Birzeit University-Faculty of Engineering and Technology

Electrical and Computer Engineering Department

COMPUTER NETWORKS ENCS3320

Network Project Report
Complete Web Server

Partners:

Mayar Abuzahra	**Student ID:** 1181239
Shahed Jamhour	**Student ID:** 1180654
Yasmeena Assi	**Student ID:** 1180899

Supervised by: Dr. Abdalkarim Awad

Section: 1

Submission Date: 14-8-2021

❖ Abstract

This project aims to implement HTTP-Web server using socket programming. Using python language, we have implemented a web server that does many types of requests [html file requests, pictures with different extensions, csv files requests]. If the request is not found, html file with specific properties will appear to tell the client that his/her request is not found.

❖ Contents

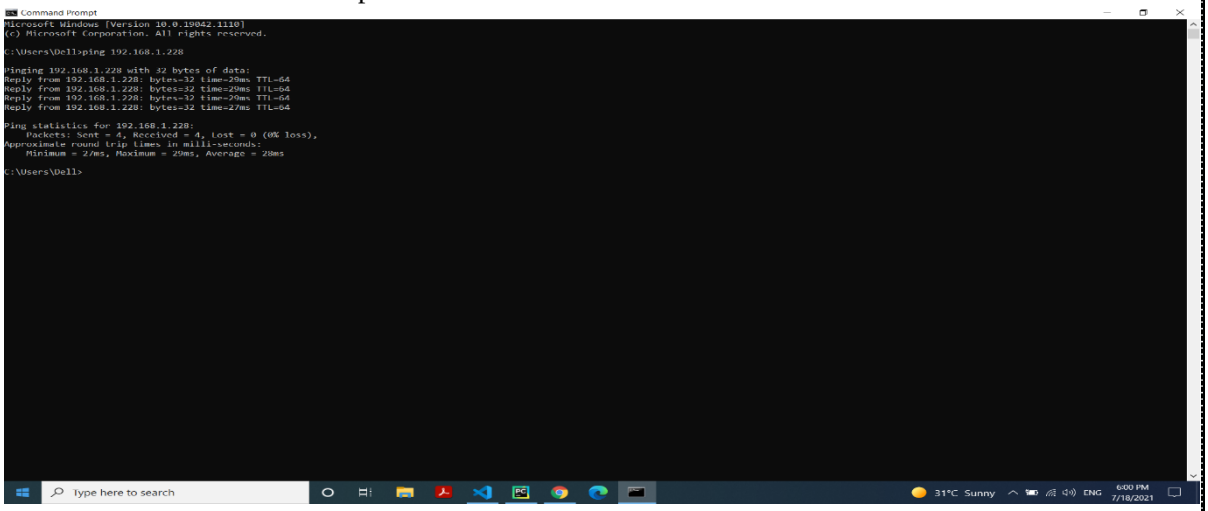
❖ Procedure & Calculations

A. Part1:

1. ping 192.168.1.228

We use this ping to test if the computer is reachable at specific Internet protocol, also to measure the round trip for a massage that has been sent from a host to a destination computer,

As we see in the picture, we can see there is 4 responses with a small response time. when we put enter after the command the request sent in 4 times, there will be a reply with a round trip static as we see or a timeout failure. As we see there is four lines which we can see from it the size of the data that we are going to see it ,and TTL time which means the time that packet has being inside the network before being discard, also , we can see the time of response, we can get effect from the ping instruction to see the time that we need to send and receive data so we know the speed ,Also as shown in the figure we can see how much packet we sent and how much we received , from this we can check if there is a lost packets.

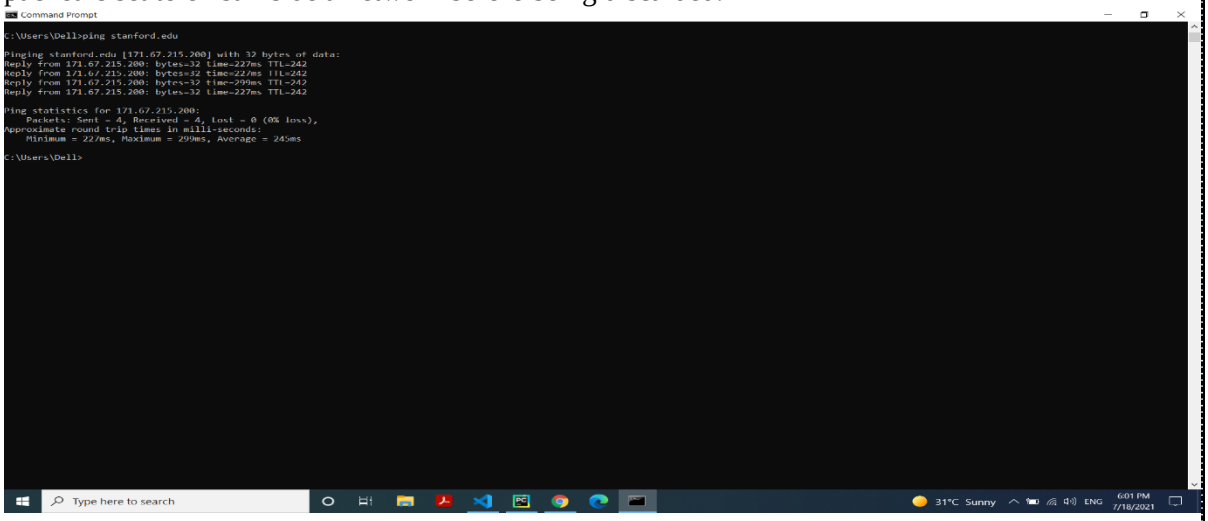


2. ping stanford.edu

As we said before we use a ping to check and test many things, from it we can transmits packet to a special Ip Address, and the time of response and the time to live.

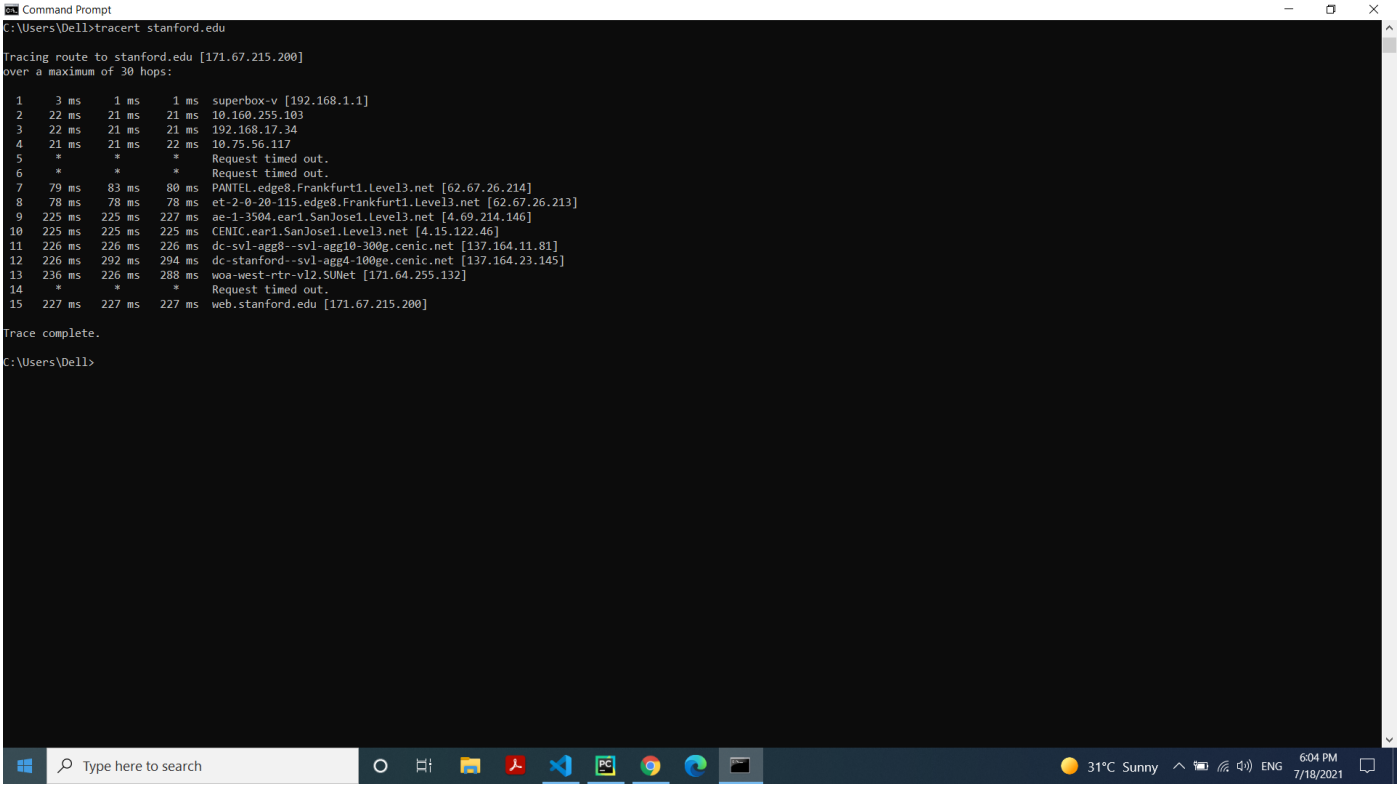
when we use this command when ping a host IP Address ,so the packet sent to the destination ip Adress that we chose it , when it reached the destination Address send a response for the request

As shown in the figure, we can see the number of packets that sent or received its equal so there is no loss , as we see from the figure lost=0, so the connection is successful due to these things , also we can notice the TTL time ,which means time to live (TTL) refers to the amount of time or that a packet is set to exist inside a network before being discarded.



3. tracert Stanford.edu

traceouts is a command line included with windows and other operating system, We can use it to see the internet connection problem like the loses and latency, so this command tells us where is the problem exactly. We use it to get the path from our networks to the destination, also it gives me information about it like the routes in the path number of hopes, also if there is a mistake, the response times it shows to us the path traffic takes to reach a specific website, also we can see the delays in each stop.

