جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering & Technology
Electrical & Computer Engineering Department

ENCS3320

Project 1 Report

Socket Programming

Prepared by :

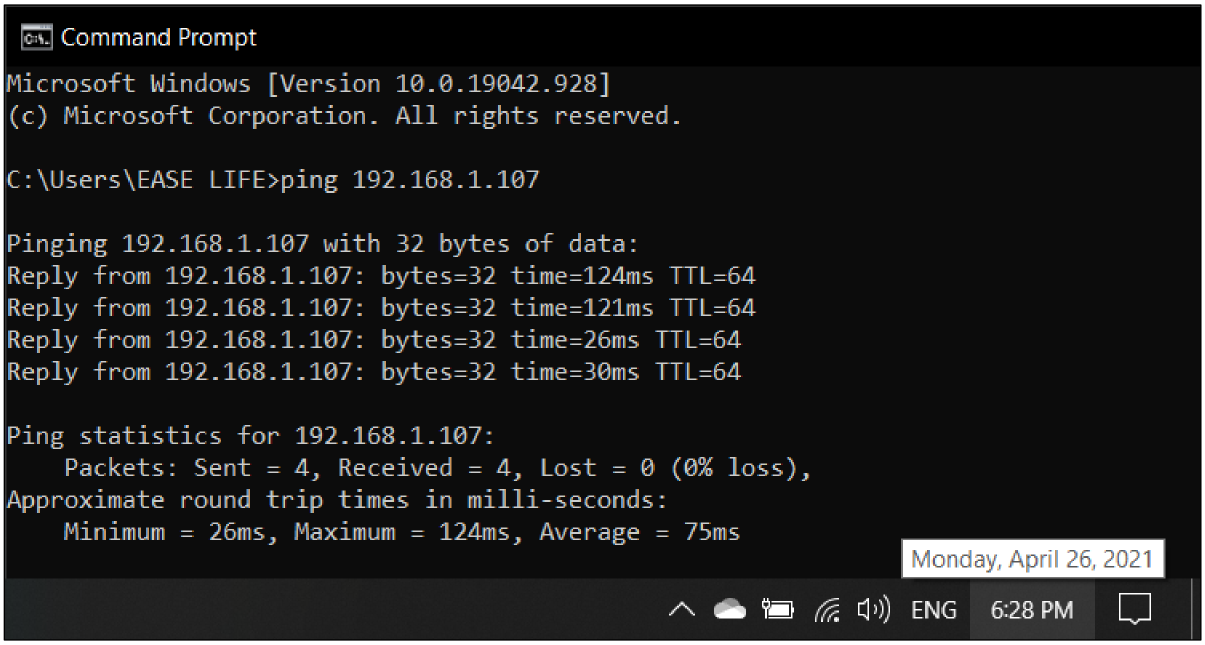
Waseem Sayara	**1182733**
Tareq Shannak	**1181404**
Manal Abuelouf	**1173359**

Date : 24/4/2021

Part I

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

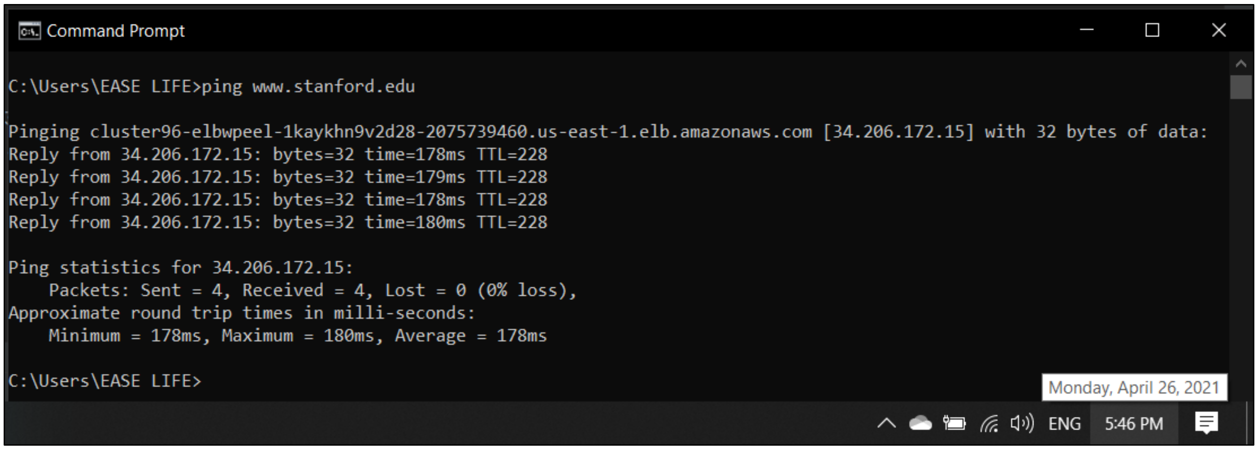
We can see from the figure below that we received a response from 192.168.1.107 when we sent 4 packets where all packets have the same TTL (time to live), all packets are received with different delays and the average is 75 ms.



```
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\EASE LIFE>ping 192.168.1.107

Pinging 192.168.1.107 with 32 bytes of data:
Reply from 192.168.1.107: bytes=32 time=124ms TTL=64
Reply from 192.168.1.107: bytes=32 time=121ms TTL=64
Reply from 192.168.1.107: bytes=32 time=26ms TTL=64
Reply from 192.168.1.107: bytes=32 time=30ms TTL=64

Ping statistics for 192.168.1.107:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 26ms, Maximum = 124ms, Average = 75ms
```

Ping www.stanford.edu

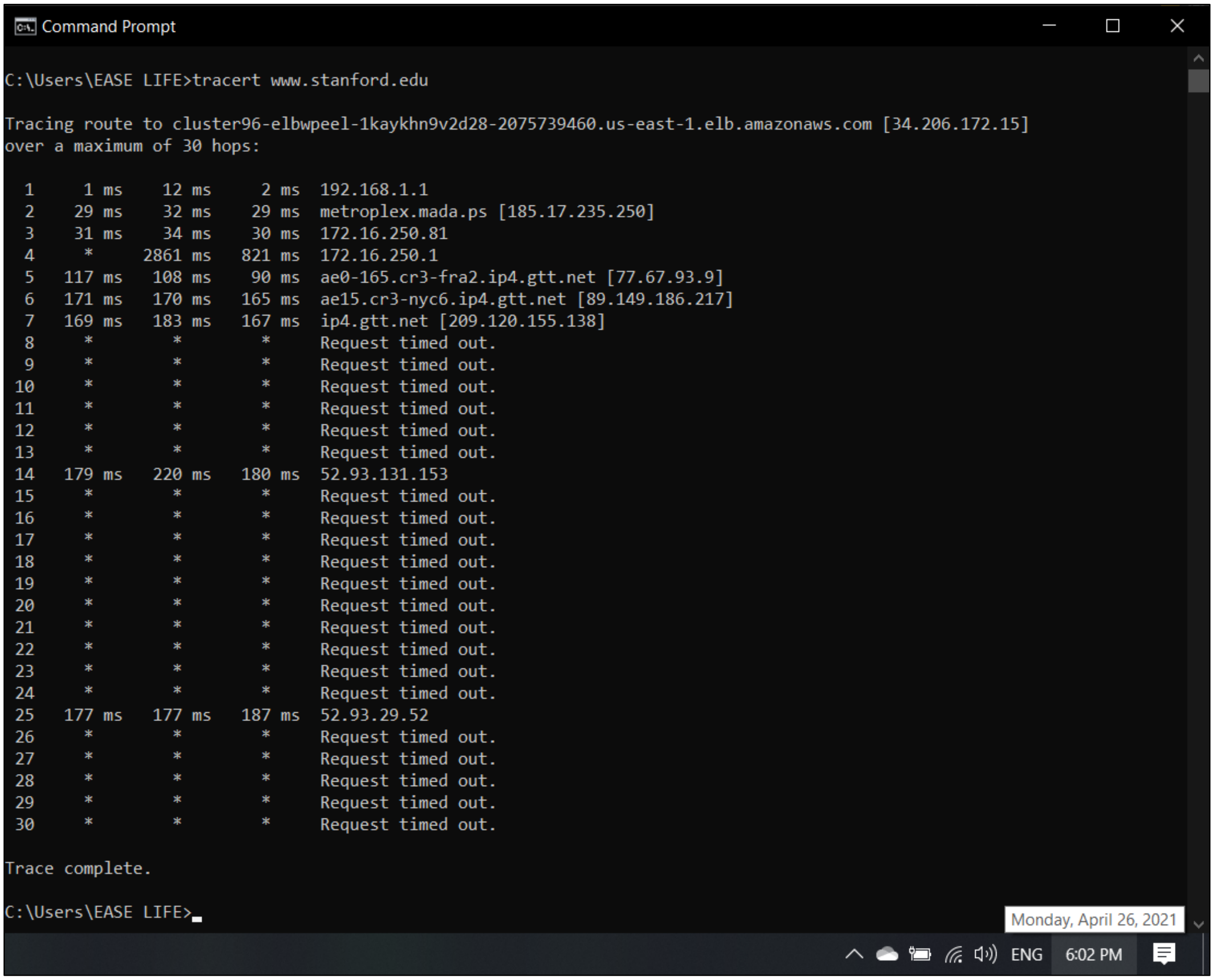
We can see from the figure below that we received a response from 34.206.172.15 when we sent 4 packets where all packets have the same TTL (time to live), all packets are received with different delays and the average is 178 ms. Also we can notice that the time to obtain a response from Stanford.edu takes more than a response from a smartphone in the same network of my laptop.



```
C:\Users\EASE LIFE>ping www.stanford.edu

Pinging cluster96-elbwpeel-1kaykhn9v2d28-2075739460.us-east-1.elb.amazonaws.com [34.206.172.15] with 32 bytes of data:
Reply from 34.206.172.15: bytes=32 time=178ms TTL=228
Reply from 34.206.172.15: bytes=32 time=179ms TTL=228
Reply from 34.206.172.15: bytes=32 time=178ms TTL=228
Reply from 34.206.172.15: bytes=32 time=180ms TTL=228

Ping statistics for 34.206.172.15:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 178ms, Maximum = 180ms, Average = 178ms

C:\Users\EASE LIFE>
```

