

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**Computer Networks**

**ENCS3320**

**Project#1**

**Prepared by:**

**Ibraheem Ajaj 1190939**

**Ahmad Raddad 1180814**

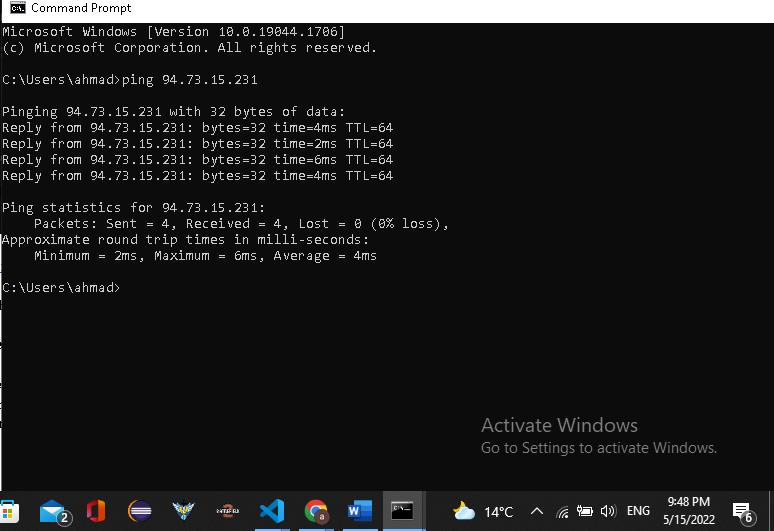
**Majd Maree 1192226**

**Instructor**

**Dr. Imad Tartir**

**Part 1: 1. Ping a device in the same network, e.g. from laptop to a smartphone**

**ping and your IP address. For example: ping 94.73.15.231. If your PC shows as below, it means that your devices are connected to the same Wi-Fi network.**

****

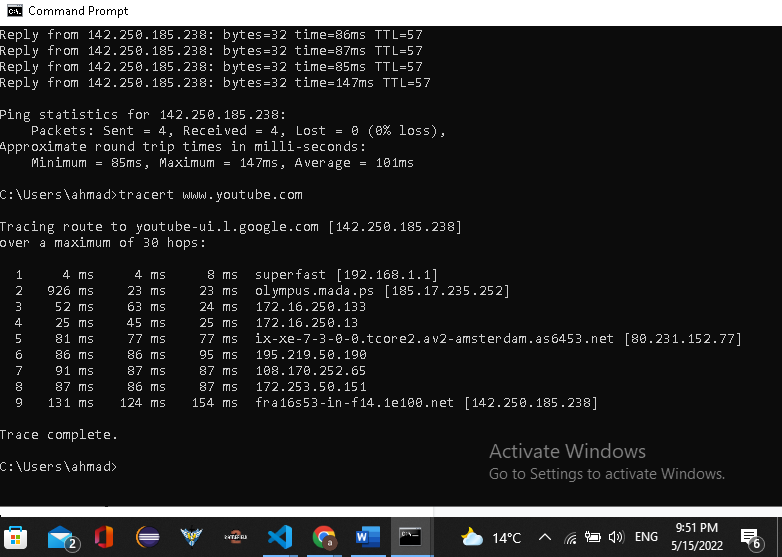
**2-ping** [**www.youtube.com**](http://www.youtube.com)

**That response shows the URL you’re pinging, the IP address associated with that URL, and the size of the packets being sent on the first line. The next four lines show the replies from each individual packet, including the time (in milliseconds) it took for the response and the time-to-live (TTL) of the packet, which is the amount of time that must pass before the packet is discarded. At the bottom, you’ll see a summary that shows how many packets were sent and received, as well as the minimum, maximum, and average response time.**

****

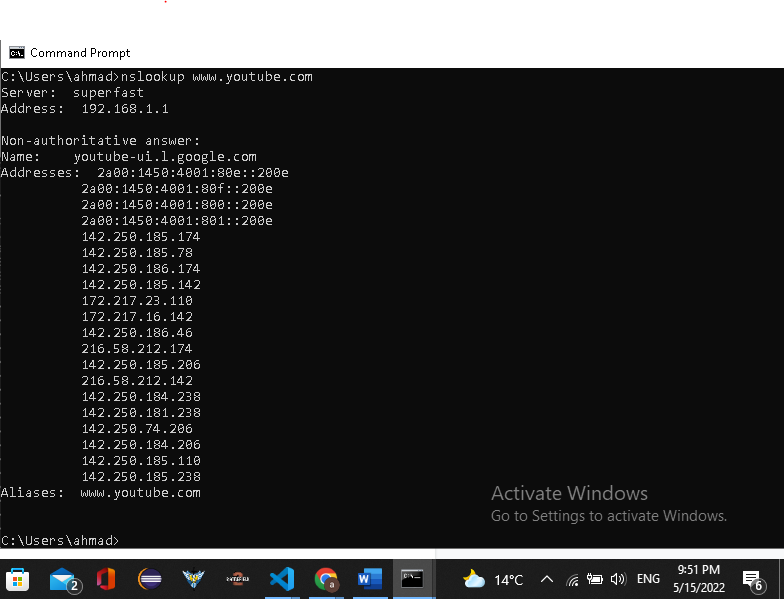
**3. tracert** [**www.youtube.com**](http://www.youtube.com)

**The command will return output indicating the hops discovered and time (in milliseconds) for each hop. The first line represents your home router (assuming you’re behind a router), the next lines represent your ISP, and each line further down represents a router that’s further away.**

****

**4. nslookup** [**www.youtube.com**](http://www.youtube.com)

**nslookup (from name server lookup) is a network administration command-line tool available in many computer operating systems for querying the Domain Name System (DNS) to obtain domain name or IP address mapping, or other DNS records. In nslookup** [**www.youtube.com**](http://www.youtube.com)**, can use this command to see how many A records are there and see the IP Addresses of each one**