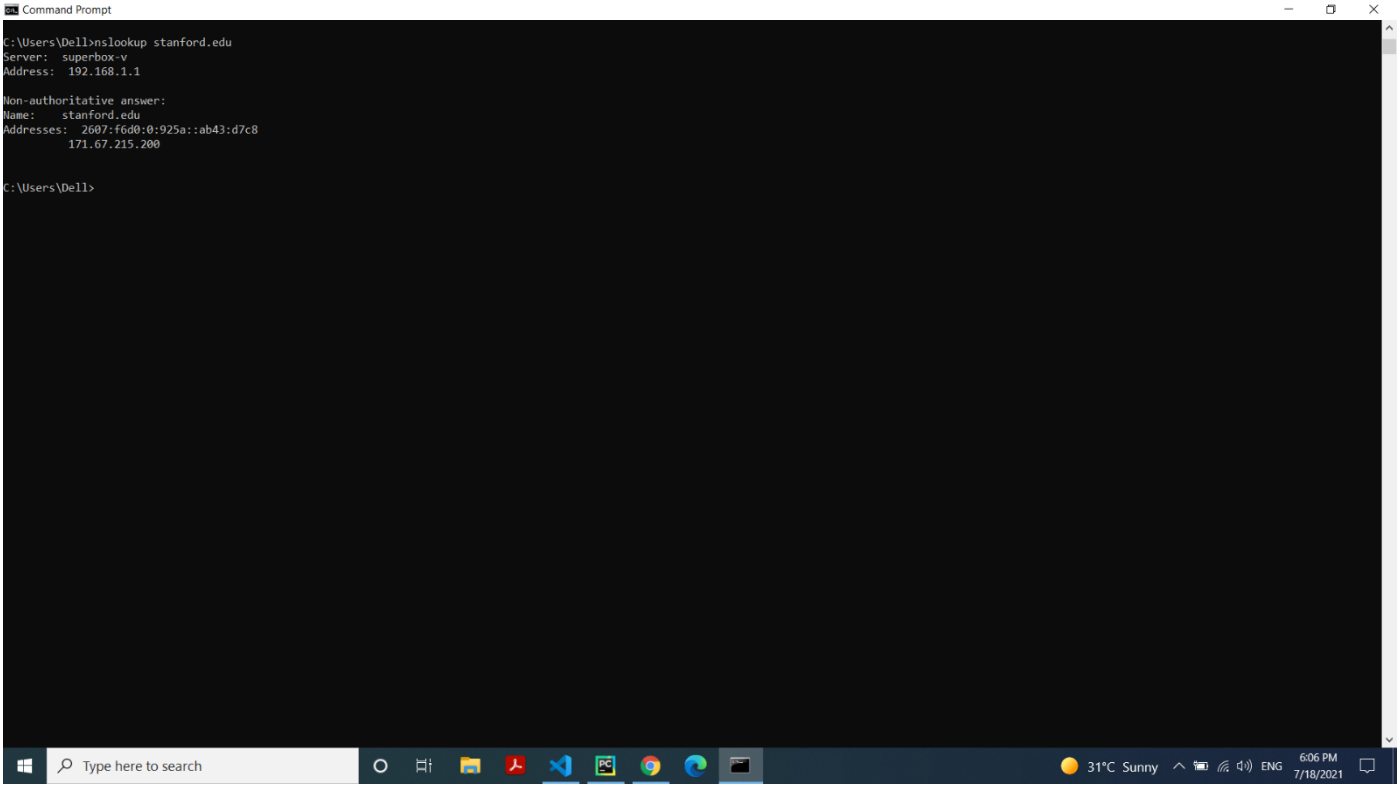


```
Command Prompt
C:\Users\Dell>tracert stanford.edu

Tracing route to stanford.edu [171.67.215.200]
over a maximum of 30 hops:

  1     3 ms     1 ms     1 ms  superbox-v [192.168.1.1]
  2    22 ms    21 ms    21 ms  10.160.255.103
  3    22 ms    21 ms    21 ms  192.168.17.34
  4    21 ms    21 ms    22 ms  10.75.56.117
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7    79 ms    83 ms    80 ms  PANTEL.edge8.Frankfurt1.Level3.net [62.67.26.214]
  8    78 ms    78 ms    78 ms  et-2-0-20-115.edge8.Frankfurt1.Level3.net [62.67.26.213]
  9   225 ms   225 ms   227 ms  ae-1-3504.ear1.SanJose1.Level3.net [4.69.214.146]
 10   225 ms   225 ms   225 ms  CENIC.ear1.SanJose1.Level3.net [4.15.122.46]
 11   226 ms   226 ms   226 ms  dc-svl-agg8--svl-agg10-300g.cenic.net [137.164.11.81]
 12   226 ms   292 ms   294 ms  dc-stanford--svl-agg4-100ge.cenic.net [137.164.23.145]
 13   236 ms   226 ms   288 ms  woa-west-rtr-vl2.SUNet [171.64.255.132]
 14     *        *        *     Request timed out.
 15   227 ms   227 ms   227 ms  web.stanford.edu [171.67.215.200]

Trace complete.

C:\Users\Dell>
Type here to search
31°C Sunny ENG 6:04 PM 7/18/2021
```

4. nslookup stanford.edu

ns lookup command is available in many operating systems as windows, Linux, macOS, we can use it to get the DNS queries and receive the IP addresses and domain names. This command has two modes: interactive mode the command from itself give a default name server when there is no arguments are given, The non-interactive mode when there is a name or internet Address to looked up for it. So, this command. Here as we see from the figure there is a local DNS server which used to reach the IP address of the server from the closer DNS. Also here in the figure shows the IP Address for a DNS server, and

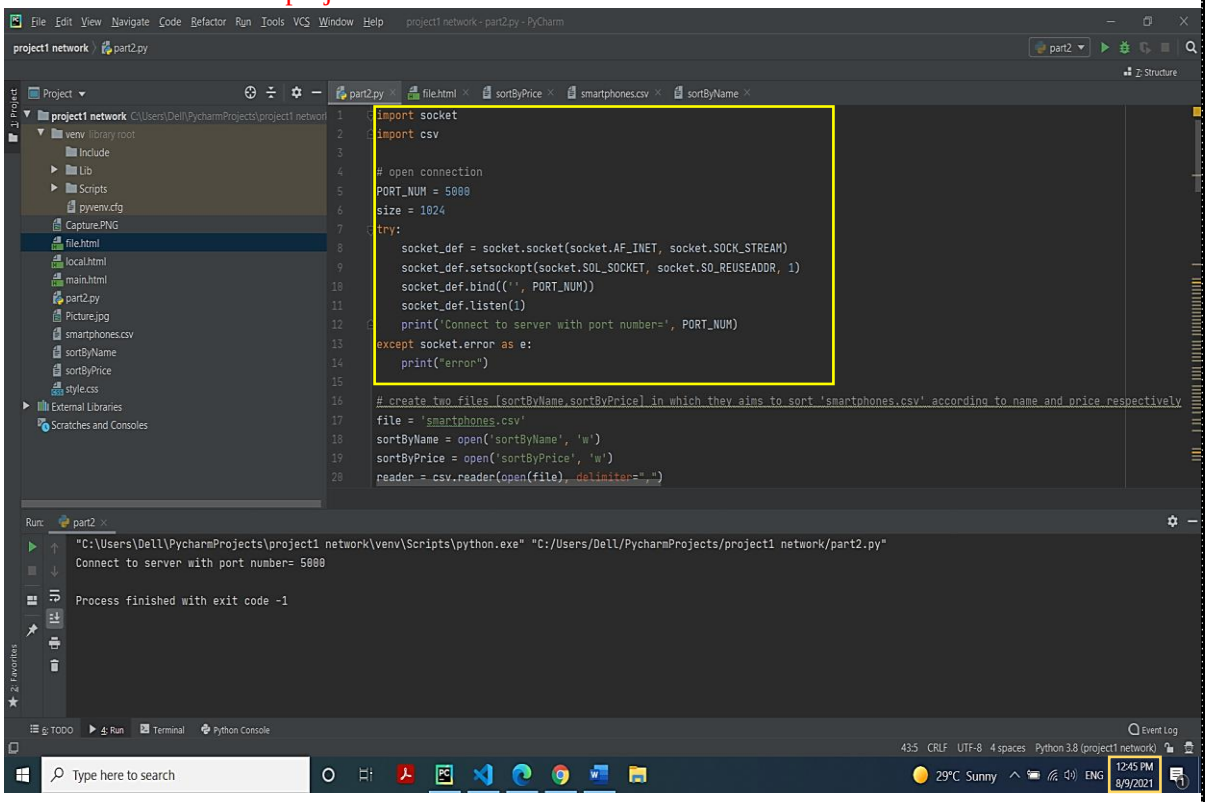
the popular name for the ,Also the IP address for the server standford, also a name of paths to reach to the server



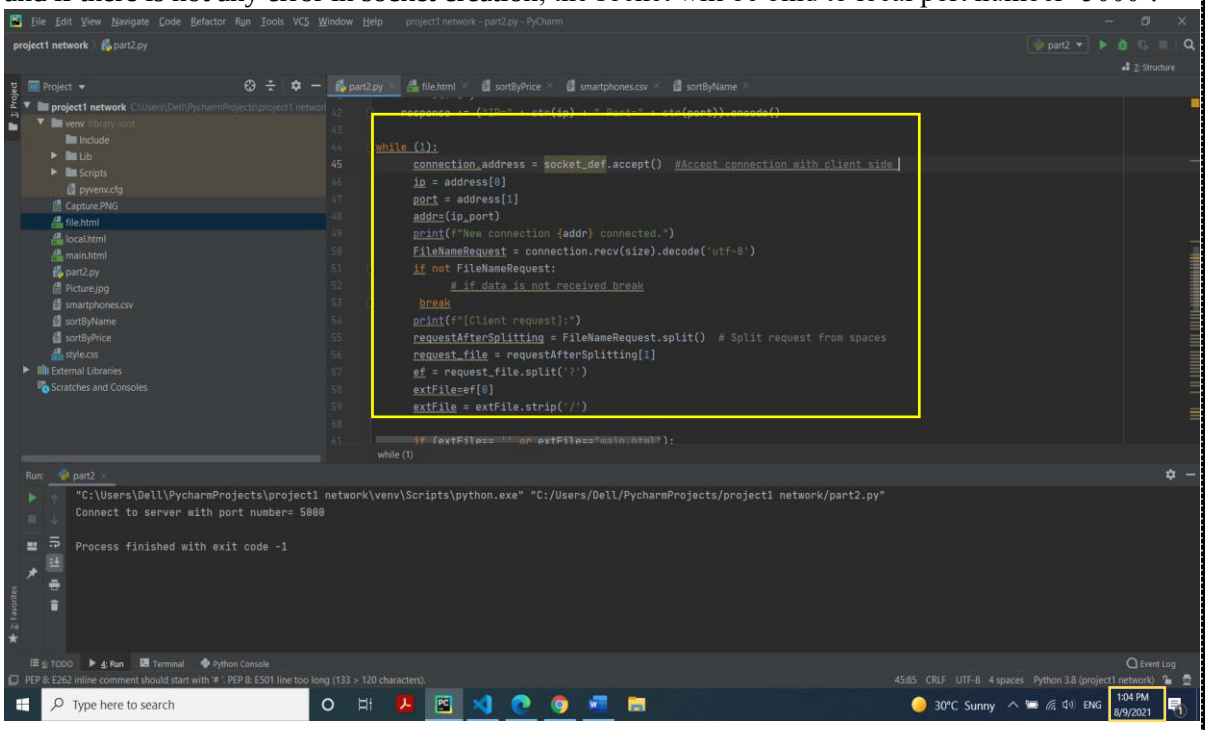
B. **Part2:**

In this part, many types of requests will be applied and different kinds of responds will be done.