Tracert www.stanford.edu

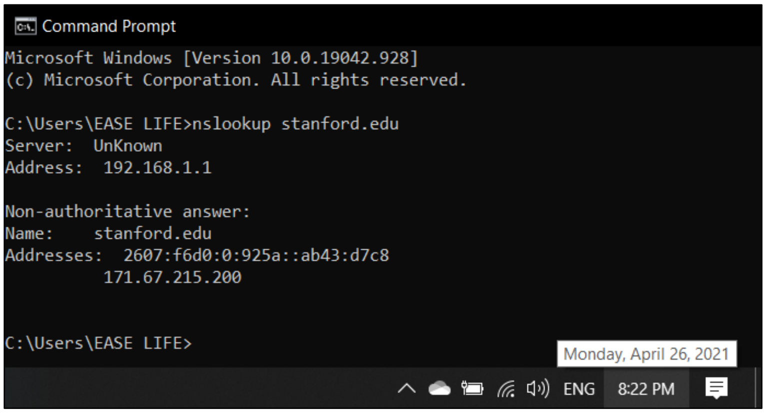
This command sends 3 messages for every router and waits the response from the router, it continues in this process until it reaches the chosen IP. We can see from the figure that there is 3 measurements from each router. When we go down in the lines, we will see the 3 measurements increases because the next router go further. When the request timed out and the measurements are *, then the packet is prevented from reaching the destination because there is a problem at that location or the route is incorrect.



```
C:\Users\EASE LIFE>tracert www.stanford.edu

Tracing route to cluster96-elbwpeel-1kaykhn9v2d28-2075739460.us-east-1.elb.amazonaws.com [34.206.172.15]
over a maximum of 30 hops:

  1     1 ms    12 ms     2 ms  192.168.1.1
  2    29 ms    32 ms    29 ms  metroplex.mada.ps [185.17.235.250]
  3    31 ms    34 ms    30 ms  172.16.250.81
  4     *     2861 ms   821 ms  172.16.250.1
  5   117 ms   108 ms    90 ms  ae0-165.cr3-fra2.ip4.gtt.net [77.67.93.9]
  6   171 ms   170 ms   165 ms  ae15.cr3-nyc6.ip4.gtt.net [89.149.186.217]
  7   169 ms   183 ms   167 ms  ip4.gtt.net [209.120.155.138]
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14   179 ms   220 ms   180 ms  52.93.131.153
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23     *        *        *     Request timed out.
 24     *        *        *     Request timed out.
 25   177 ms   177 ms   187 ms  52.93.29.52
 26     *        *        *     Request timed out.
 27     *        *        *     Request timed out.
 28     *        *        *     Request timed out.
 29     *        *        *     Request timed out.
 30     *        *        *     Request timed out.

Trace complete.

C:\Users\EASE LIFE>
```

Nslookup www.stanford.edu

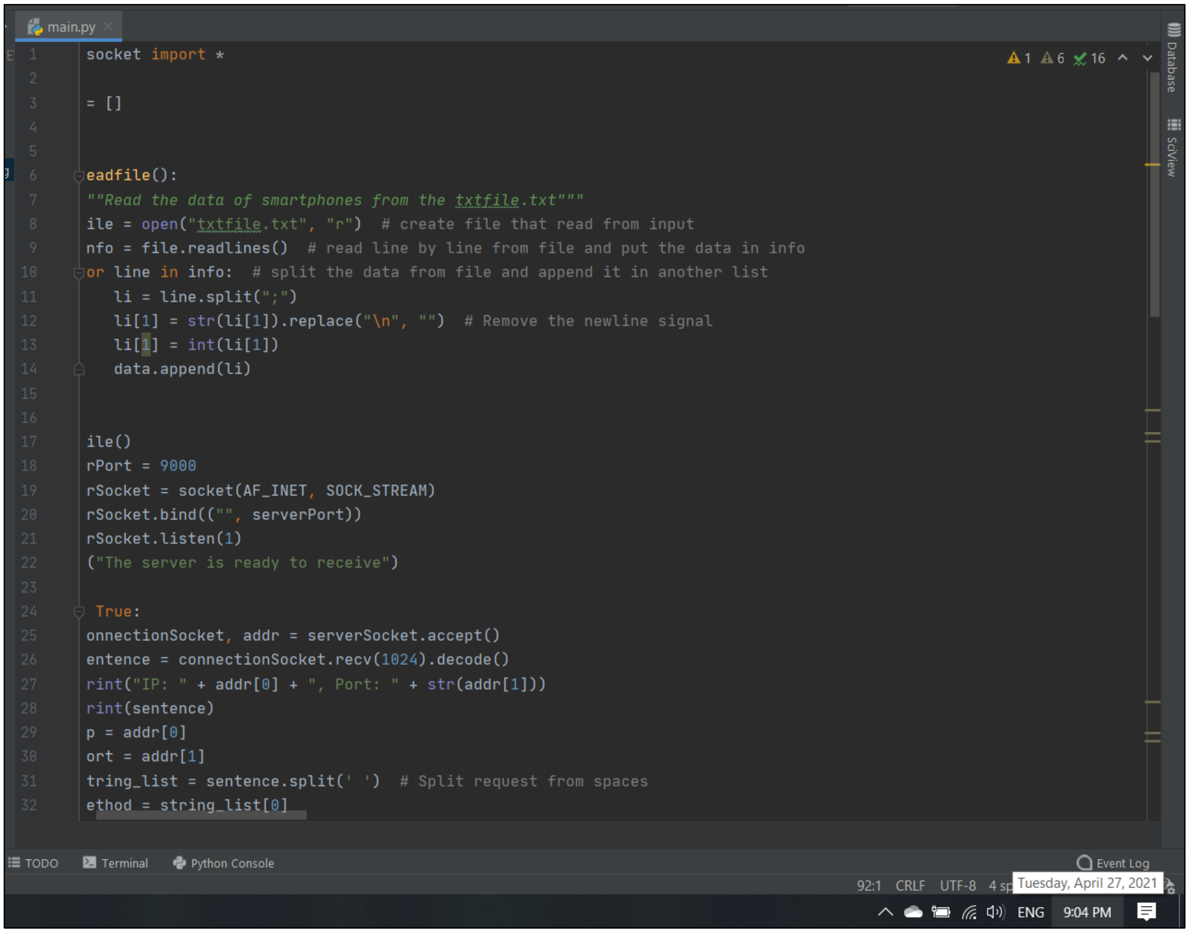
We can see that it prints the IP address corresponding to the host which is my laptop's IP address, and prints the name and addresses of the server which is the host that we sent a probe.



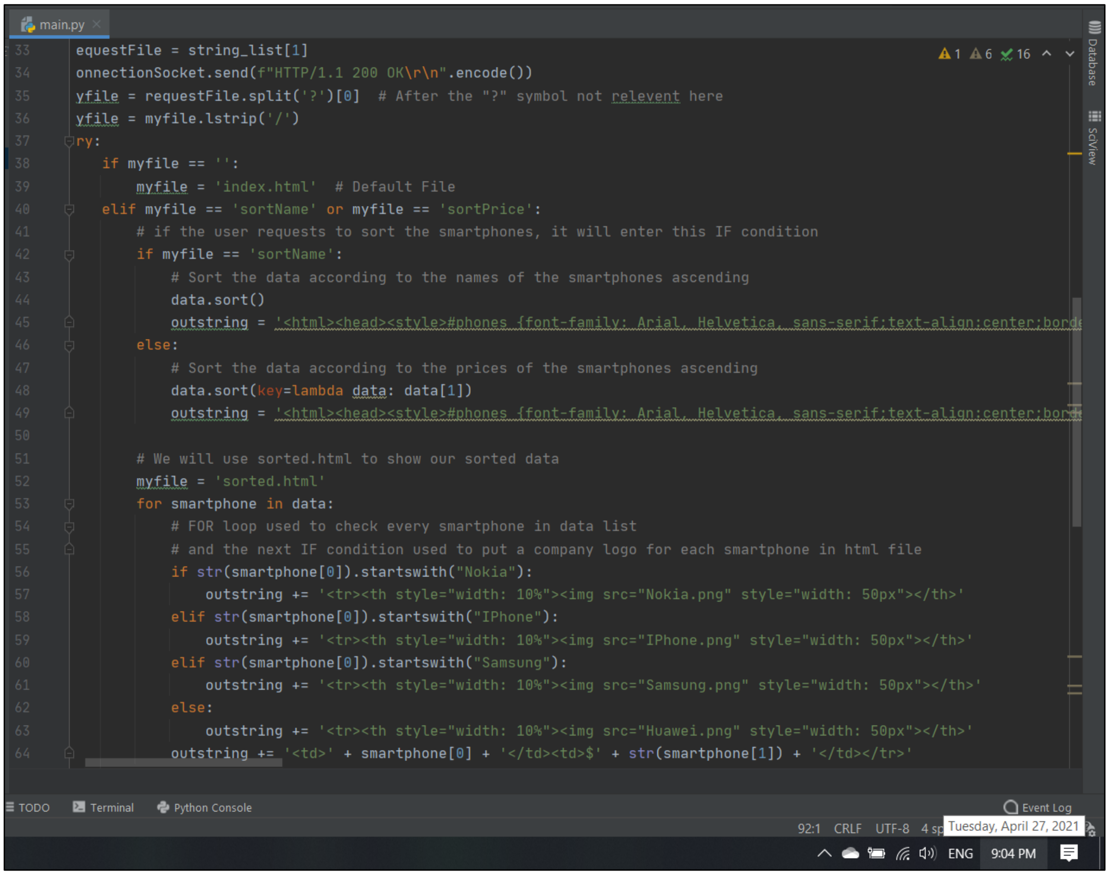
```
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\EASE LIFE>nslookup stanford.edu
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:    stanford.edu
Addresses:  2607:f6d0:0:925a::ab43:d7c8
          171.67.215.200


C:\Users\EASE LIFE>
```

