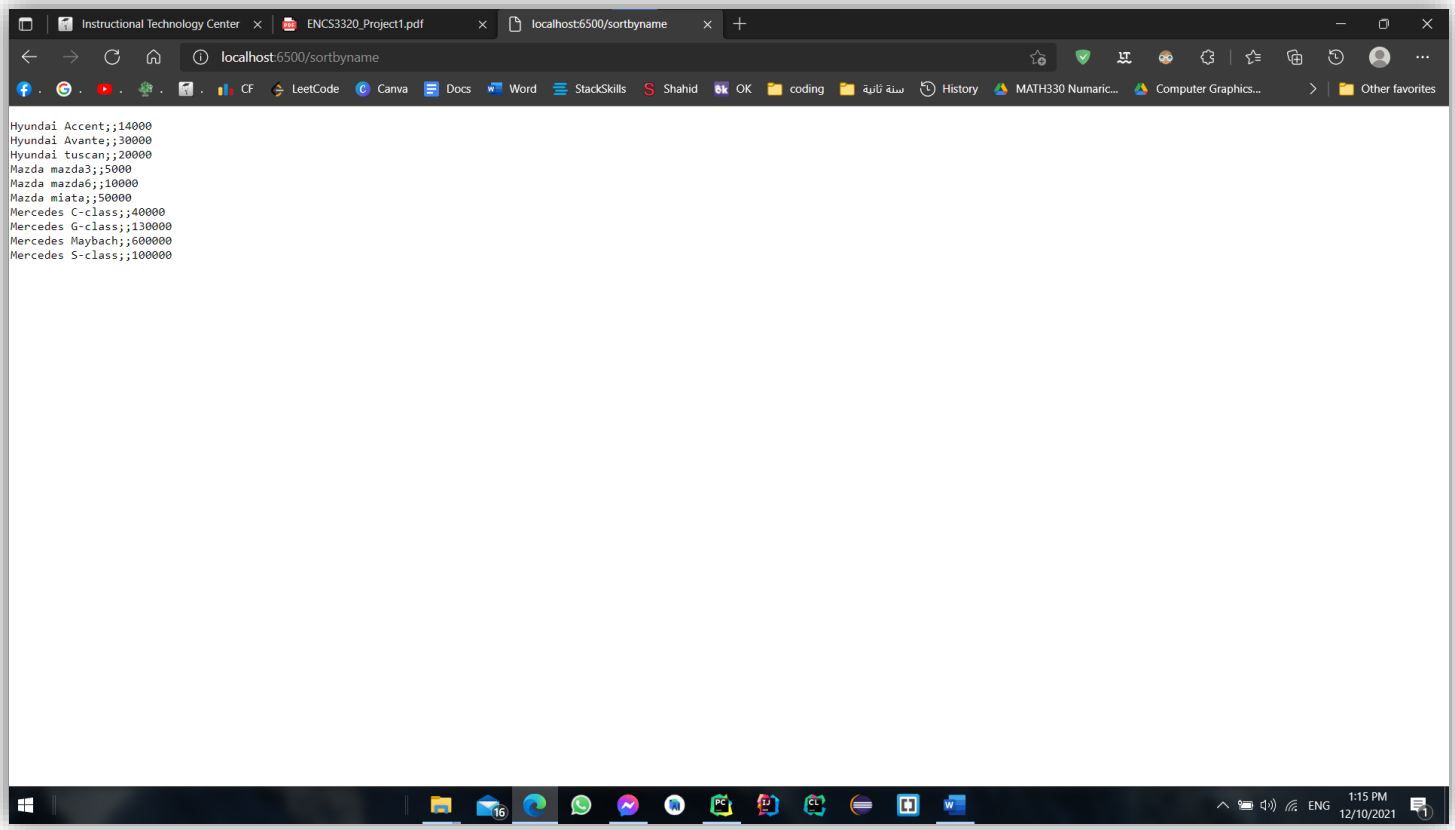

ENCS3320-Simple Webserver
localhost:6500
Home
Momen Bazzar
Mohammad Buirat
Mohammad Abu Jaber
Extra
MOHAMMAD ABU JABER
University ID: 1190298.
I'm a computer engineering student at Birzeit university.
My skills are: C, Java, circuit analysis, teamworking.
My hobbies are: programming, swimming, soccer, watching movies and anime, and CARS.
READ MORE
Type here to search
10°C Clear
ENG
9:37 PM
12/9/2021