All files are attached in project submission:

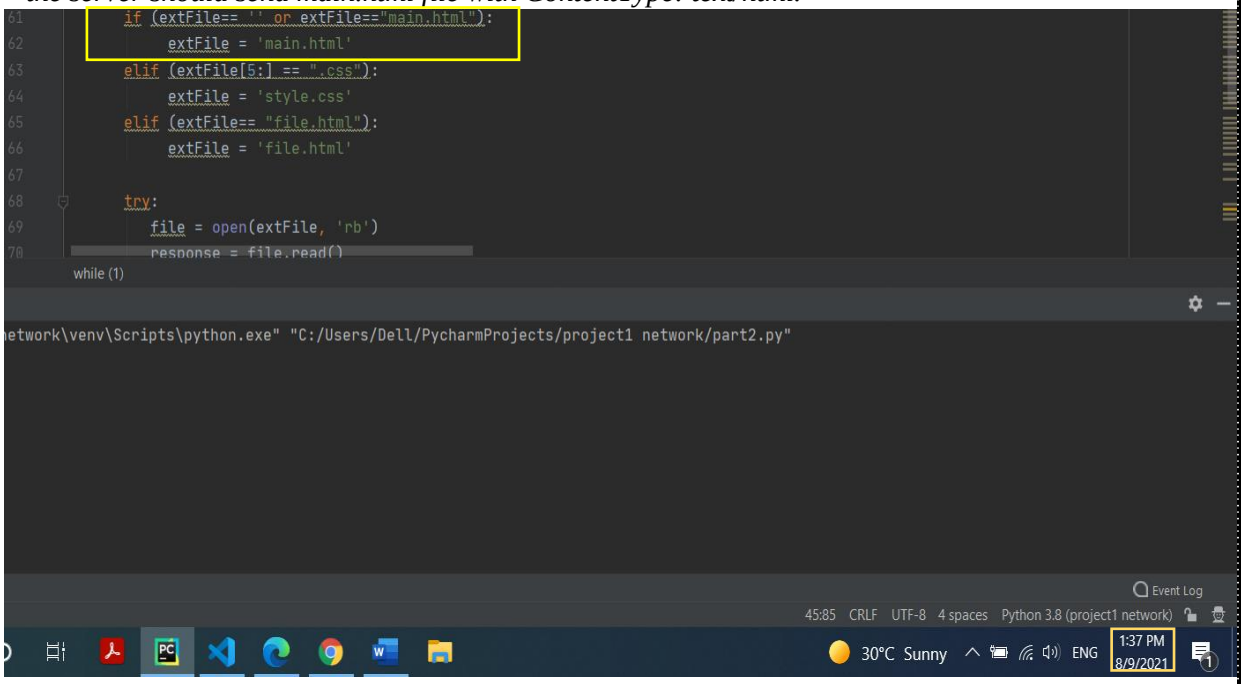


After importing socket and csv libraries, this piece of code aims to create a UDP socket for server and if there is not any error in socket creation, the socket will be bind to local port number '5000'.

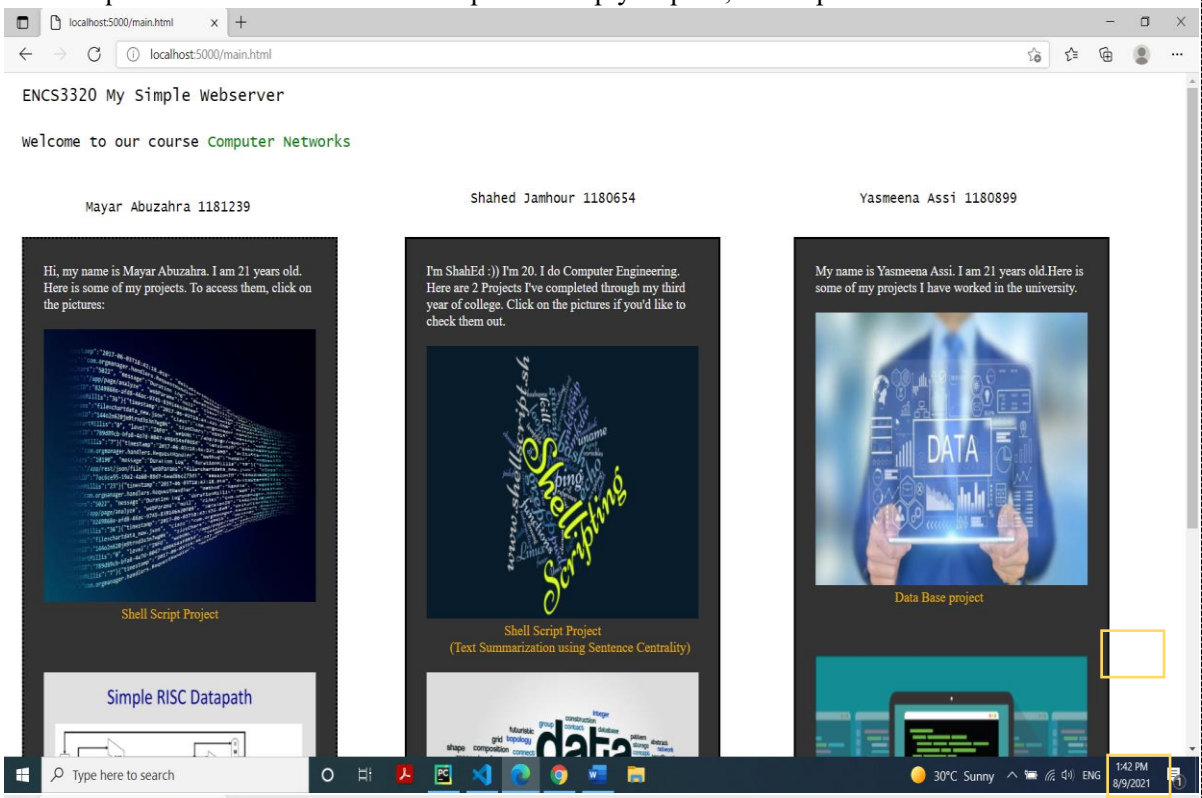


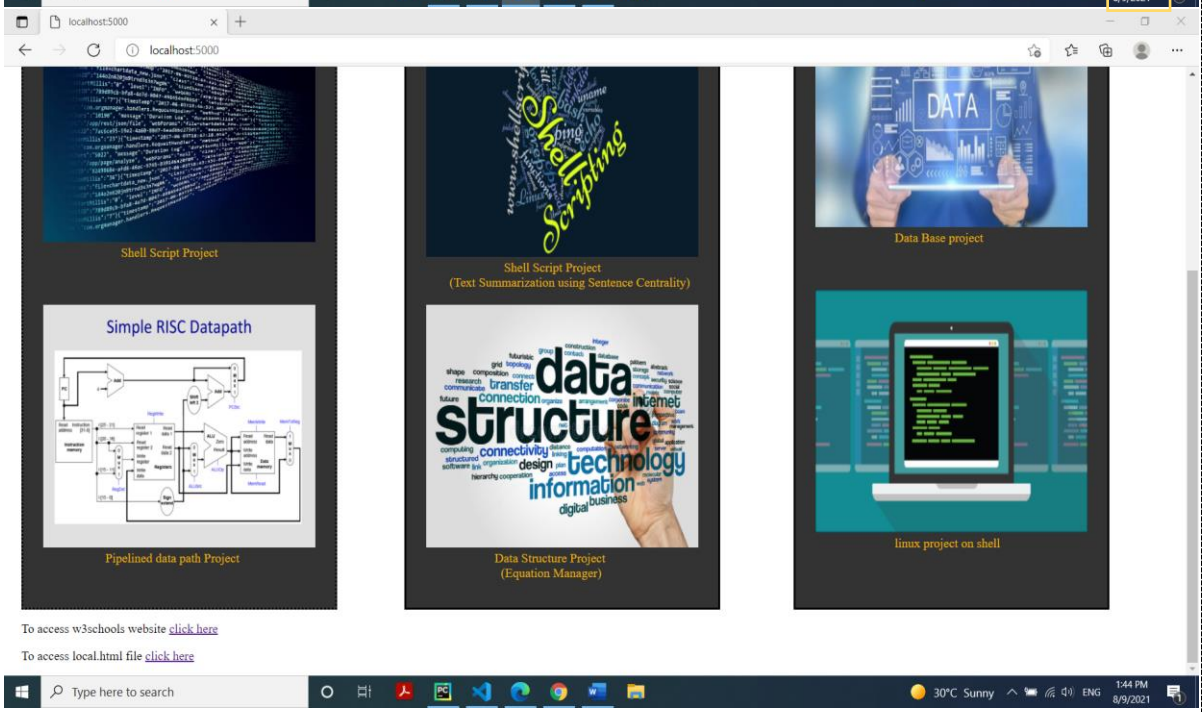
At first, line 45 determines which file represents the client request [html file, pictures, text files].
To get the name of the request file: 1- Split the ‘FileNameRequest’ by spaces and take the second string and save it in ‘request_file’.
2- Split ‘request_file’ by ‘?’ and take the first value after splitting and delete ‘/’ from it because out request will be in this format: http://localhost:5000/file_name.
After getting the file name, we need to check which file request is applied:

1- if the request is / or /index.html (for example localhost:9000/ or localhost:9000/index.html) then the server should send main.html file with ContentType: text/html.