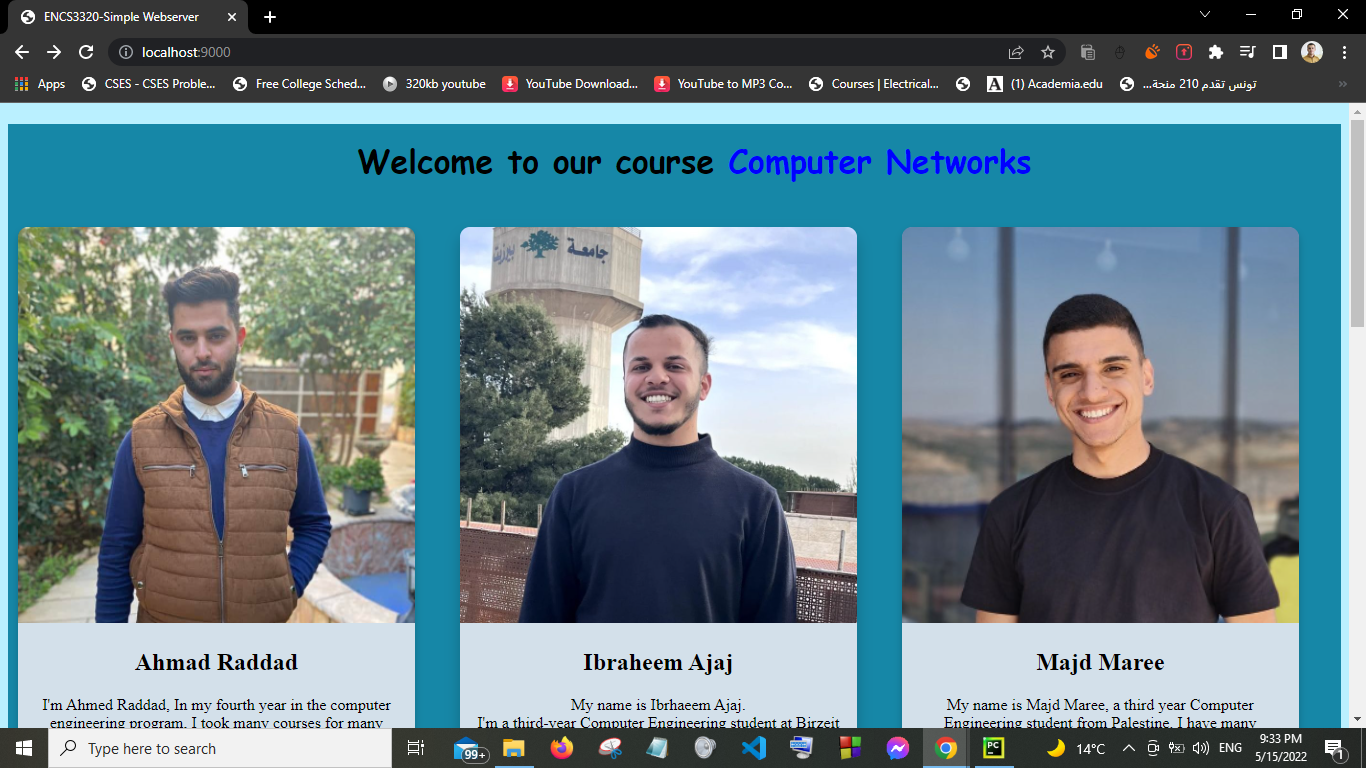
****

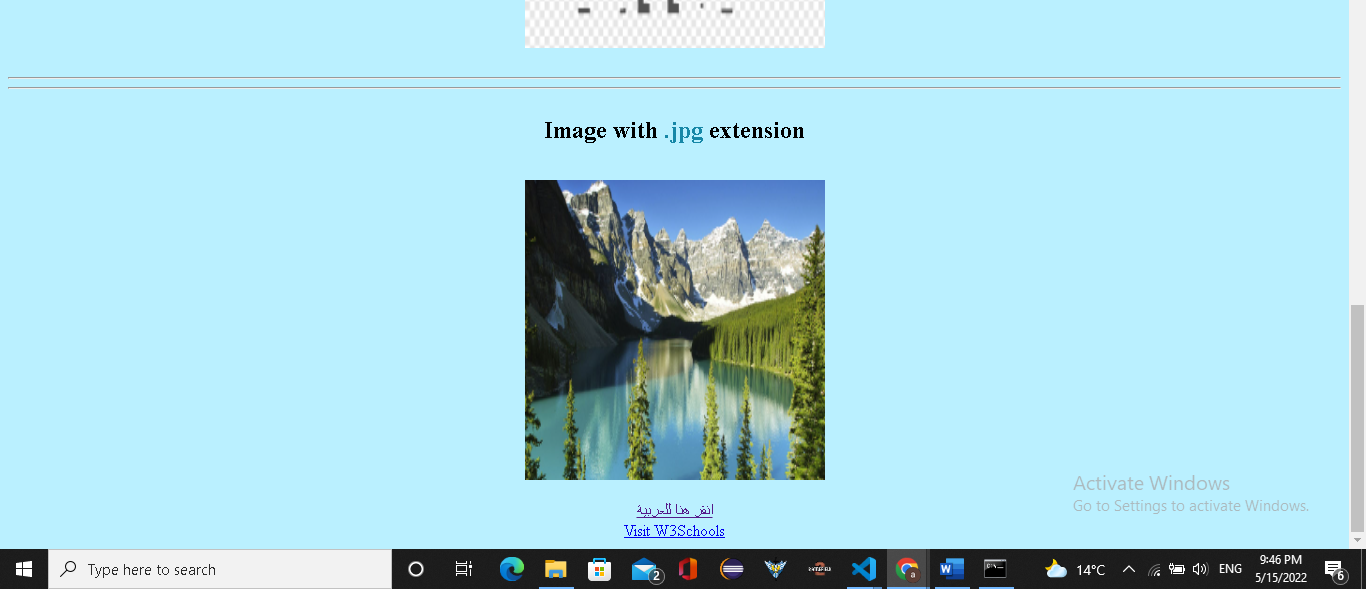
**Part 2**

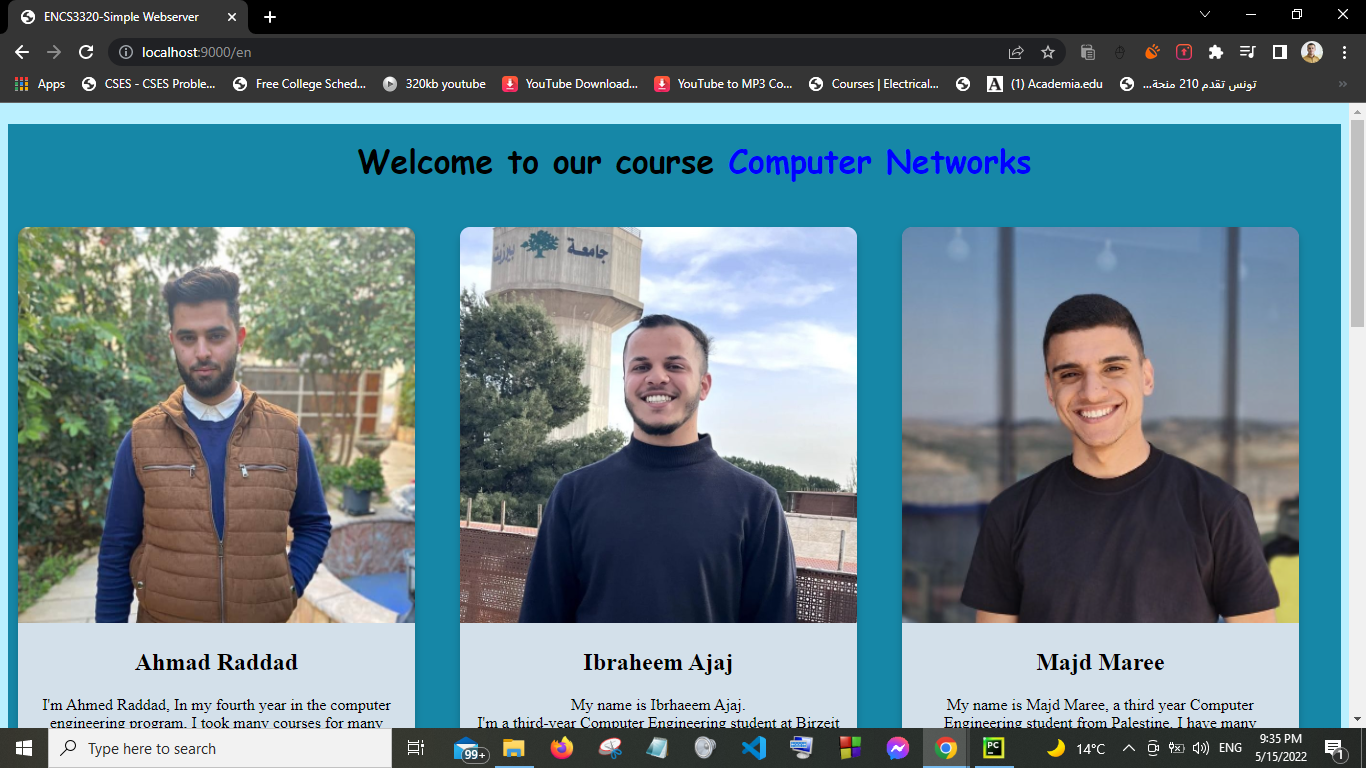
**Run the server:**

[**http://localhost:9000/**](http://localhost:9000/) **or** [**http://localhost:9000/en**](http://localhost:9000/en) **or** [**http://localhost:9000/mai**](http://localhost:9000/en)**n\_en.html will open the following html file (main\_en.html)**

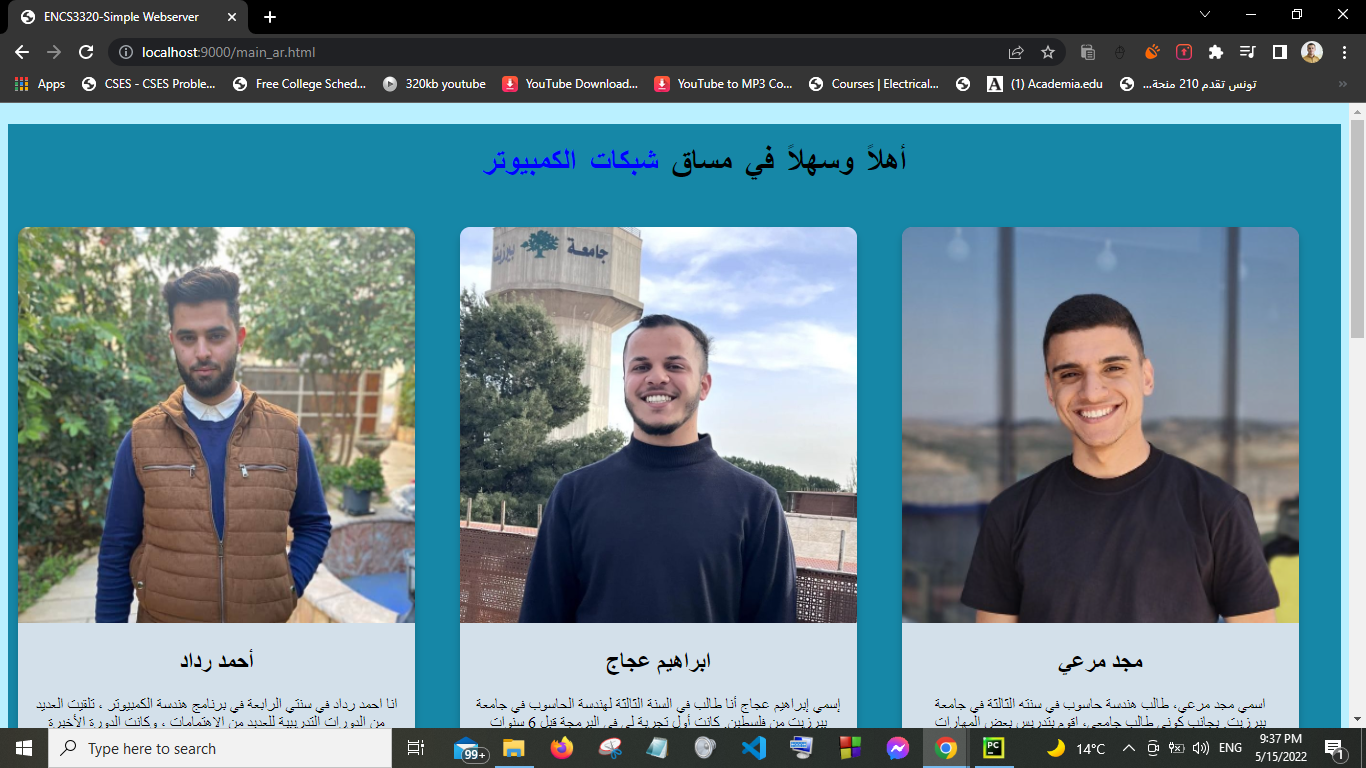
**When the server starts the main (english) page opens (It opens also when the links ends with en or main\_en.html)**