3.7. Localhost:6500/x

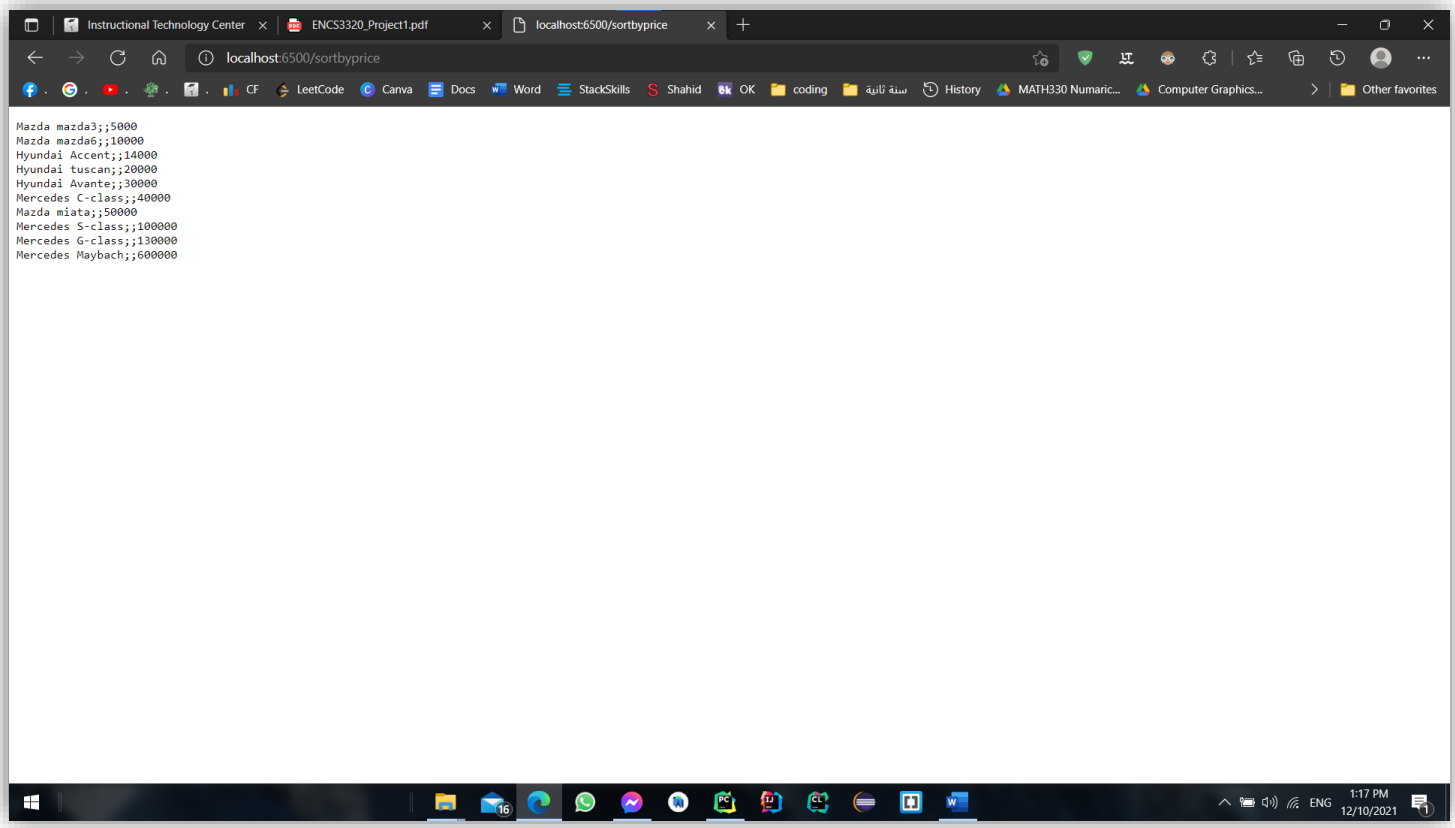


/X return the error page because it's not a valid request or file