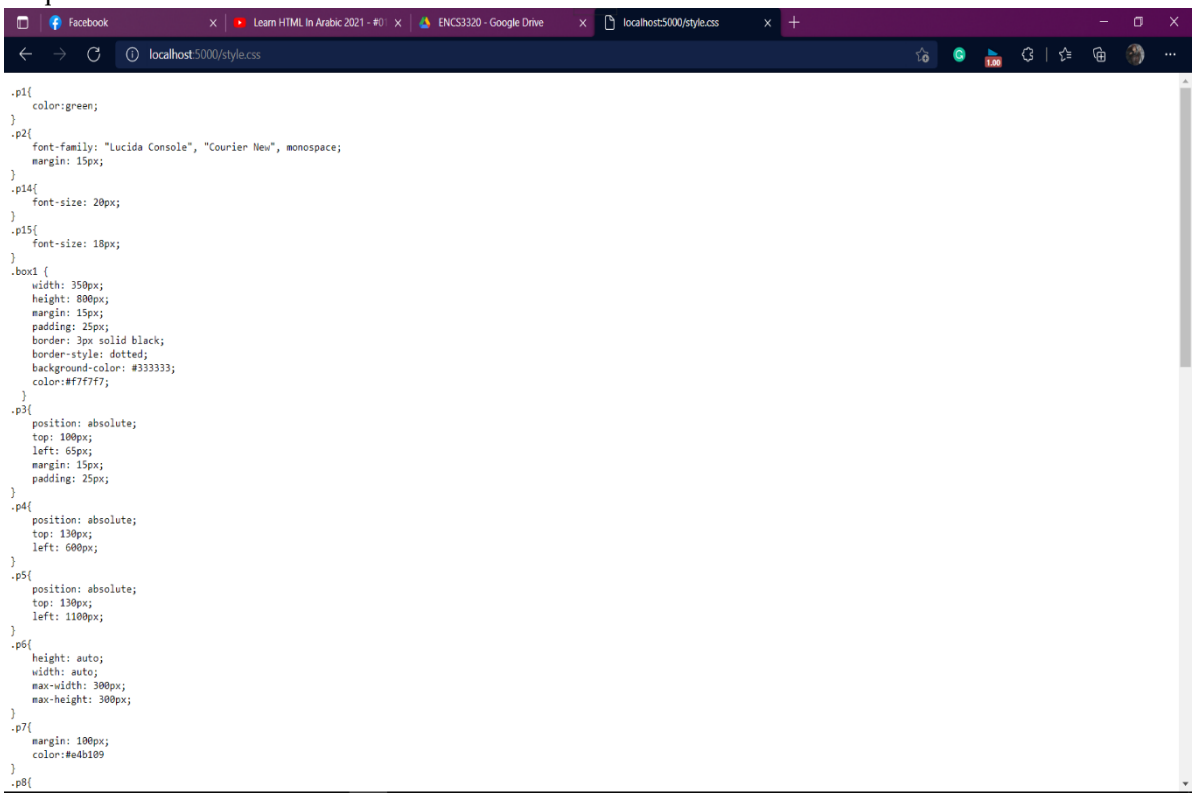


If the request is ‘main.html’ or if the request is empty request, the response is:

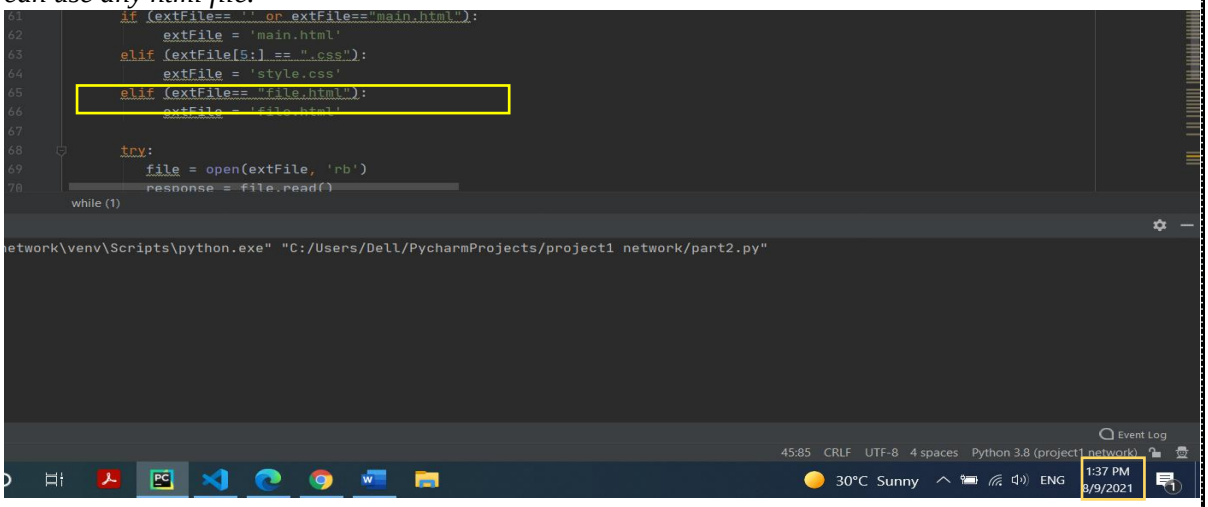




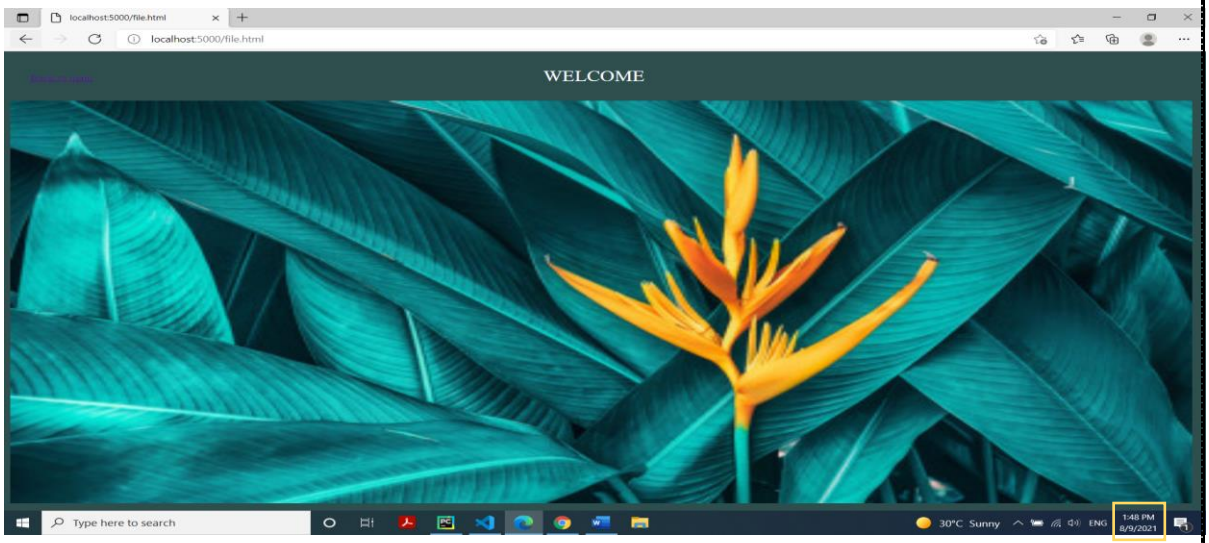
Request test from another device:



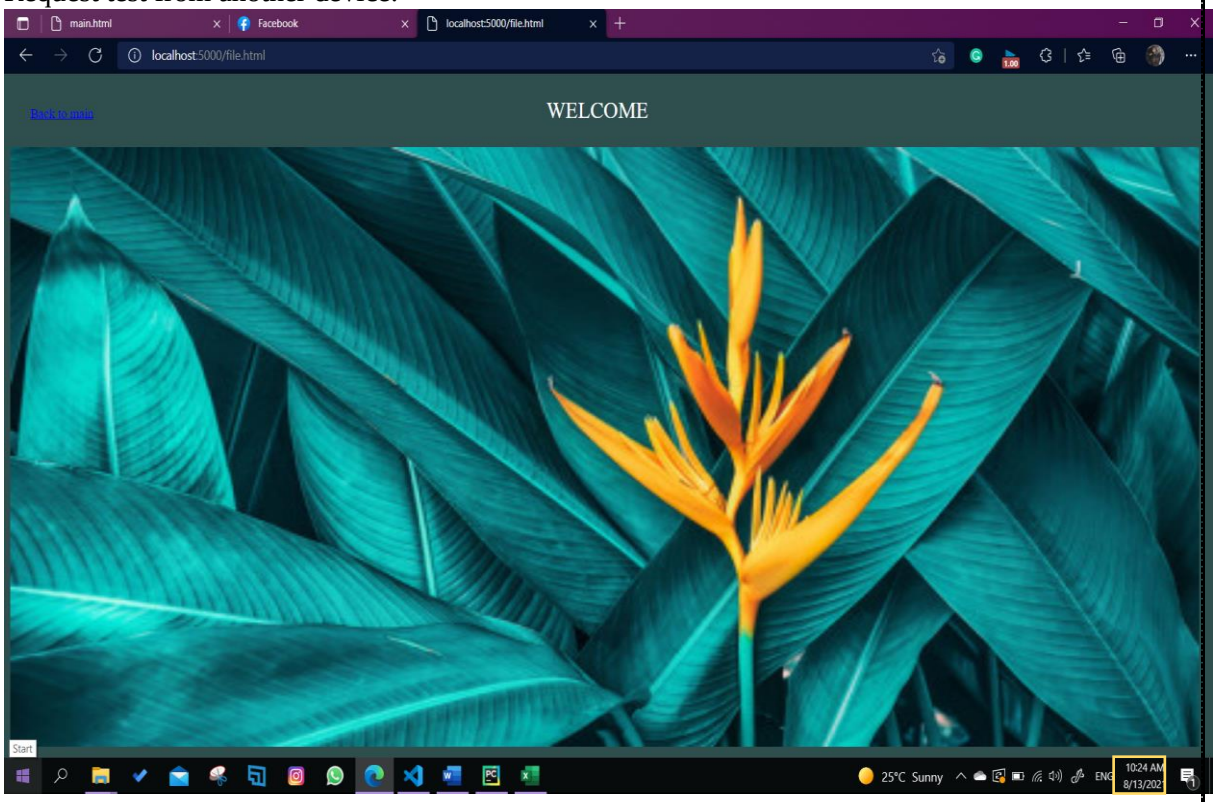
2- if the request is /file.html then the server should send html file with Content-Type: text/html. You can use any html file.