Part II

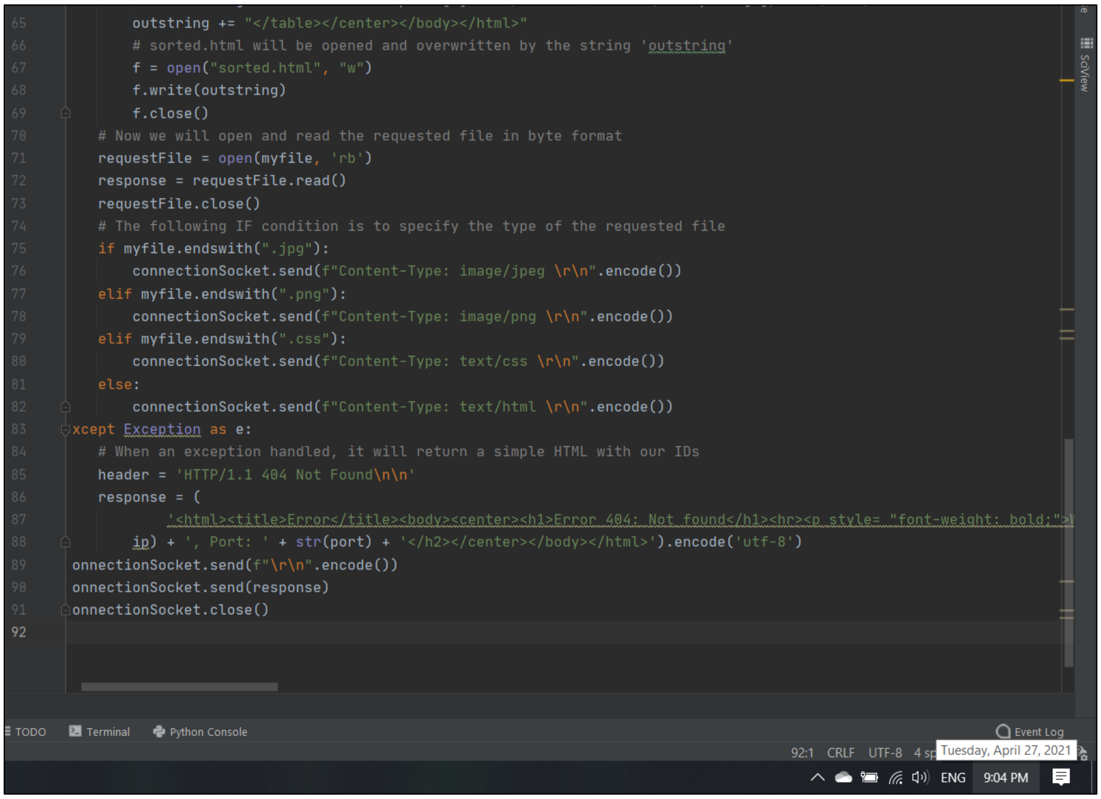
Code with comments



```
socket import *

= []


eadfile():
""Read the data of smartphones from the txtfile.txt"""
ile = open("txtfile.txt", "r")  # create file that read from input
nfo = file.readlines()  # read line by line from file and put the data in info
or line in info:  # split the data from file and append it in another list
   li = line.split(";")
   li[1] = str(li[1]).replace("\n", "")  # Remove the newline signal
   li[1] = int(li[1])
   data.append(li)


ile()
rPort = 9000
rSocket = socket(AF_INET, SOCK_STREAM)
rSocket.bind(("", serverPort))
rSocket.listen(1)
("The server is ready to receive")

 True:
onnectionSocket, addr = serverSocket.accept()
entence = connectionSocket.recv(1024).decode()
rint("IP: " + addr[0] + ", Port: " + str(addr[1]))
rint(sentence)
p = addr[0]
ort = addr[1]
tring_list = sentence.split(' ')  # Split request from spaces
ethod = string_list[0]
```

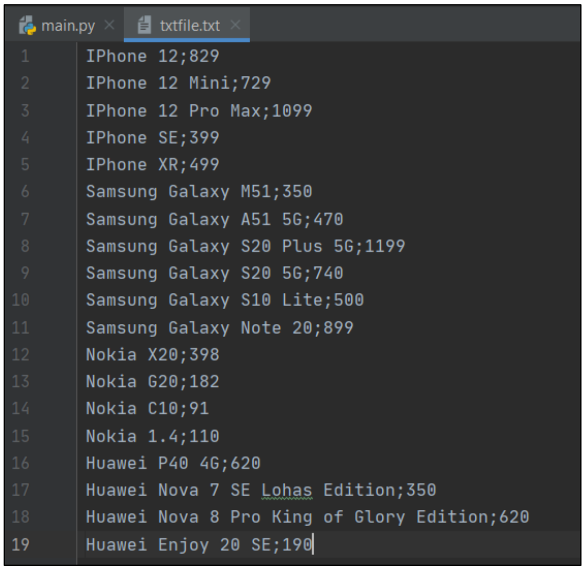


```
equestFile = string_list[1]
onnectionSocket.send(f"HTTP/1.1 200 OK\r\n".encode())
yfile = requestFile.split('?')[0]  # After the "?" symbol not relevent here
yfile = myfile.lstrip('/')
ry:
   if myfile == '':
       myfile = 'index.html'  # Default File
   elif myfile == 'sortName' or myfile == 'sortPrice':
       # if the user requests to sort the smartphones, it will enter this IF condition
       if myfile == 'sortName':
           # Sort the data according to the names of the smartphones ascending
           data.sort()
           outstring = '<html><head><style>#phones {font-family: Arial, Helvetica, sans-serif;text-align:center;borde
       else:
           # Sort the data according to the prices of the smartphones ascending
           data.sort(key=lambda data: data[1])
           outstring = '<html><head><style>#phones {font-family: Arial, Helvetica, sans-serif;text-align:center;borde

       # We will use sorted.html to show our sorted data
       myfile = 'sorted.html'
       for smartphone in data:
           # FOR loop used to check every smartphone in data list
           # and the next IF condition used to put a company logo for each smartphone in html file
           if str(smartphone[0]).startswith("Nokia"):
               outstring += '<tr><th style="width: 10%"><img src="Nokia.png" style="width: 50px"></th>'
           elif str(smartphone[0]).startswith("IPhone"):
               outstring += '<tr><th style="width: 10%"><img src="IPhone.png" style="width: 50px"></th>'
           elif str(smartphone[0]).startswith("Samsung"):
               outstring += '<tr><th style="width: 10%"><img src="Samsung.png" style="width: 50px"></th>'
           else:
               outstring += '<tr><th style="width: 10%"><img src="Huawei.png" style="width: 50px"></th>'
           outstring += '<td>' + smartphone[0] + '</td><td>$' + str(smartphone[1]) + '</td></tr>'
```



```
       outstring += "</table></center></body></html>"
       # sorted.html will be opened and overwritten by the string 'outstring'
       f = open("sorted.html", "w")
       f.write(outstring)
       f.close()
   # Now we will open and read the requested file in byte format
   requestFile = open(myfile, 'rb')
   response = requestFile.read()
   requestFile.close()
   # The following IF condition is to specify the type of the requested file
   if myfile.endswith(".jpg"):
       connectionSocket.send(f"Content-Type: image/jpeg \r\n".encode())
   elif myfile.endswith(".png"):
       connectionSocket.send(f"Content-Type: image/png \r\n".encode())
   elif myfile.endswith(".css"):
       connectionSocket.send(f"Content-Type: text/css \r\n".encode())
   else:
       connectionSocket.send(f"Content-Type: text/html \r\n".encode())
xcept Exception as e:
   # When an exception handled, it will return a simple HTML with our IDs
   header = 'HTTP/1.1 404 Not Found\n\n'
   response = (
           '<html><title>Error</title><body><center><h1>Error 404: Not found</h1><hr><p style= "font-weight: bold;">
       ip) + ', Port: ' + str(port) + '</h2></center></body></html>').encode('utf-8')
onnectionSocket.send(f"\r\n".encode())
onnectionSocket.send(response)
onnectionSocket.close()
```