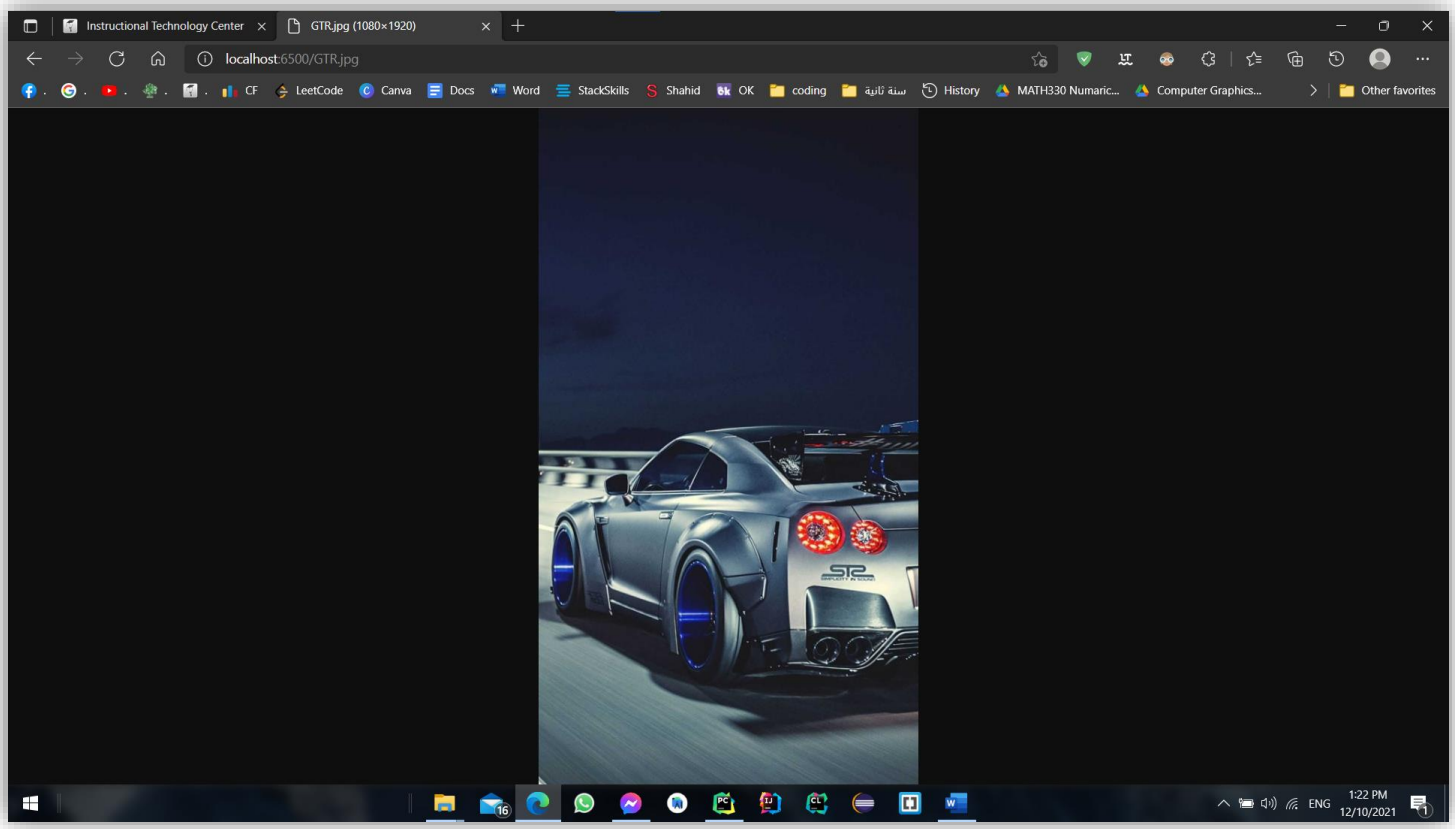
3.8. /SortByName



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
```