If the request is ‘file.html’, the response is:



Request test from another device:



3- if the request is /file.css then the server should send html file with Content-Type: text/html. You can use any image.

```
        if (extFile== '' or extFile=="main.html"):
            extFile = 'main.html'
        elif (extFile[5:] == ".css"):
            extFile = 'style.css'
        elif (extFile== "file.html"):
            extFile = 'file.html'

        try:
            file = open(extFile, 'rb')
            response = file.read()
while (1)
```

```
network\venv\Scripts\python.exe" "C:/Users/Dell/PycharmProjects/project1 network/part2.py"
```

If the request is ‘style.css, the response is:

```
.p1{
    color:green;
}
.p2{
    font-family: "Lucida Console", "Courier New", monospace;
    margin: 15px;
}
.p14{
    font-size: 20px;
}
.p15{
    font-size: 18px;
}
.box1 {
    width: 350px;
    height: 800px;
    margin: 15px;
    padding: 25px;
    border: 3px solid black;
    border-style: dotted;
    background-color: #333333;
    color:#f7f7f7;
  }
.p3{
    position: absolute;
    top: 100px;
    left: 65px;
    margin: 15px;
    padding: 25px;
}
.p4{
    position: absolute;
    top: 130px;
    left: 600px;
}
.p5{
    position: absolute;
    top: 130px;
    left: 1100px;
}
.p6{
    height: auto;
    width: auto;
    max-width: 300px;
    max-height: 300px;
}
.p7{
    margin: 100px;
```

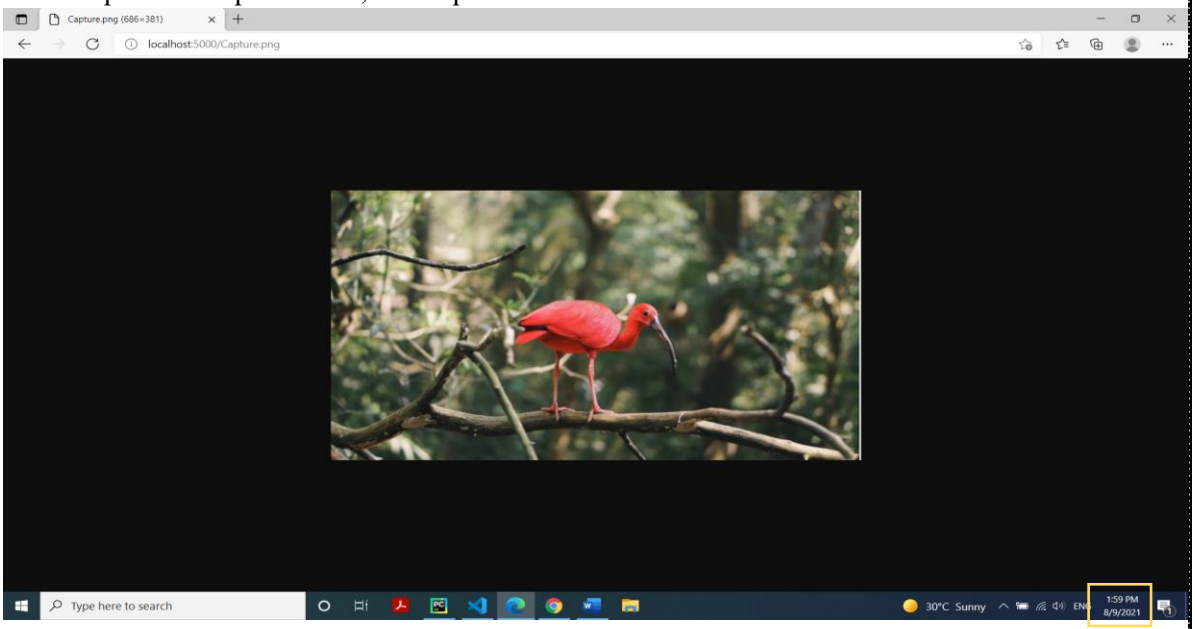
4- if the request is /imagename.png then the server should send the png image with Content-Type: image/png. You can use any image.

At first, the file will be read in try-catch format:

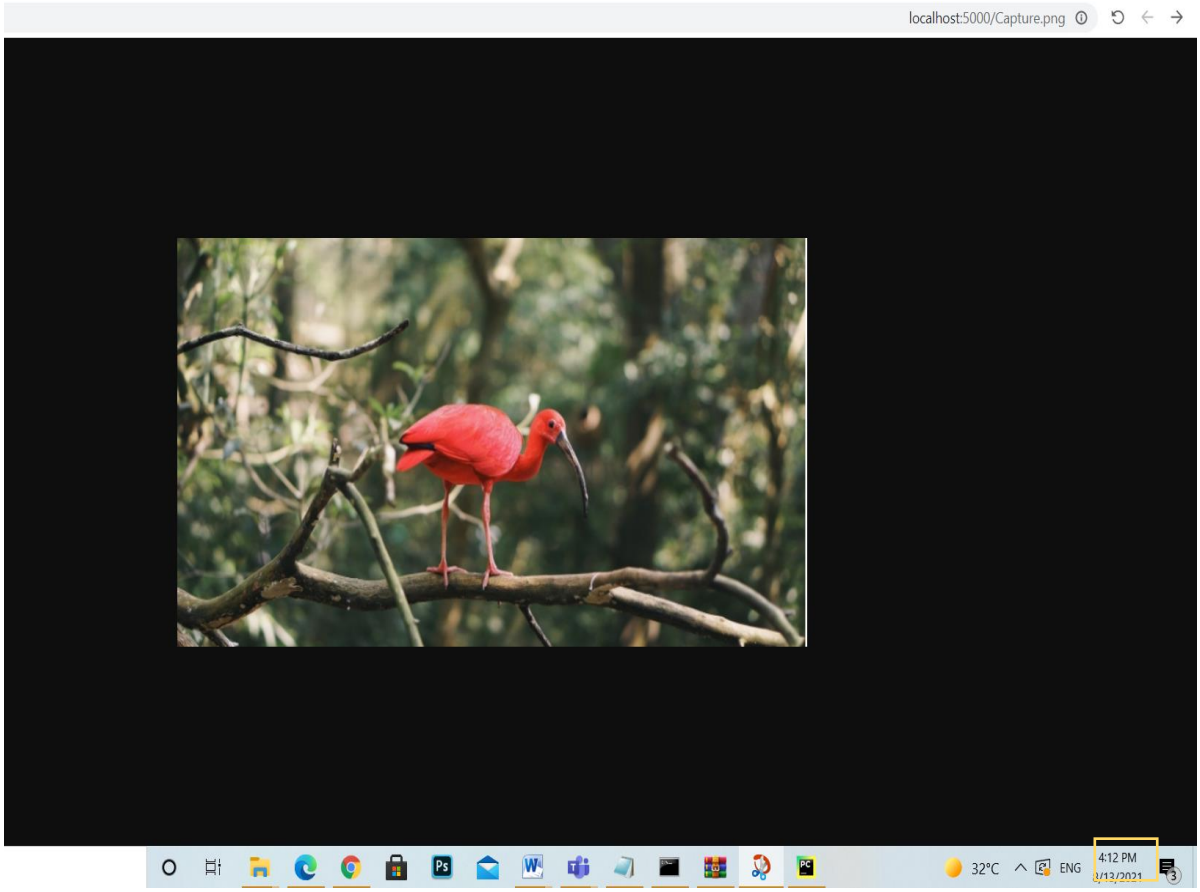
```
        extFile = 'file.html'

    try:
        file = open(extFile, 'rb')
        response = file.read()
        file.close()

        if (extFile[7:]==".png"):
            ft = 'Capture.png'
        elif (extFile[7:]==".jpg"):
            ft = 'Picture.jpg'
        elif (extFile.endswith(".csv")):
            ft = file
        else:
            ft = ''

        header_line ='HTTP/1.1 200 OK\n'+ 'Content-Type: ' + str(ft) + '\n\n'
        print(header_line)
```

```
New connection ('127.0.0.1', 49757) connected.
[Client request]:
HTTP/1.1 200 OK
Content-Type:

New connection ('127.0.0.1', 62829) connected.

Process finished with exit code -1
```