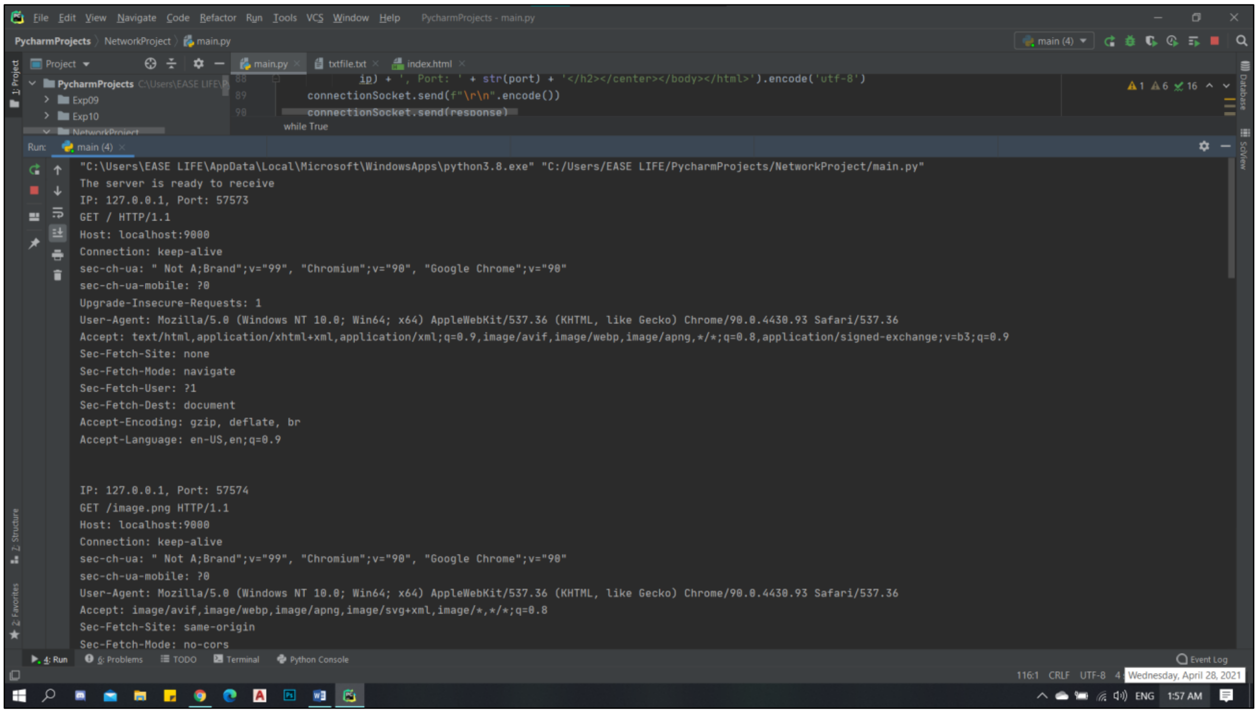
This is the text file that contains the names of the smartphones with their prices.



```
IPhone 12;829
IPhone 12 Mini;729
IPhone 12 Pro Max;1099
IPhone SE;399
IPhone XR;499
Samsung Galaxy M51;350
Samsung Galaxy A51 5G;470
Samsung Galaxy S20 Plus 5G;1199
Samsung Galaxy S20 5G;740
Samsung Galaxy S10 Lite;500
Samsung Galaxy Note 20;899
Nokia X20;398
Nokia G20;182
Nokia C10;91
Nokia 1.4;110
Huawei P40 4G;620
Huawei Nova 7 SE Lohas Edition;350
Huawei Nova 8 Pro King of Glory Edition;620
Huawei Enjoy 20 SE;190
```

Screenshot of the HTTP request printed on the command line

When we ran the code and open localhost:9000 in the browser, we obtained the next output on the command line which is the request of the index.html, followed by 4 image requests for index.html.

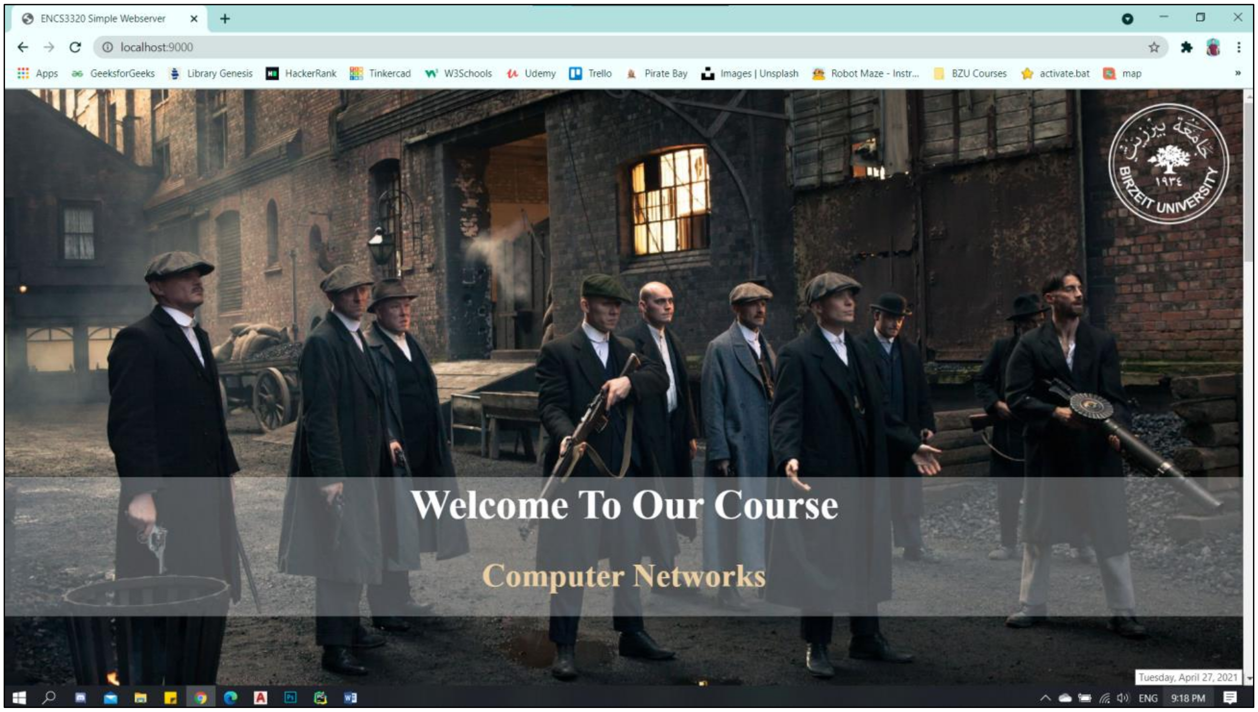

File Edit View Navigate Code Refactor Run Tools VCS Window Help
PycharmProjects - main.py
PycharmProjects 〉 NetworkProject 〉 main.py
Project
PycharmProjects C:\Users\EASE LIFE\P
Exp09
Exp10
NetworkProject
main.py
txtfile.txt
index.html
Run:
main (4)
4: Run
6: Problems
TODO
Terminal
Python Console
Event Log
116:1 CRLF UTF-8 4
Wednesday, April 28, 2021
ENG
1:57 AM

```
                ip) + ', Port: ' + str(port) + '</h2></center></body></html>').encode('utf-8')
        connectionSocket.send(f"\r\n".encode())
        connectionSocket.send(response)
    while True
```

```
"C:\Users\EASE LIFE\AppData\Local\Microsoft\WindowsApps\python3.8.exe" "C:/Users/EASE LIFE/PycharmProjects/NetworkProject/main.py"
The server is ready to receive
IP: 127.0.0.1, Port: 57573
GET / HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="90", "Google Chrome";v="90"
sec-ch-ua-mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 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: en-US,en;q=0.9


IP: 127.0.0.1, Port: 57574
GET /image.png HTTP/1.1
Host: localhost:9000
Connection: keep-alive
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="90", "Google Chrome";v="90"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 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
```

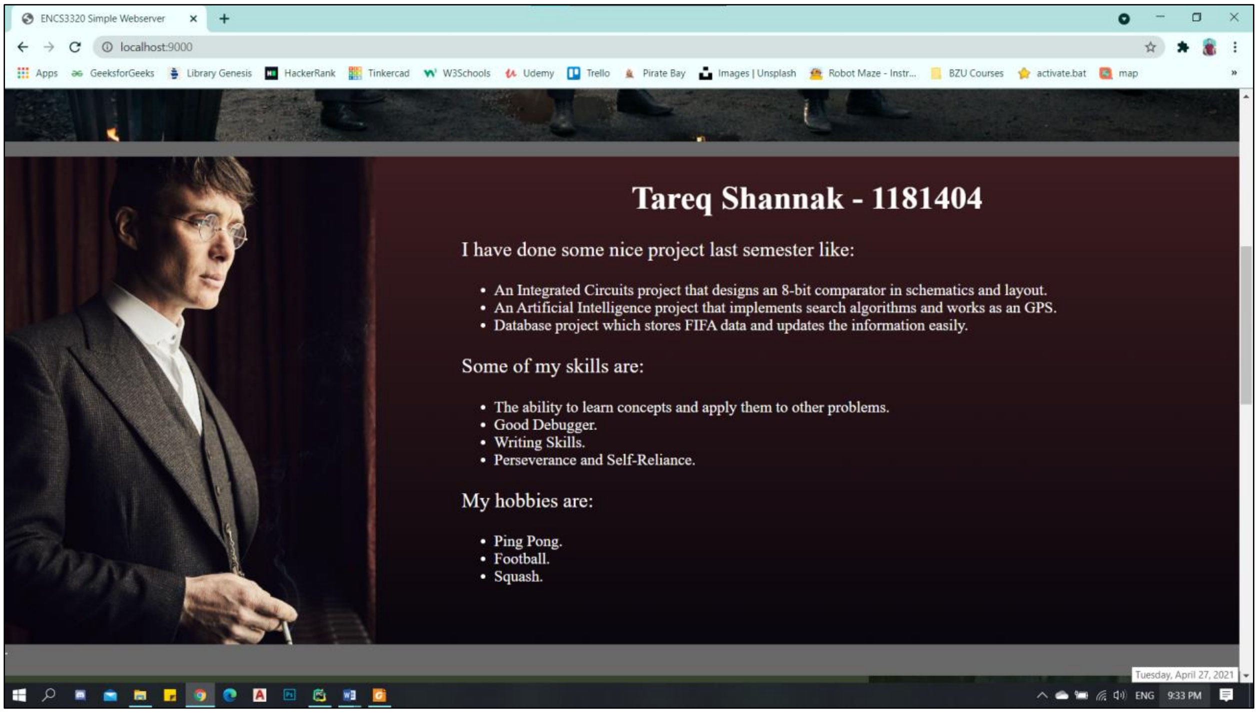
Screenshots in the browser for all requests

Localhost:9000 or Localhost:9000/index.html



Welcome To Our Course

Computer Networks



Tareq Shannak - 1181404

I have done some nice project last semester like:

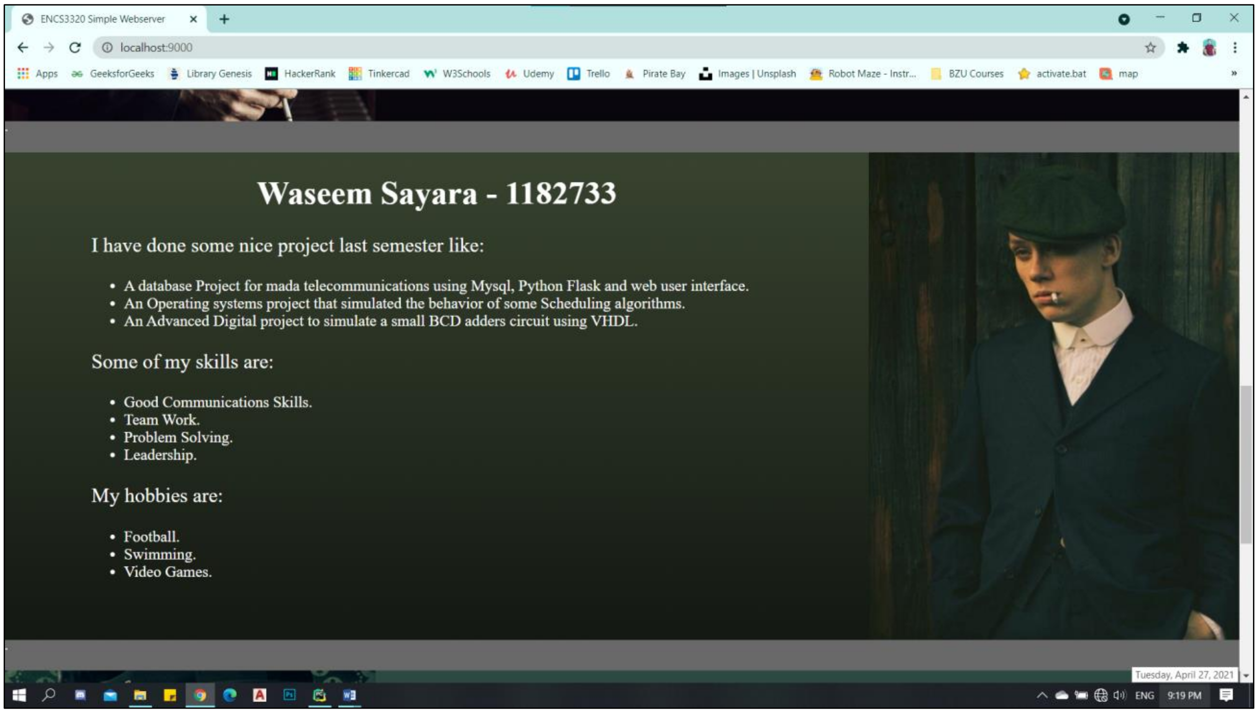
- An Integrated Circuits project that designs an 8-bit comparator in schematics and layout.
- An Artificial Intelligence project that implements search algorithms and works as an GPS.
- Database project which stores FIFA data and updates the information easily.

Some of my skills are:

- The ability to learn concepts and apply them to other problems.
- Good Debugger.
- Writing Skills.
- Perseverance and Self-Reliance.

My hobbies are:

- Ping Pong.
- Football.
- Squash.



Waseem Sayara - 1182733

I have done some nice project last semester like:

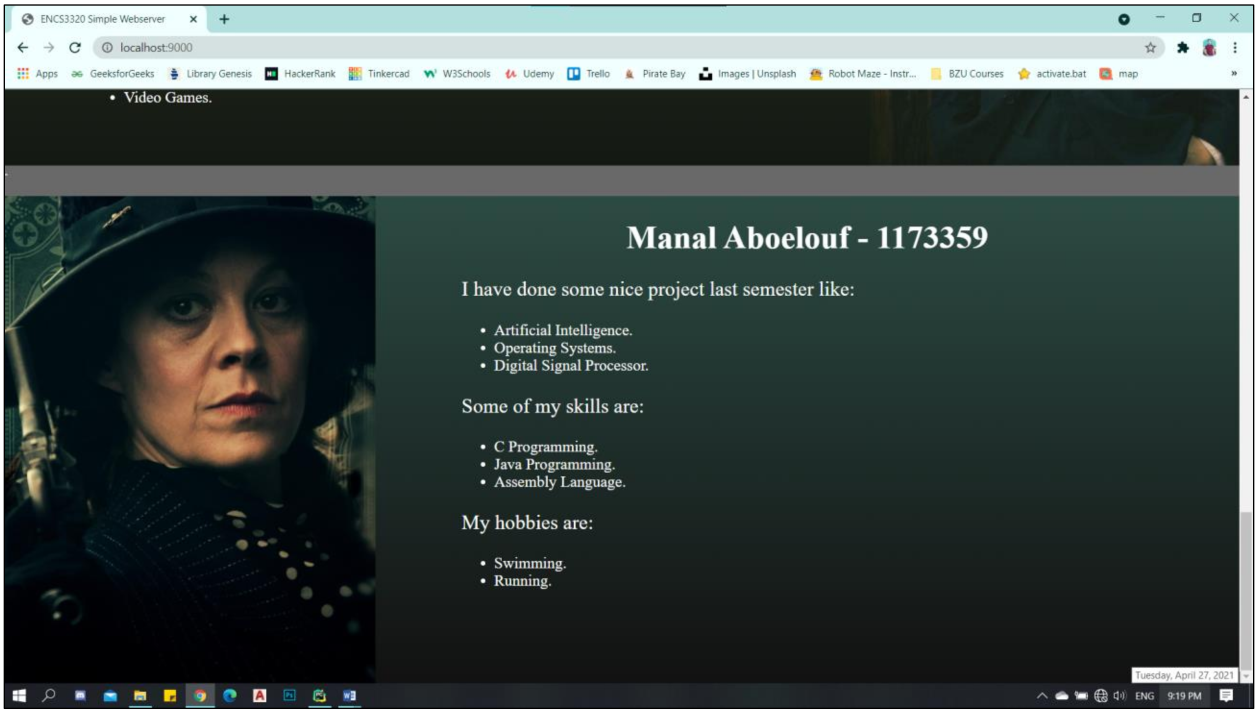
- A database Project for mada telecommunications using Mysql, Python Flask and web user interface.
- An Operating systems project that simulated the behavior of some Scheduling algorithms.
- An Advanced Digital project to simulate a small BCD adders circuit using VHDL.

Some of my skills are:

- Good Communications Skills.
- Team Work.
- Problem Solving.
- Leadership.

My hobbies are:

- Football.
- Swimming.
- Video Games.



- Video Games.

Manal Aboelouf - 1173359

I have done some nice project last semester like:

- Artificial Intelligence.
- Operating Systems.
- Digital Signal Processor.

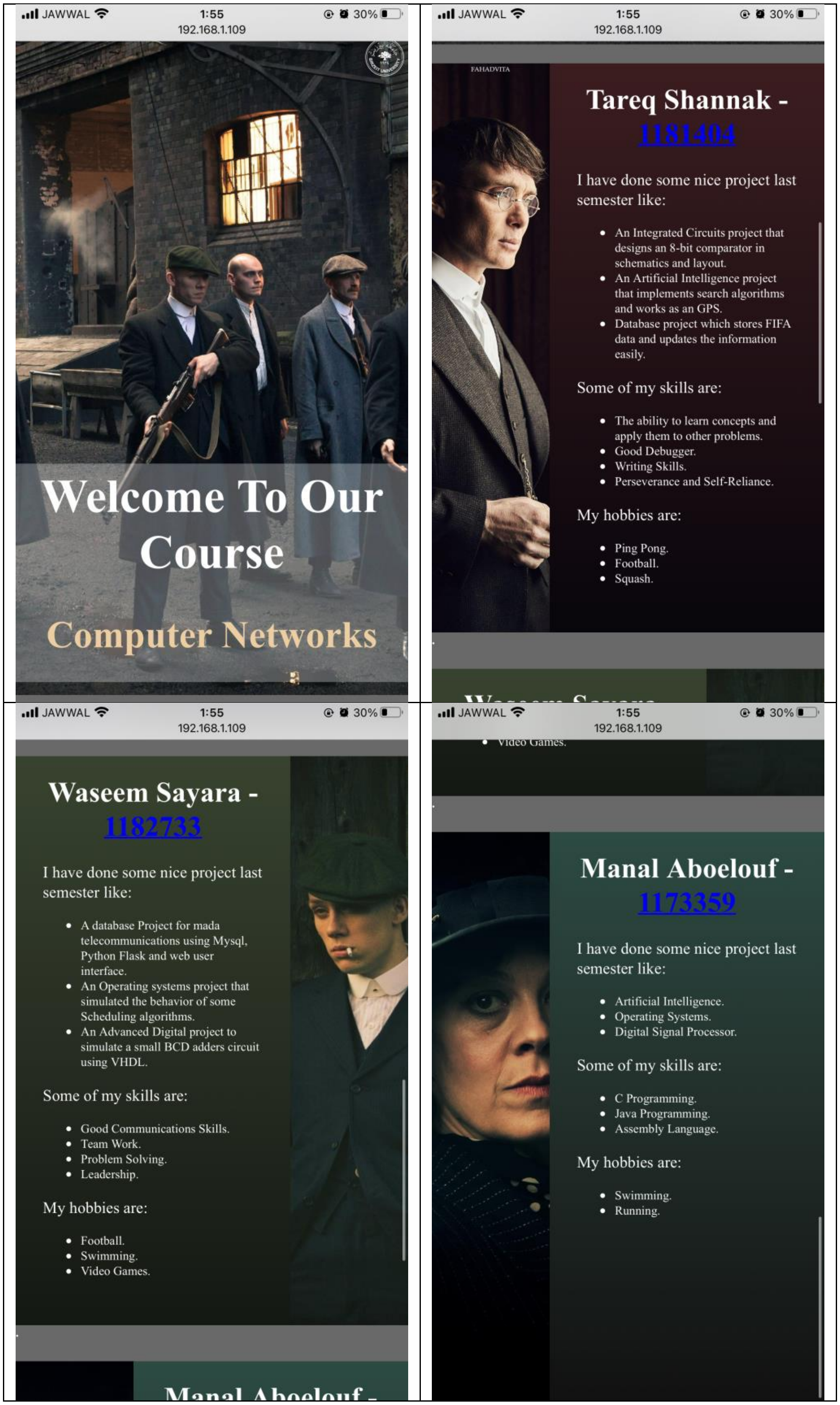
Some of my skills are:

- C Programming.
- Java Programming.
- Assembly Language.