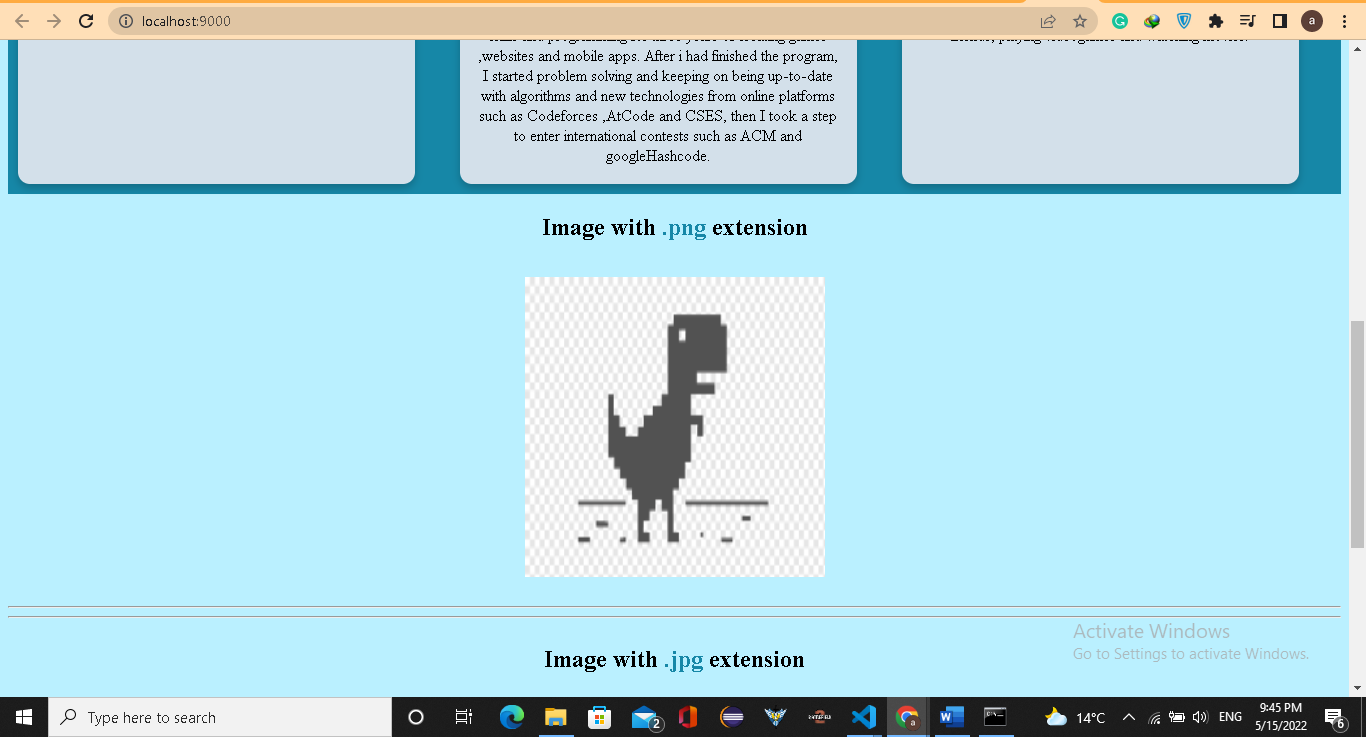
****

****

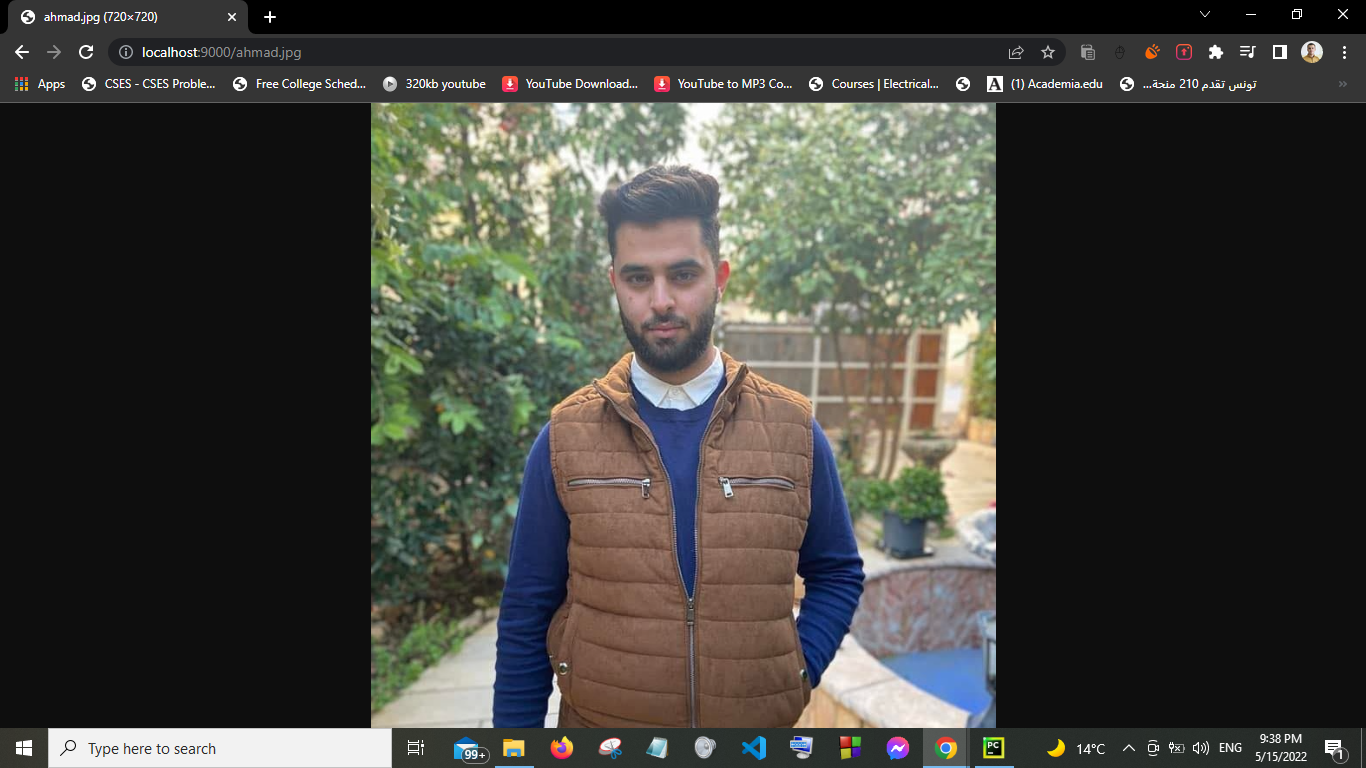
****

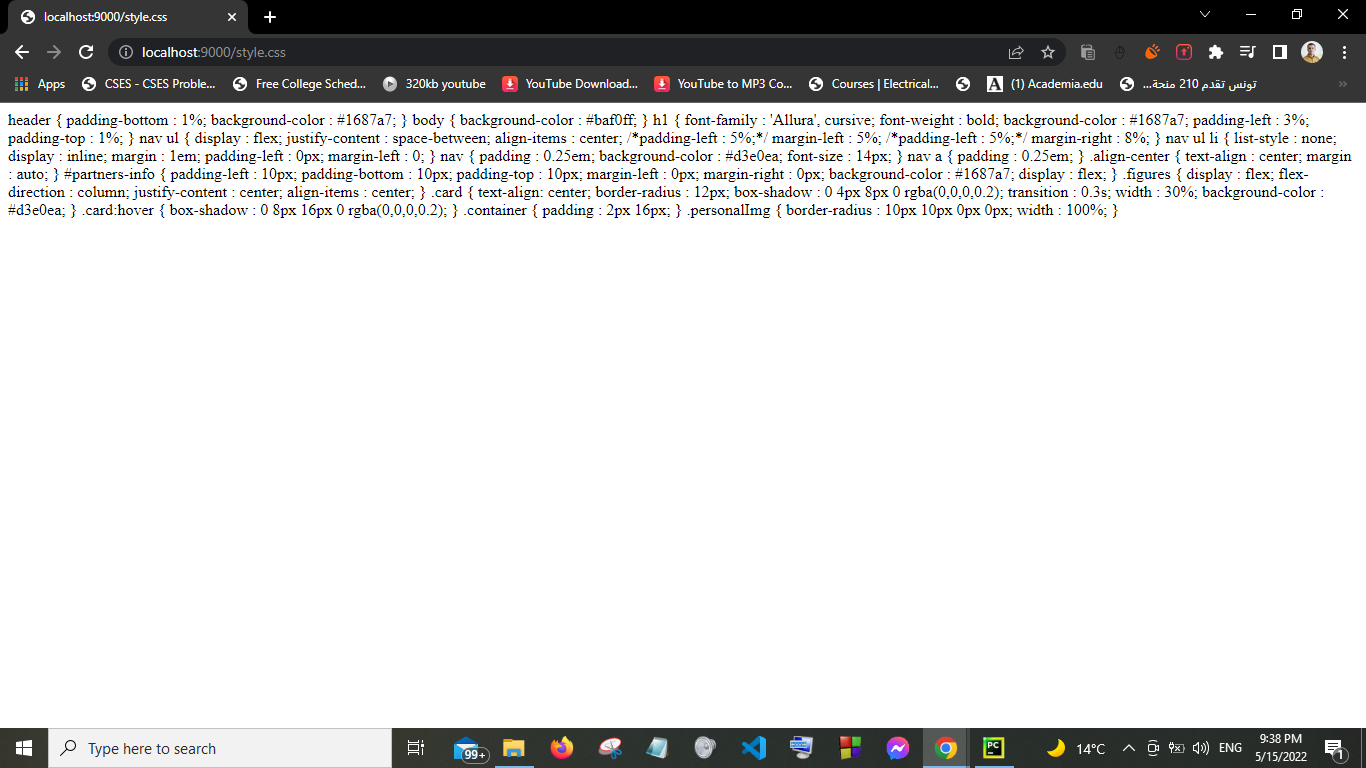
**And below the main page html file (main\_ar.html) but in arabic (**[**http://localhost:9000/ar**](http://localhost:9000/ar) **or** [**http://localhost:9000/main\_ar.html**](http://localhost:9000/main_ar.html) **)**