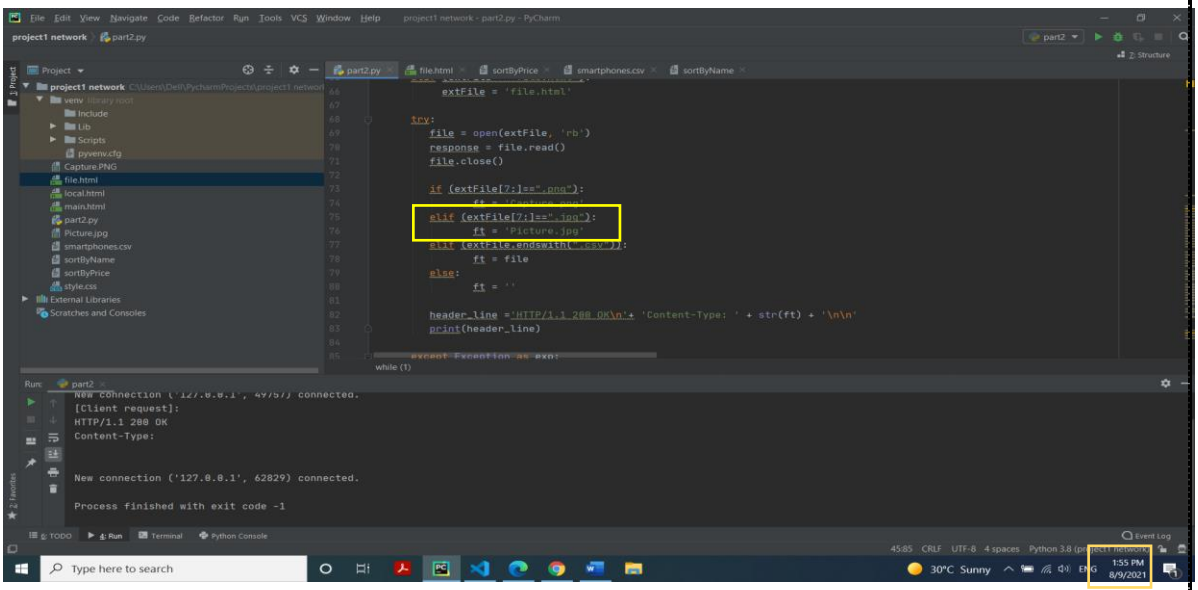
If the request is ‘Capture.PNG, the response is:



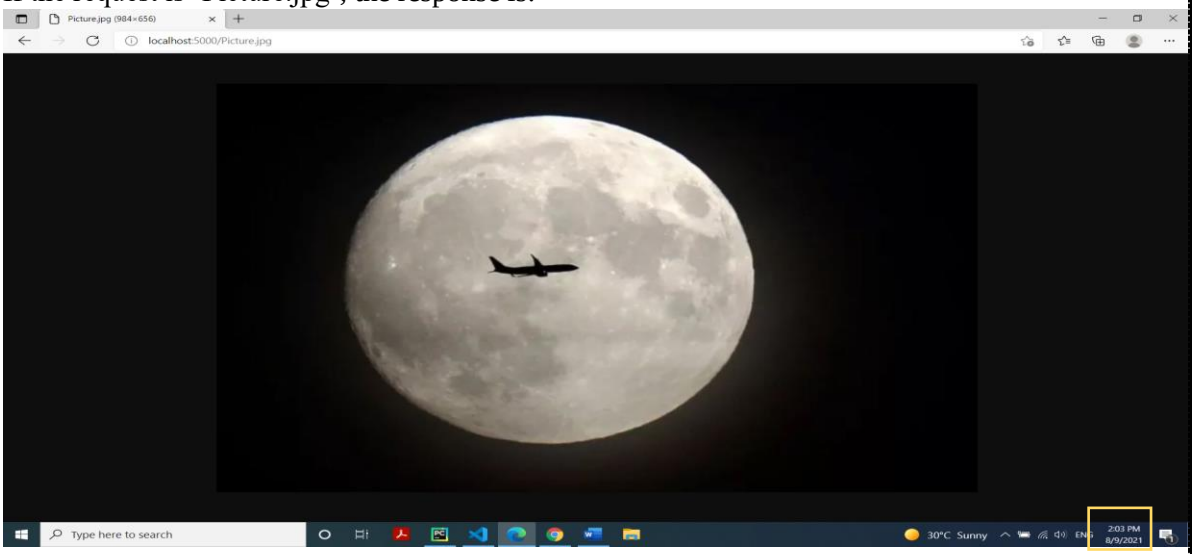
Request test from another device:



5- if the request is /imagename.jpg then the server should send the jpg image with Content-Type: image/jpeg. You can use any image.

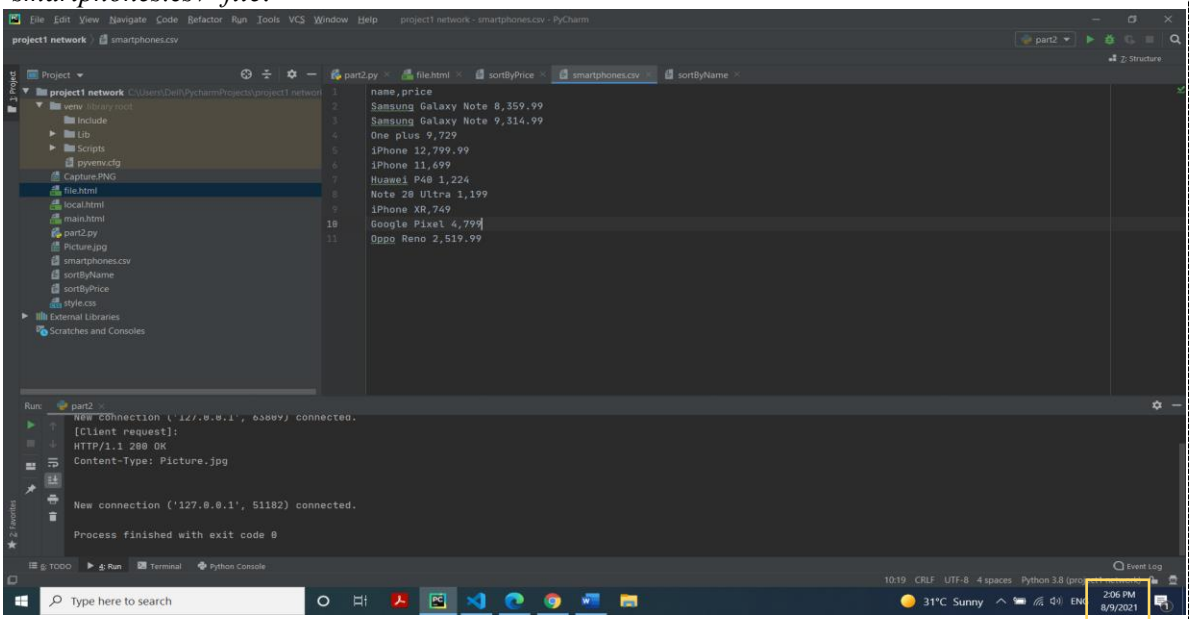


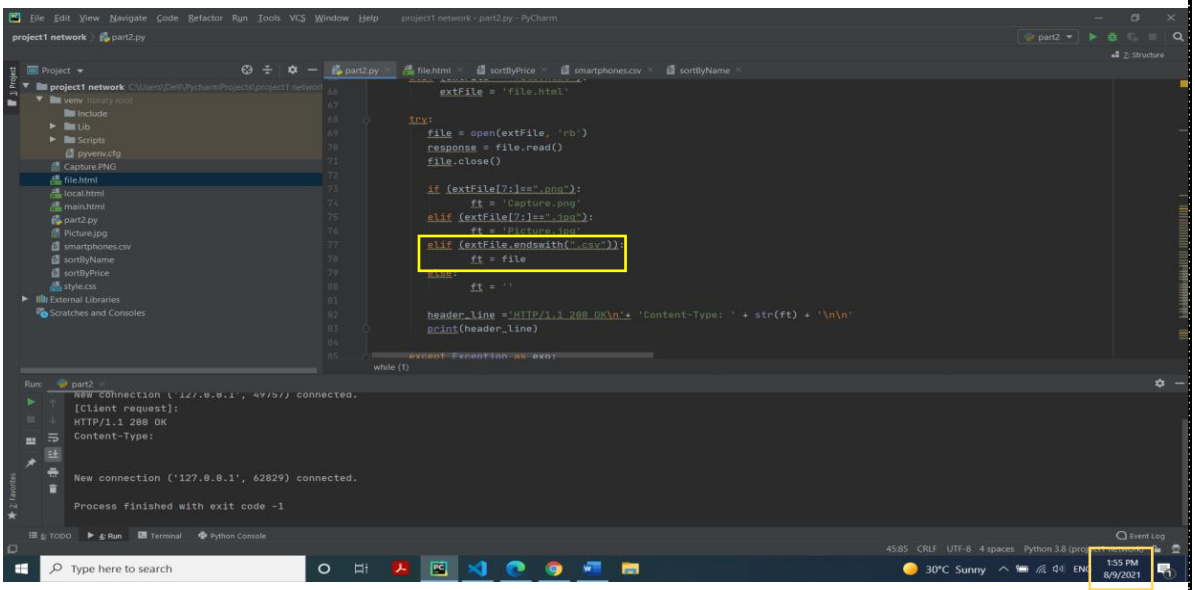
If the request is ‘Picture.jpg’, the response is:



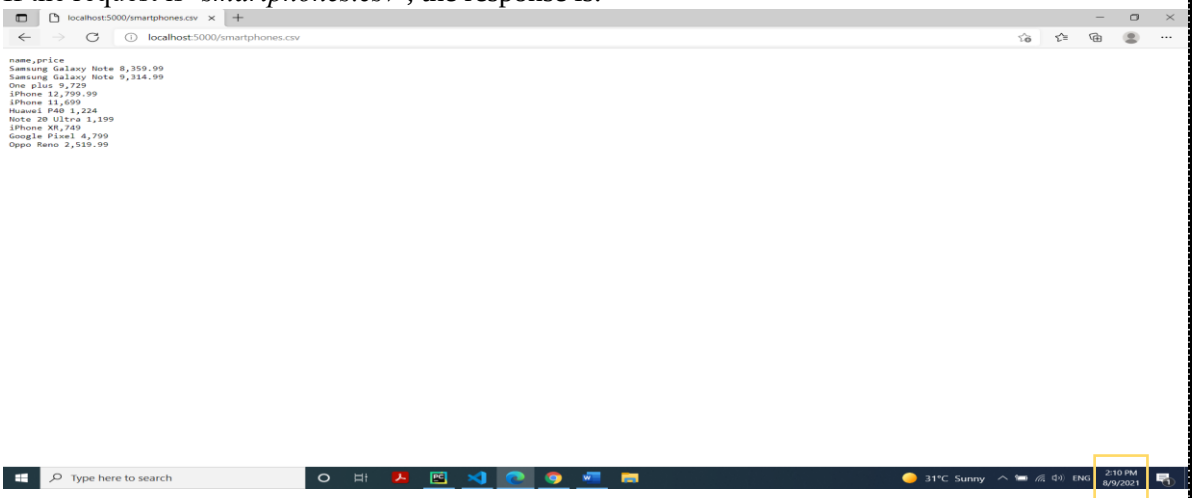
6- Include a text file (or you can use csv file) that contains names and prices of at least 10 smartphones