My hobbies are:

- Swimming.
- Running.

Localhost:9000 or Localhost:9000/index.html From IPhone



Welcome To Our Course

Computer Networks

Tareq Shannak - 1181404

I have done some nice project last semester like:

- An Integrated Circuits project that designs an 8-bit comparator in schematics and layout.
- An Artificial Intelligence project that implements search algorithms and works as an GPS.
- Database project which stores FIFA data and updates the information easily.

Some of my skills are:

- The ability to learn concepts and apply them to other problems.
- Good Debugger.
- Writing Skills.
- Perseverance and Self-Reliance.

My hobbies are:

- Ping Pong.
- Football.
- Squash.

Waseem Sayara - 1182733

I have done some nice project last semester like:

- A database Project for mada telecommunications using Mysql, Python Flask and web user interface.
- An Operating systems project that simulated the behavior of some Scheduling algorithms.
- An Advanced Digital project to simulate a small BCD adders circuit using VHDL.

Some of my skills are:

- Good Communications Skills.
- Team Work.
- Problem Solving.
- Leadership.

My hobbies are:

- Football.
- Swimming.
- Video Games.

Manal Aboelouf - 1173359

I have done some nice project last semester like:

- Artificial Intelligence.
- Operating Systems.
- Digital Signal Processor.

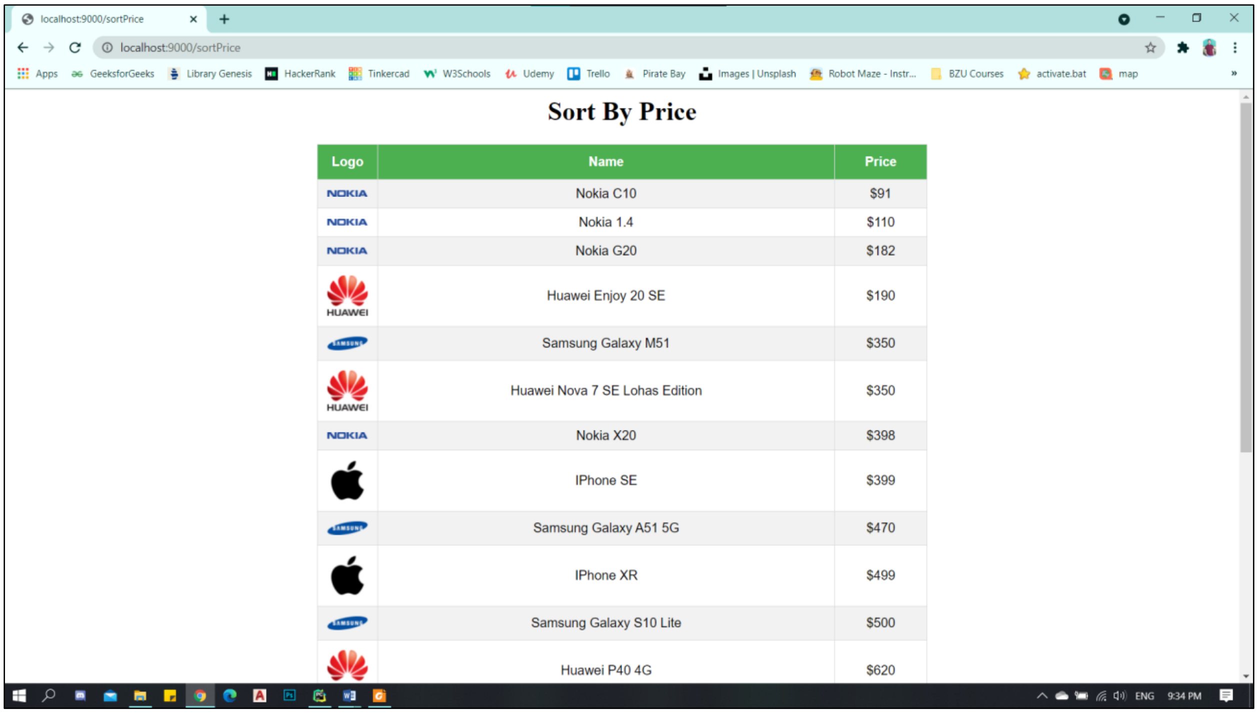
Some of my skills are:

- C Programming.
- Java Programming.
- Assembly Language.

My hobbies are:

- Swimming.
- Running.

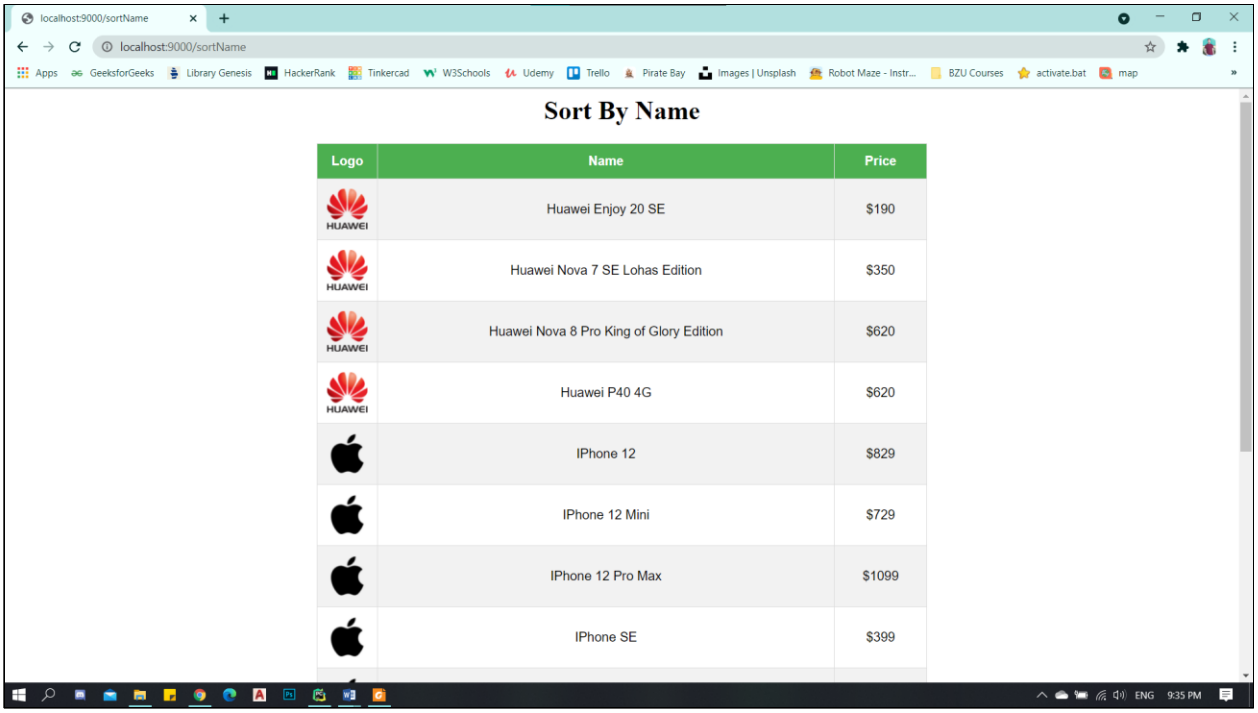
Localhost:9000/sortPrice



Sort By Price

Logo	Name	Price
NOKIA	Nokia C10	$91
NOKIA	Nokia 1.4	$110
NOKIA	Nokia G20	$182
HUAWEI	Huawei Enjoy 20 SE	$190
SAMSUNG	Samsung Galaxy M51	$350
HUAWEI	Huawei Nova 7 SE Lohas Edition	$350
NOKIA	Nokia X20	$398
	IPhone SE	$399
SAMSUNG	Samsung Galaxy A51 5G	$470
	IPhone XR	$499
SAMSUNG	Samsung Galaxy S10 Lite	$500
	Huawei P40 4G	$620