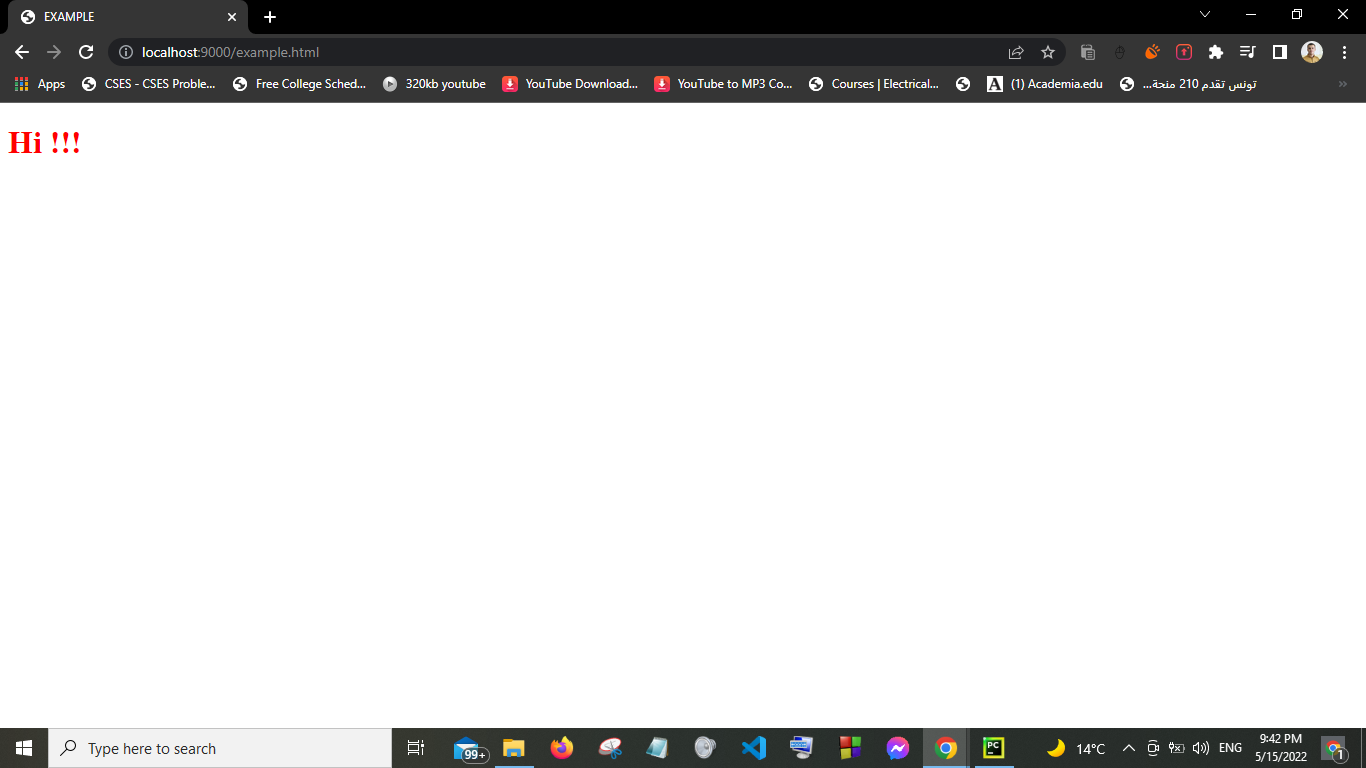
****

****

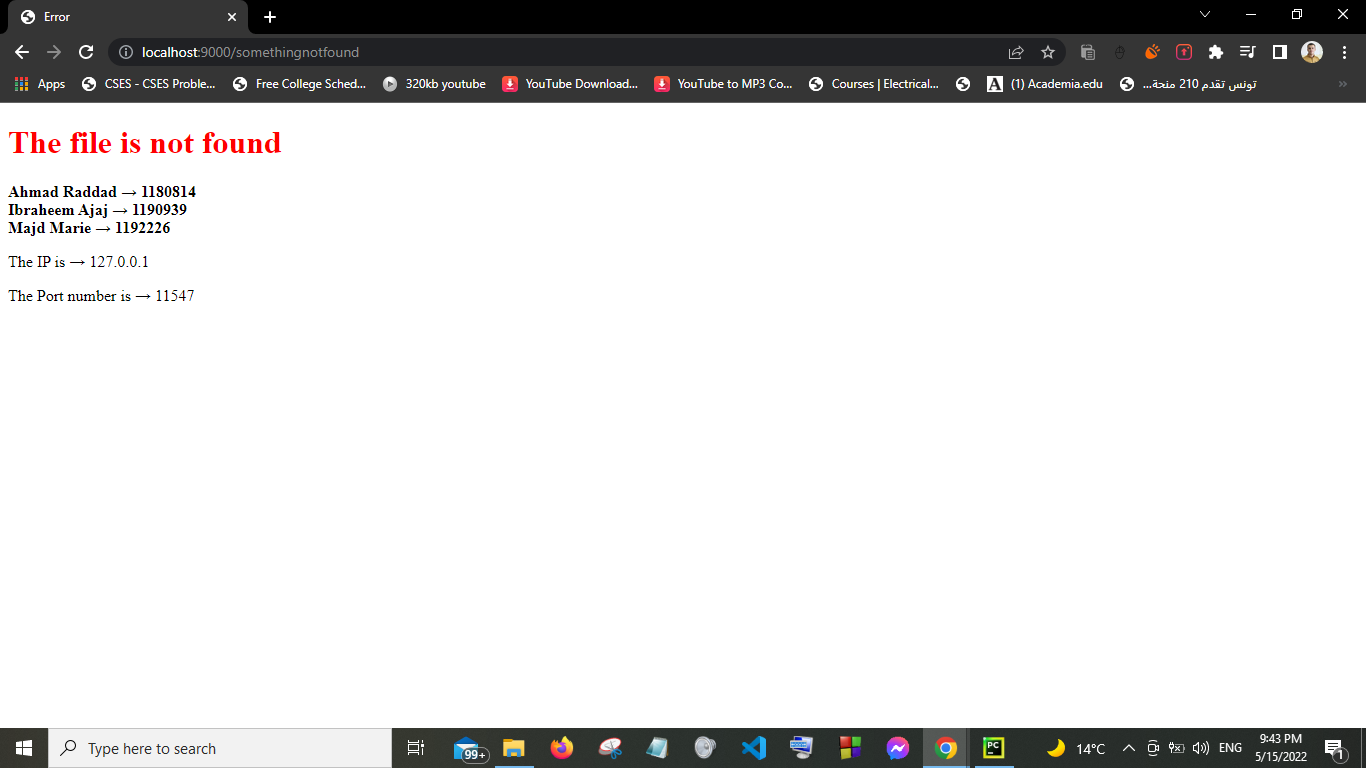
**If the link ends with file name it opens the file if found :**