‘smartphones.csv’ file:



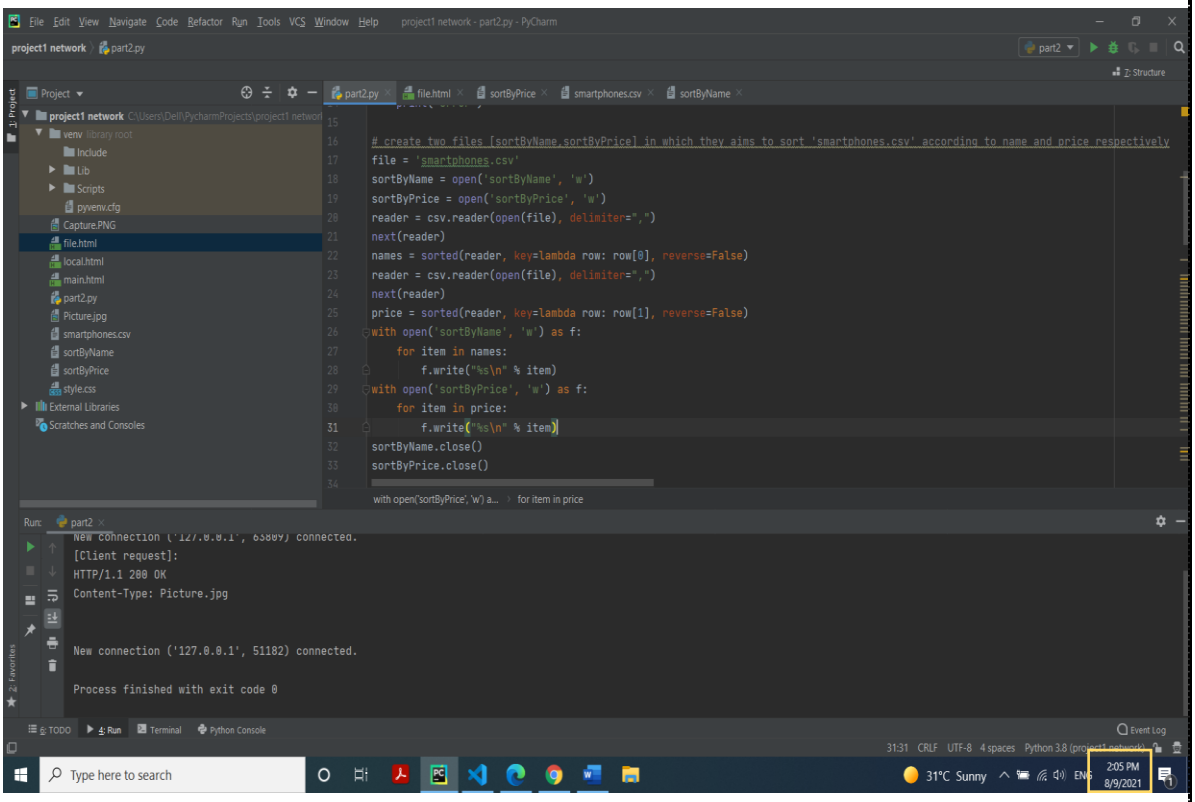


If the request is '*smartphones.csv'*, the response is:

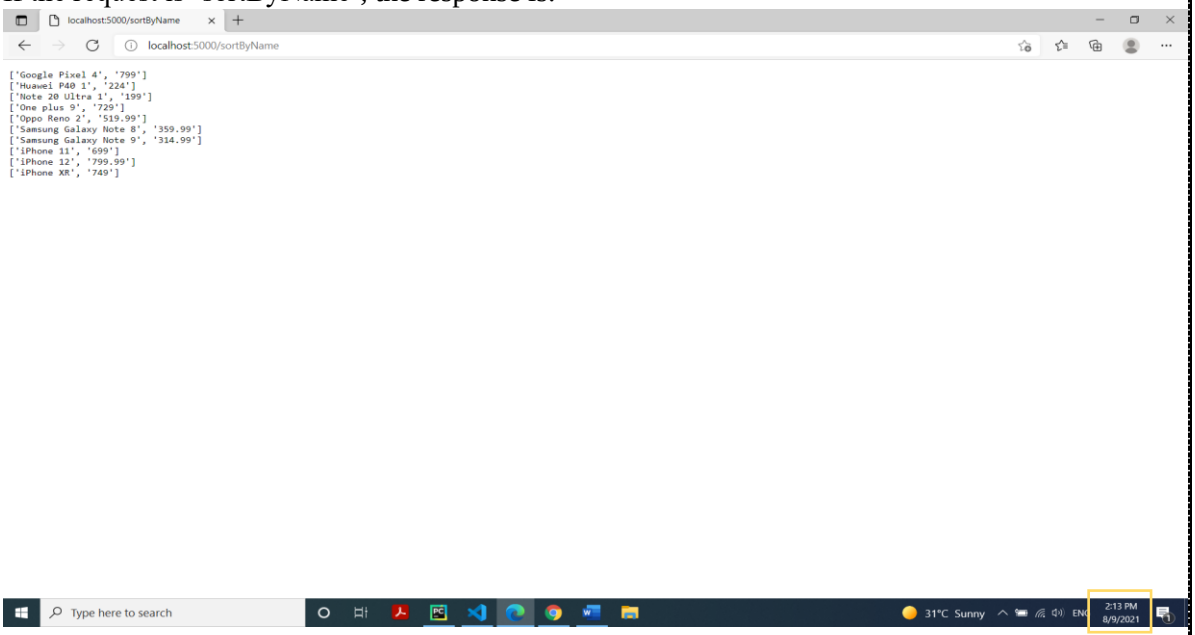


7- if the request is /SortByName then the output on the browser should be the names and prices of the smartphones sorted by the name. The server should send text page with Content-Type: text/plain. I you wish, you can use text/html to display the output in a more convenient way.

This piece of code aims to sort file according to name and save it in 'sortByName' file.

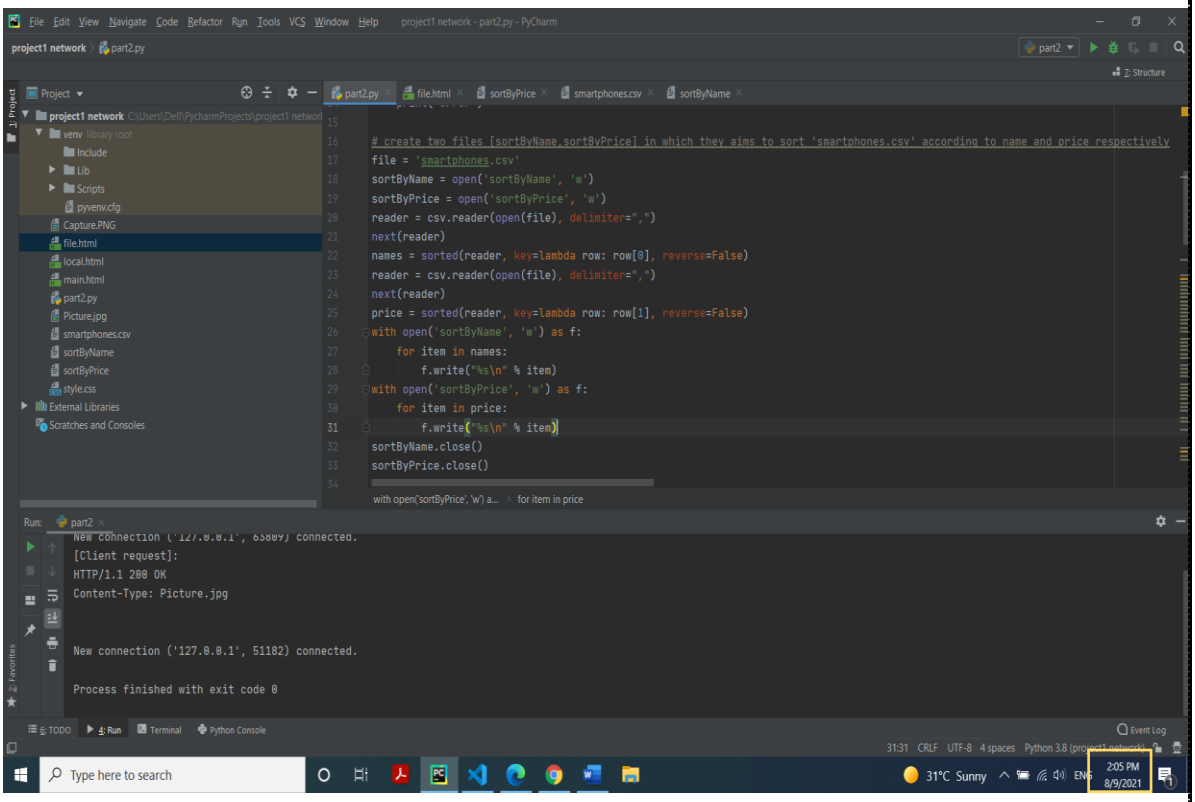


If the request is ‘sortByName’, the response is:

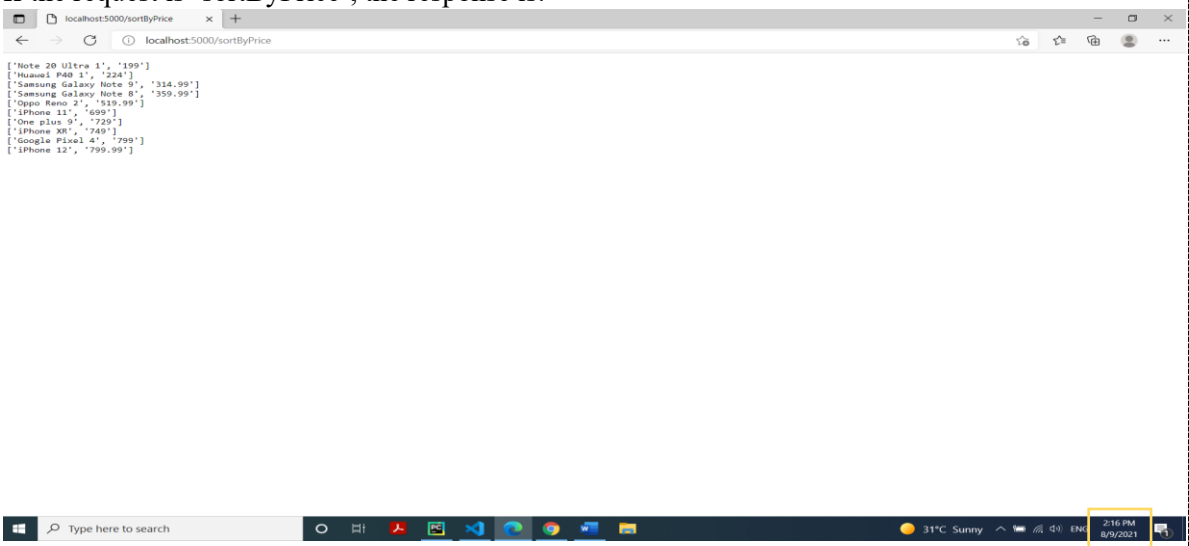


8- if the request is /SortByPrice then the output on the browser should be name and price of the smartphones sorted by its price. The server should send text page with ContentType: text/plain. I you wish, you can use text/html to display the output in a more convenient way.