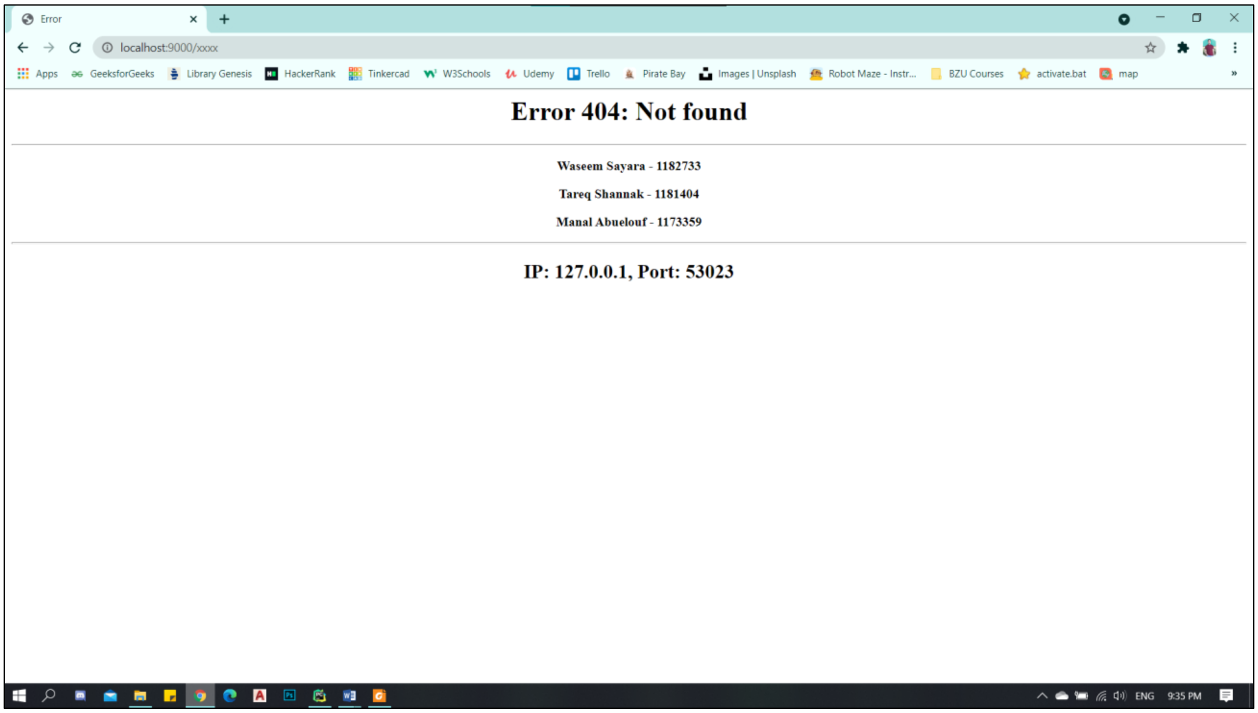
Localhost:9000/sortName



Sort By Name

Logo	Name	Price
HUAWEI	Huawei Enjoy 20 SE	$190
HUAWEI	Huawei Nova 7 SE Lohas Edition	$350
HUAWEI	Huawei Nova 8 Pro King of Glory Edition	$620
HUAWEI	Huawei P40 4G	$620
	IPhone 12	$829
	IPhone 12 Mini	$729
	IPhone 12 Pro Max	$1099
	IPhone SE	$399