****

****

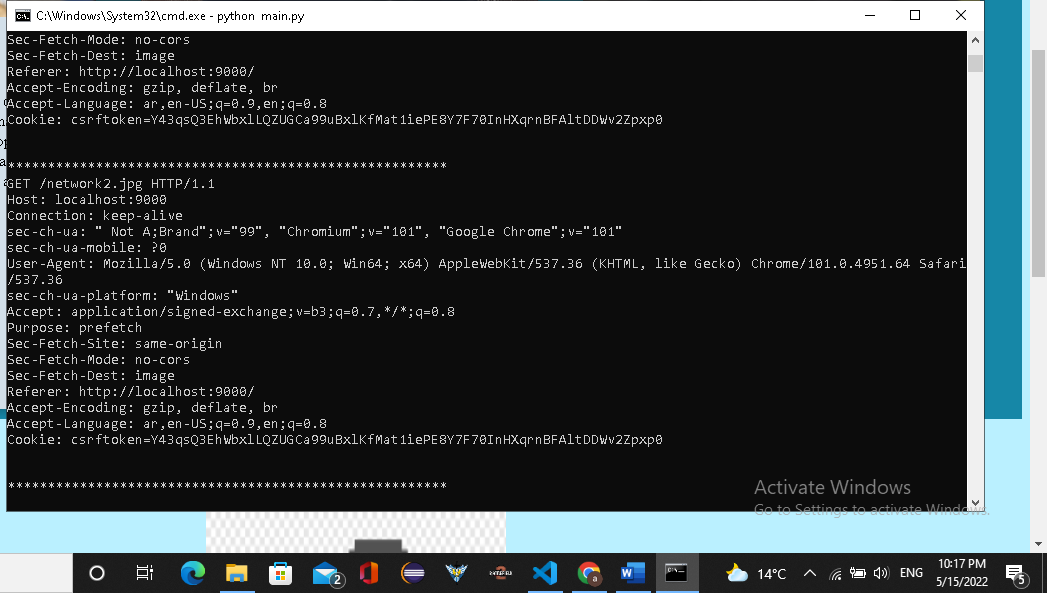
****

**If the file not found then Error 404 response occurs and error page opens:**

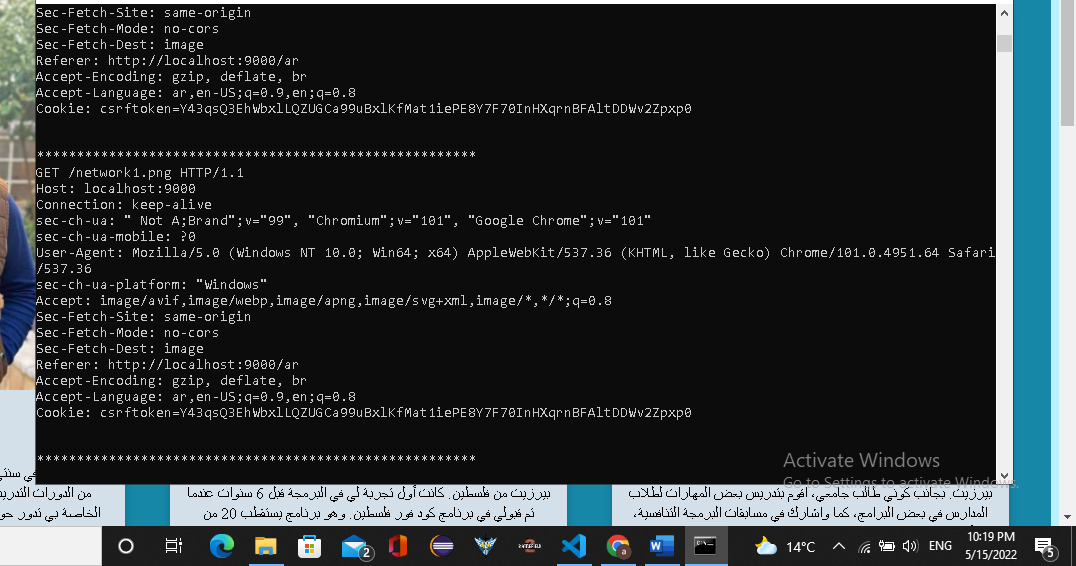
****

**Here is the main page (English page) response :**

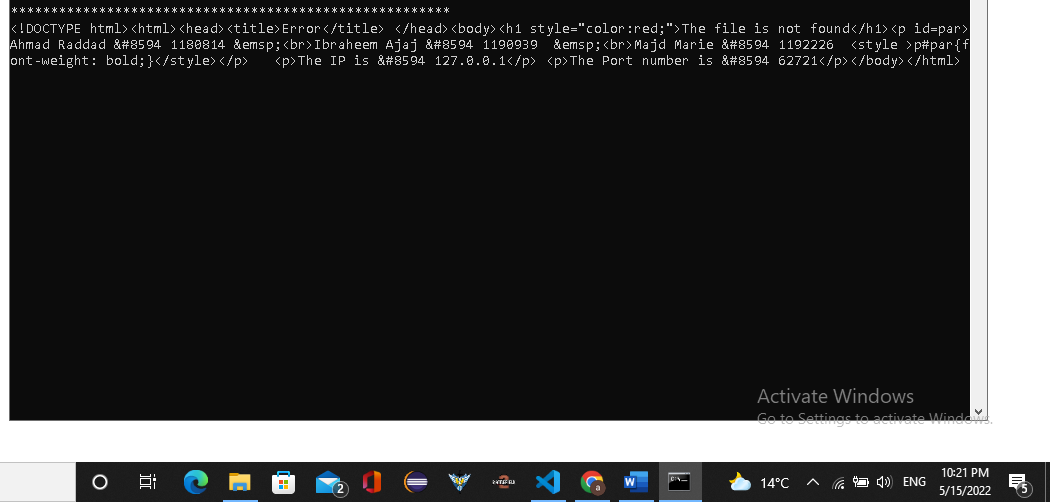
**Localhost:9000**

****

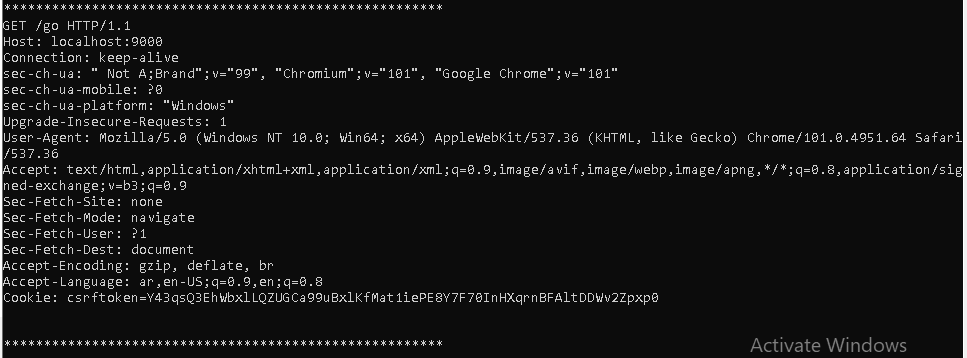
**This for (**[**http://localhost:9000/ar**](http://localhost:9000/ar)**)**

****

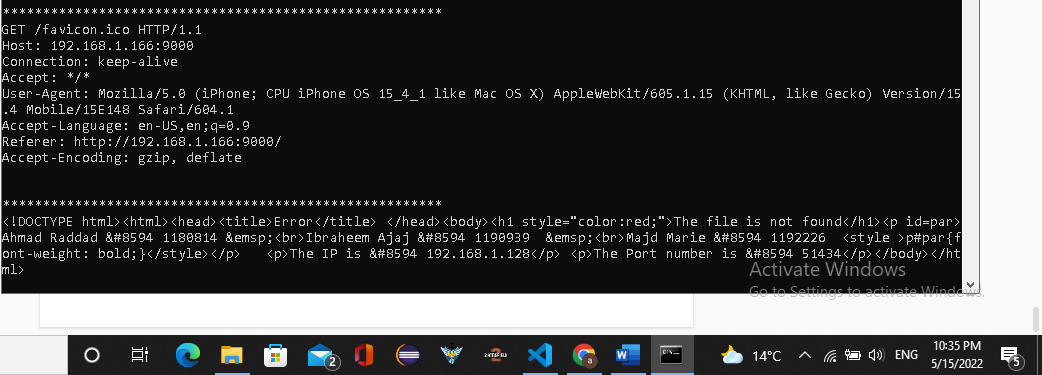
**This is the HTTP response in case there is an error**