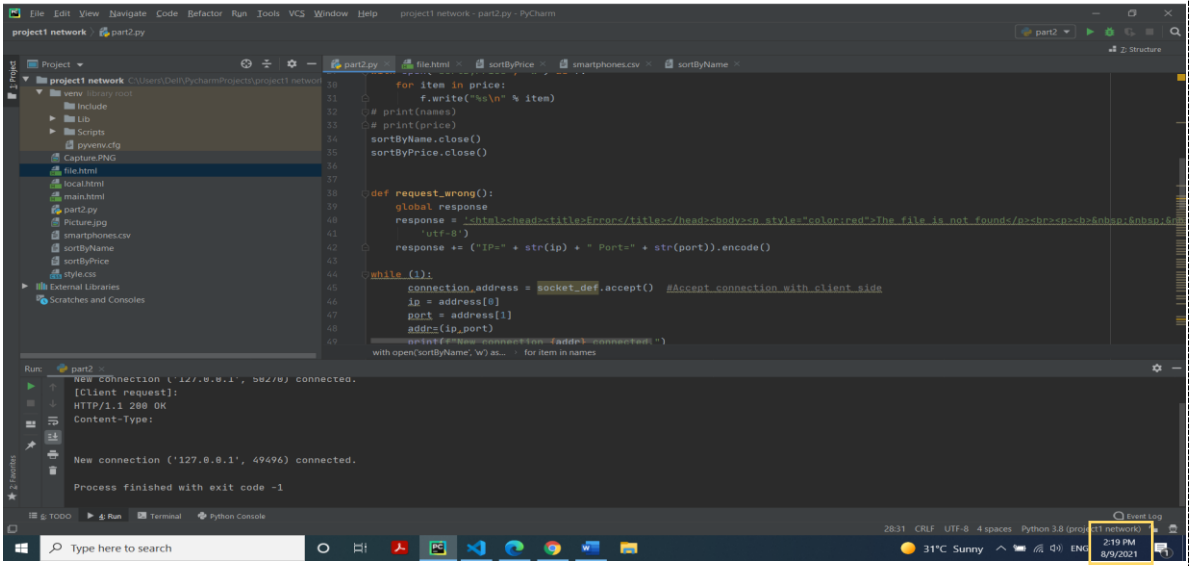
This piece of code aims to sort file according to price and save it in ‘sortByPrice’ file.



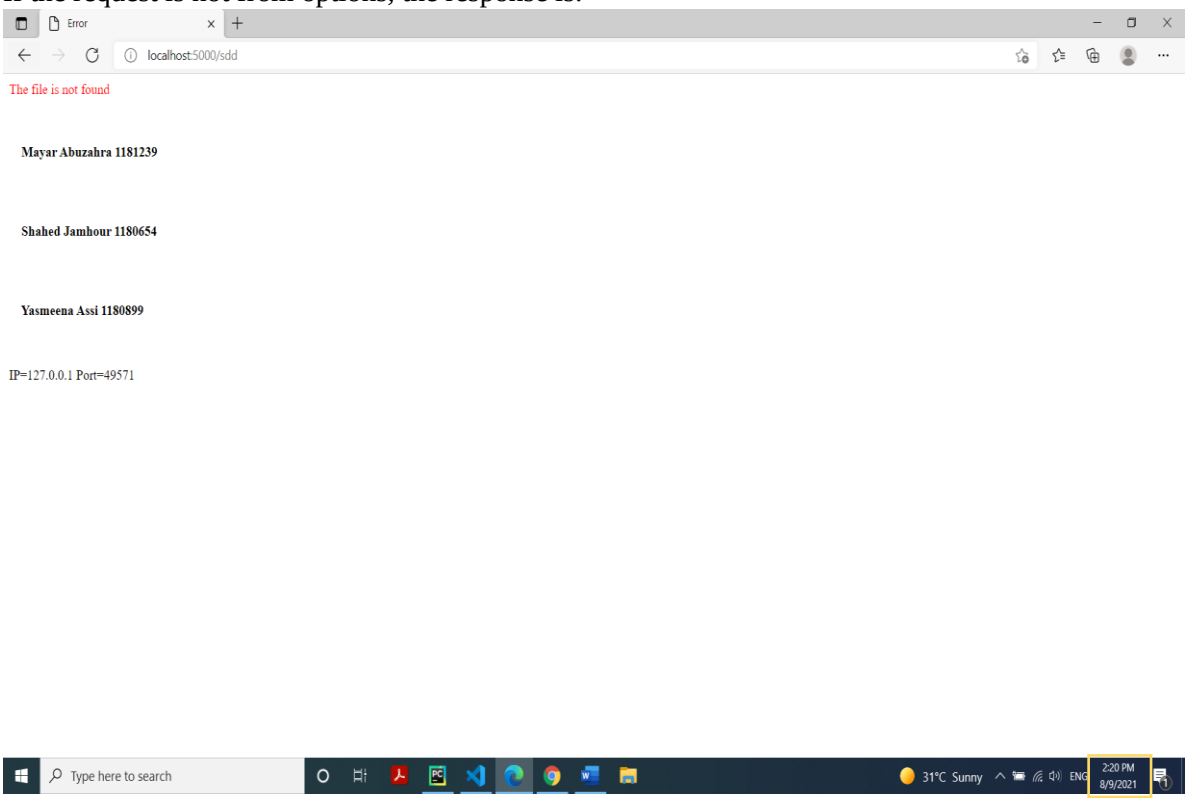
If the request is ‘sortByPrice’, the response is:



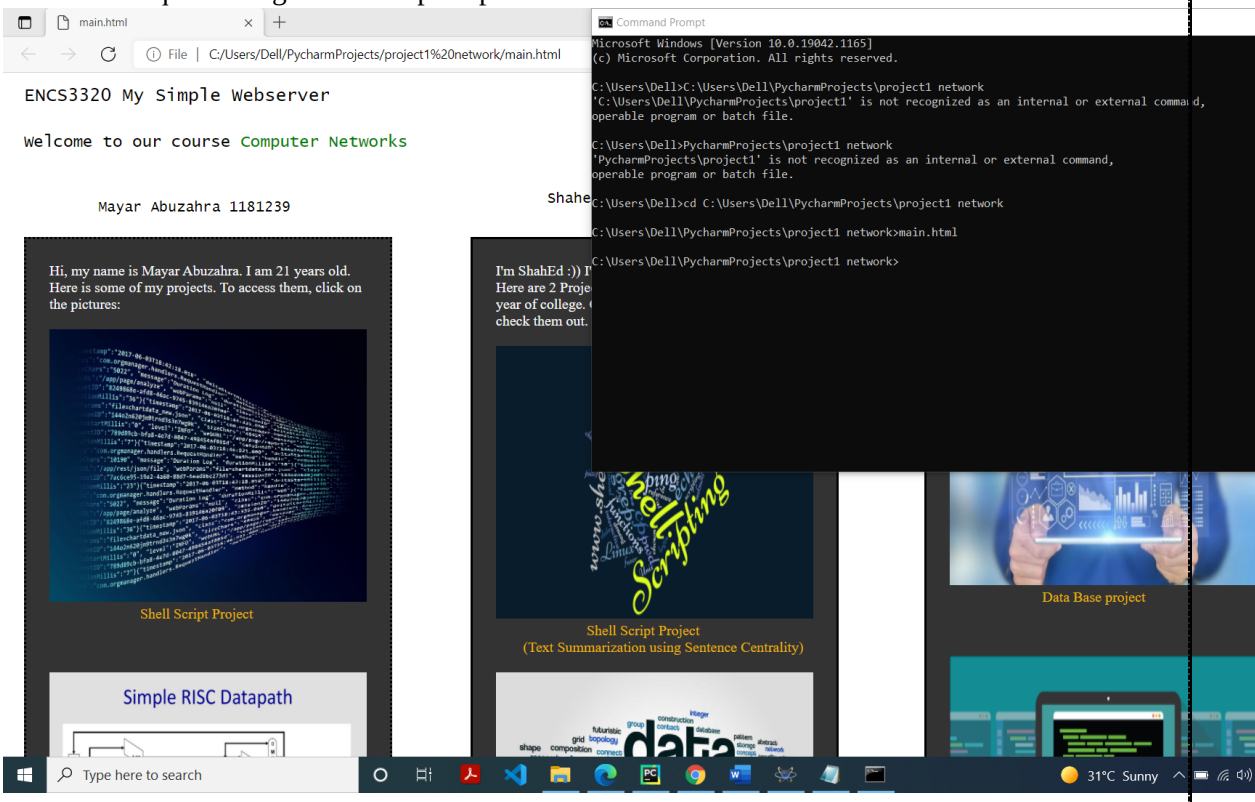
9- If the request is wrong or the file doesn’t exist the server should return a simple HTML webpage that contains (Content-Type: text/html)



If the request is not from options, the response is:



And this a request using command prompt:



❖ Conclusion

In summary, after doing this project, we have noticed how the HTTP-Web server works and how it responds to various requests by viewing the appropriate webpages and files.

On the whole, it was a beautiful project that let us realize the importance of using HTTP-Web server, and all of its objectives were obtained.