Localhost:9000/xxxx



Error 404: Not found

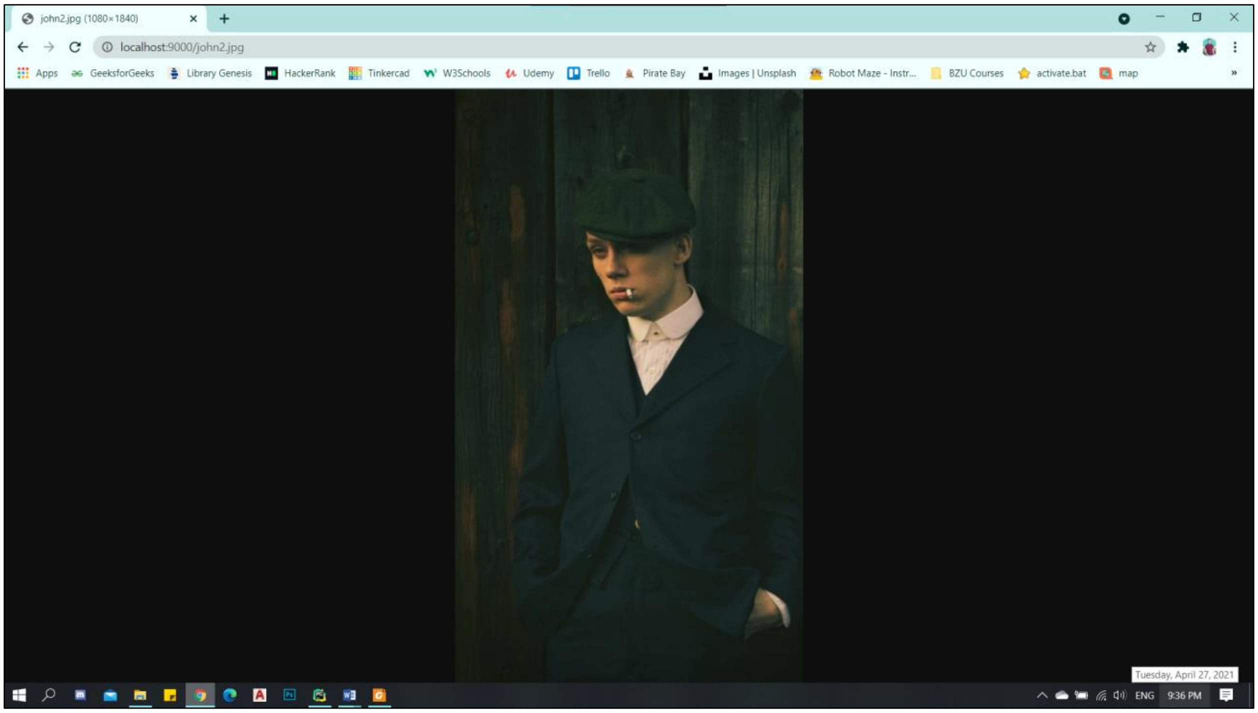
Waseem Sayara - 1182733

Tareq Shannak - 1181404

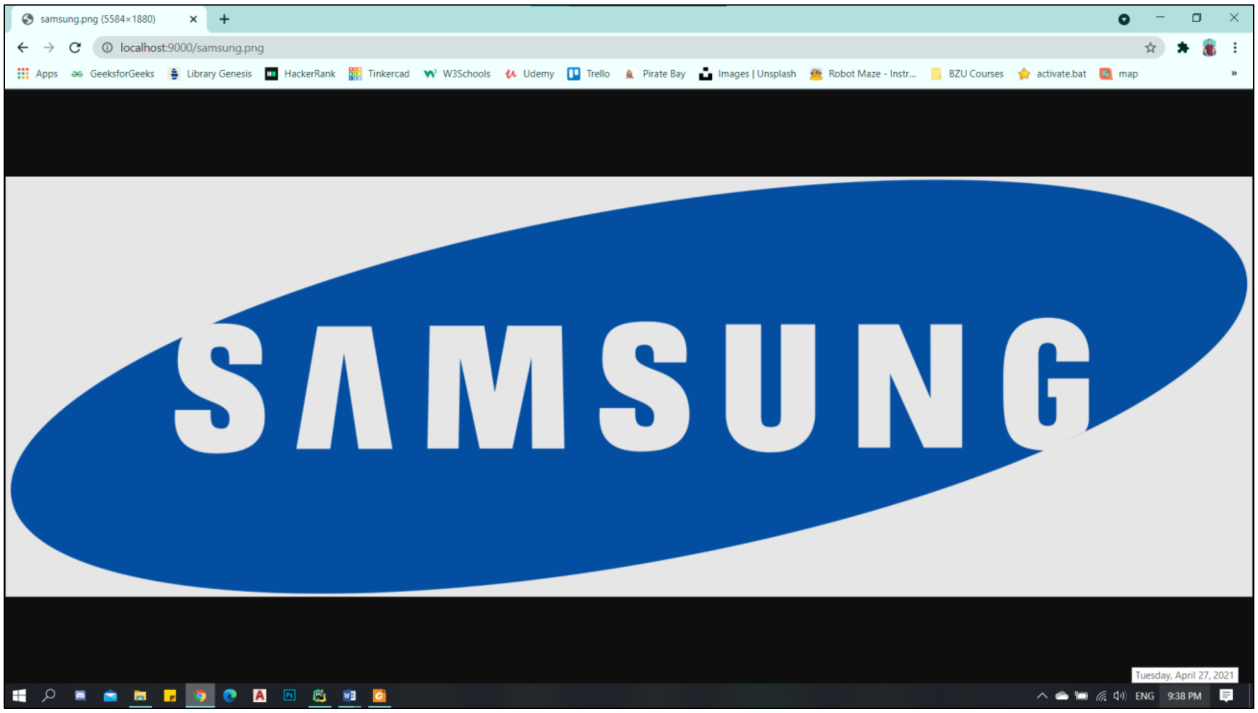
Manal Abuelouf - 1173359

IP: 127.0.0.1, Port: 53023

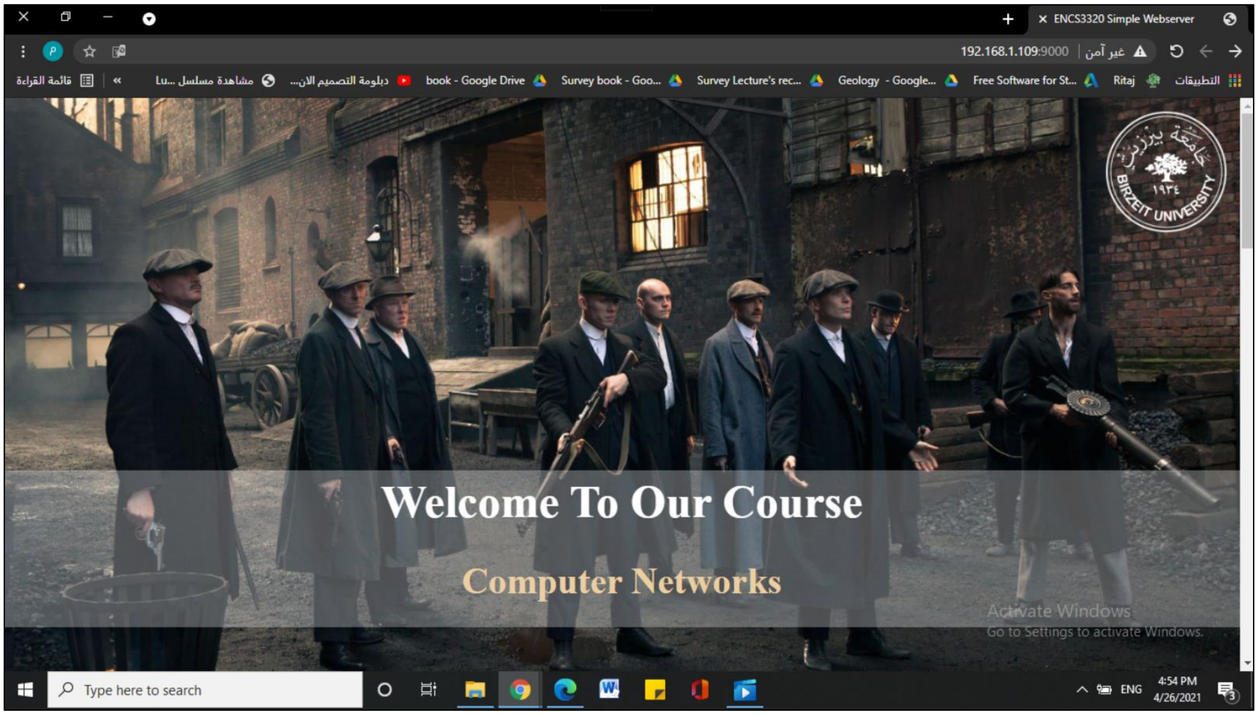
http://localhost:9000/john2.jpg



http://localhost:9000/samsung.png



Screenshot from another computer: 192.168.1.109:9000/index.html

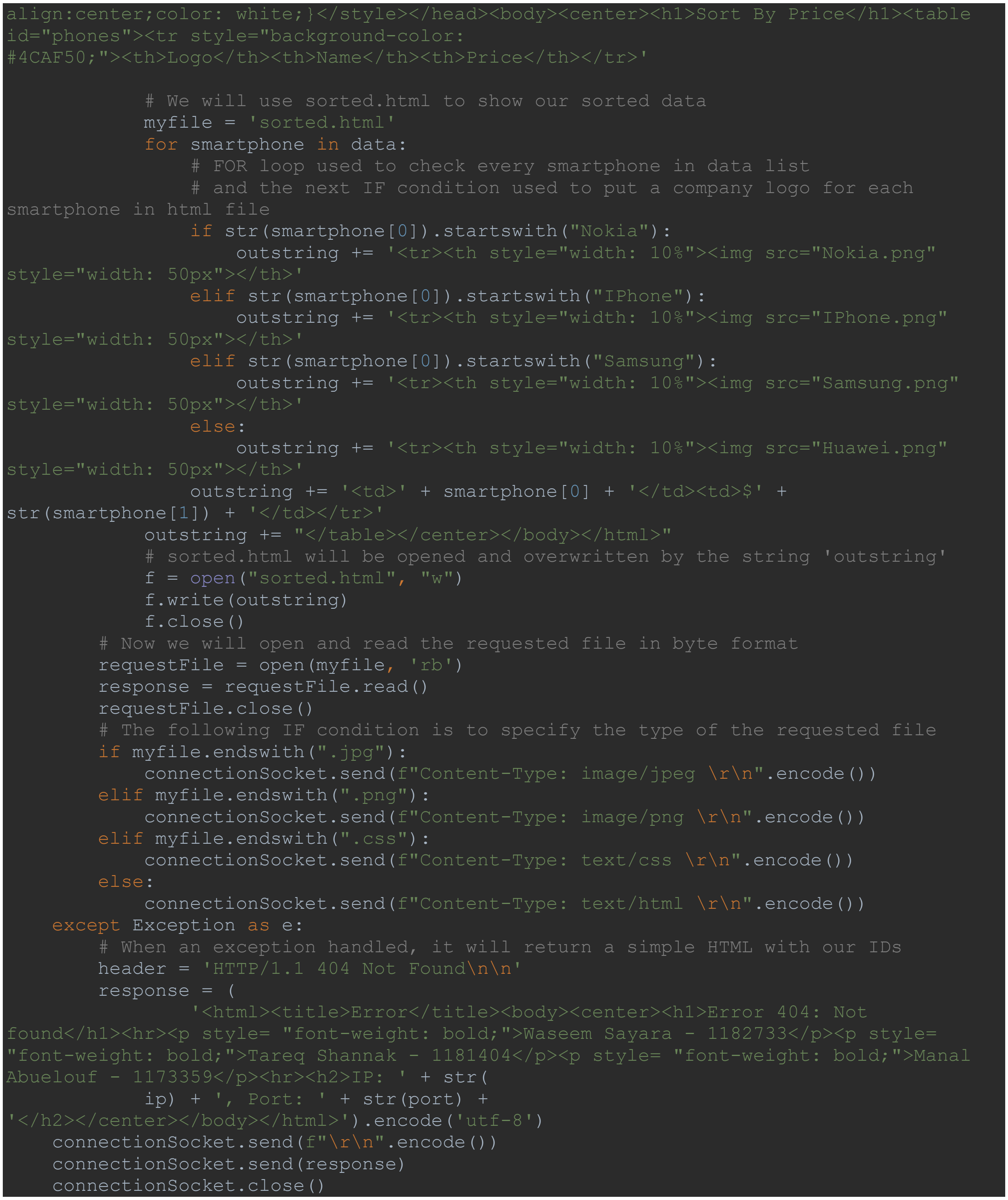


Welcome To Our Course

Computer Networks

Code as text

```
from socket import *

data = []


def readfile():
    """Read the data of smartphones from the txtfile.txt"""
    file = open("txtfile.txt", "r")  # create file that read from input
    info = file.readlines()  # read line by line from file and put the data in info
    for line in info:  # split the data from file and append it in another list
        li = line.split(";")
        li[1] = str(li[1]).replace("\n", "")  # Remove the newline signal
        li[1] = int(li[1])
        data.append(li)


readfile()
serverPort = 9000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(("", serverPort))
serverSocket.listen(1)
print("The server is ready to receive")

while True:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    print("IP: " + addr[0] + ", Port: " + str(addr[1]))
    print(sentence)
    ip = addr[0]
    port = addr[1]
    string_list = sentence.split(' ')  # Split request from spaces
    method = string_list[0]
    requestFile = string_list[1]
    connectionSocket.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'  # Default File
        elif myfile == 'sortName' or myfile == 'sortPrice':
            # if the user requests to sort the smartphones, it will enter this IF
condition
            if myfile == 'sortName':
                # Sort the data according to the names of the smartphones ascending
                data.sort()
                outstring = '<html><head><style>#phones {font-family: Arial,
Helvetica, sans-serif;text-align:center;border-collapse: collapse;width: 50%;}
#phones td, #phones th {border: 1px solid #ddd;padding: 8px;}  #phones tr:nth-
child(even){background-color: #f2f2f2;}  #phones tr:hover {background-color: #ddd;}
#phones th {padding-top: 12px;padding-bottom: 12px;text-align: left;text-
align:center;color: white;}</style></head><body><center><h1>Sort By Name</h1><table
id="phones"><tr style="background-color:
#4CAF50;"><th>Logo</th><th>Name</th><th>Price</th></tr>'
            else:
                # Sort the data according to the prices of the smartphones ascending
                data.sort(key=lambda data: data[1])
                outstring = '<html><head><style>#phones {font-family: Arial,
Helvetica, sans-serif;text-align:center;border-collapse: collapse;width: 50%;}
#phones td, #phones th {border: 1px solid #ddd;padding: 8px;}  #phones tr:nth-
child(even){background-color: #f2f2f2;}  #phones tr:hover {background-color: #ddd;}
#phones th {padding-top: 12px;padding-bottom: 12px;text-align: left;text-
```



```
align:center;color: white;}</style></head><body><center><h1>Sort By Price</h1><table
id="phones"><tr style="background-color:
#4CAF50;"><th>Logo</th><th>Name</th><th>Price</th></tr>'

            # We will use sorted.html to show our sorted data
            myfile = 'sorted.html'
            for smartphone in data:
                # FOR loop used to check every smartphone in data list
                # and the next IF condition used to put a company logo for each
smartphone in html file
                if str(smartphone[0]).startswith("Nokia"):
                    outstring += '<tr><th style="width: 10%"><img src="Nokia.png"
style="width: 50px"></th>'
                elif str(smartphone[0]).startswith("IPhone"):
                    outstring += '<tr><th style="width: 10%"><img src="IPhone.png"
style="width: 50px"></th>'
                elif str(smartphone[0]).startswith("Samsung"):
                    outstring += '<tr><th style="width: 10%"><img src="Samsung.png"
style="width: 50px"></th>'
                else:
                    outstring += '<tr><th style="width: 10%"><img src="Huawei.png"
style="width: 50px"></th>'
                outstring += '<td>' + smartphone[0] + '</td><td>$' +
str(smartphone[1]) + '</td></tr>'
            outstring += "</table></center></body></html>"
            # sorted.html will be opened and overwritten by the string 'outstring'
            f = open("sorted.html", "w")
            f.write(outstring)
            f.close()
        # Now we will open and read the requested file in byte format
        requestFile = open(myfile, 'rb')
        response = requestFile.read()
        requestFile.close()
        # The following IF condition is to specify the type of the requested file
        if myfile.endswith(".jpg"):
            connectionSocket.send(f"Content-Type: image/jpeg \r\n".encode())
        elif myfile.endswith(".png"):
            connectionSocket.send(f"Content-Type: image/png \r\n".encode())
        elif myfile.endswith(".css"):
            connectionSocket.send(f"Content-Type: text/css \r\n".encode())
        else:
            connectionSocket.send(f"Content-Type: text/html \r\n".encode())
    except Exception as e:
        # When an exception handled, it will return a simple HTML with our IDs
        header = 'HTTP/1.1 404 Not Found\n\n'
        response = (
                '<html><title>Error</title><body><center><h1>Error 404: Not
found</h1><hr><p style= "font-weight: bold;">Waseem Sayara - 1182733</p><p style=
"font-weight: bold;">Tareq Shannak - 1181404</p><p style= "font-weight: bold;">Manal
Abuelouf - 1173359</p><hr><h2>IP: ' + str(
            ip) + ', Port: ' + str(port) +
'</h2></center></body></html>').encode('utf-8')
    connectionSocket.send(f"\r\n".encode())
    connectionSocket.send(response)
    connectionSocket.close()
```