****

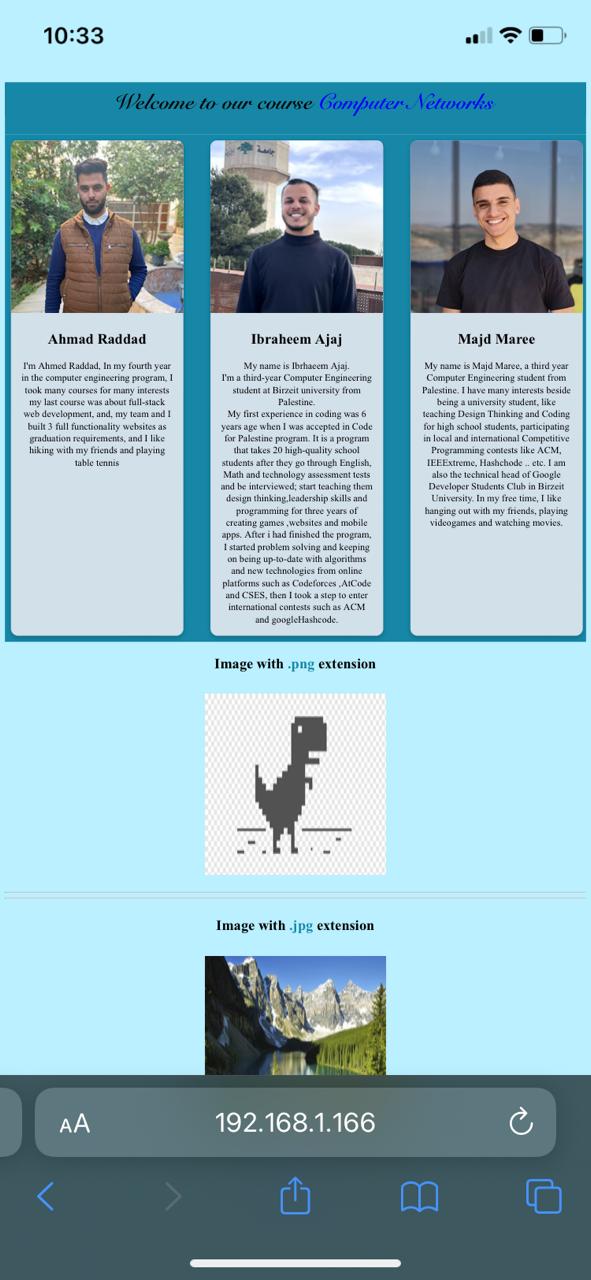
**(**[**http://localhost:9000/**](http://localhost:9000/ar)**go**)



The response when the page opened in another device:



Here is how the page looks like at another device:



Code:

from socket import \*

def Socketing(URL):

contentType = "" # define an empty var

extension = URL.split(".") # Splitting the URl to take the extension

if(len(extension) > 1): # Here we check the extension to determine the content type

if(extension[1] == "png"):

contentType = "image/png"

elif (extension[1] == "jpg"):

contentType = "image/jpg"

elif (extension[1] == "css"):

contentType = "text/css"

else:

contentType = "text/html"

else:

contentType = "text/html"

# Routing according to the user input at the suburl

if(URL == "" or URL == "en"):

URL = "main\_en.html"

elif(URL == "ar"):

URL = "main\_ar.html"

return [contentType, URL.strip()]

def main():

serverPort = 9000 # port number

ser = socket(AF\_INET, SOCK\_STREAM)

ser.bind(('', serverPort)) # here getting the socket

ser.listen(1) # the server is listening

print('The server is listening ...')

while True:

Socketconn, client\_address = ser.accept()

senc = Socketconn.recv(1024).decode()

token = senc.split("/")

if len(token)>=2:

URL = token[1].split(" ") #here to get the pure URl(no sapces in it)

contentType, fileName = Socketing(URL[0])

print(senc)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

# client\_address[0] this means to store client IP in ip var

ip = client\_address[0] # ip of the client

port = client\_address[1] # port of the client

try:

if (fileName=="go"):

Socketconn.send(bytes("HTTP/1.1 307 Temporary Redirect \r\n", "UTF-8"))

Socketconn.send(bytes("location: https://www.google.com \r\n", "UTF-8"))

Socketconn.send(bytes("\r\n", "UTF-8"))

Socketconn.close()

elif (fileName=="cn"):

Socketconn.send(bytes("HTTP/1.1 307 Temporary Redirect \r\n", "UTF-8"))

Socketconn.send(bytes("location: https://edition.cnn.com \r\n", "UTF-8"))

Socketconn.send(bytes("\r\n", "UTF-8"))

Socketconn.close()

elif (fileName=="bzu"):

Socketconn.send(bytes("HTTP/1.1 307 Temporary Redirect \r\n", "UTF-8"))

Socketconn.send(bytes("location: https://www.birzeit.edu \r\n", "UTF-8"))

Socketconn.send(bytes("\r\n", "UTF-8"))

Socketconn.close()

else:

with open(fileName, "rb") as f:

fileContent = f.read()

Socketconn.send(bytes("HTTP/1.1 200 OK\r\n", "UTF-8"))

Socketconn.send(bytes("Content-Type:" + contentType + "\r\n", "UTF-8"))

Socketconn.send(b"\r\n")

Socketconn.send(fileContent)

Socketconn.close()

except IOError:

# There is an html code in case there is an error in the request so this will appear to the user 404 not found page

fileContent = '<!DOCTYPE html><html><head><title>Error</title> </head><body><h1 style="color:red;">The file is not found</h1><p id=par>Ahmad Raddad &#8594 1180814 &emsp;<br>Ibraheem Ajaj &#8594 1190939 &emsp;<br>Majd Marie &#8594 1192226 <style >p#par{font-weight: bold;}</style></p> <p>The IP is &#8594 ' + str(ip)+'</p> <p>The Port number is &#8594 '+str(port)+'</p></body></html>'

Socketconn.send(bytes("HTTP/1.1 404 Not Found \r\n", "UTF-8"))

Socketconn.send(bytes("Content-Type: text/html\r\n", "UTF-8"))

Socketconn.send(bytes("\r\n", "UTF-8"))

Socketconn.send(bytes(fileContent, "UTF-8"))

Socketconn.close()

main()