

Department of Electrical and Computer Engineering

ENCS3320-Computer Networks

First Semester, 2023/2024

Project1

Prepared By:

Ziad Masalma 1202199 **Section : 1**

Mujahed Abuali 1211047 **Section : 2**

Manal Abubaker 1221098 **Section : 4**

Date :19/11/2023

PART I:

1)

Ping: Ping is a network protocol used to test the availability and responsiveness of a device or server on a network.

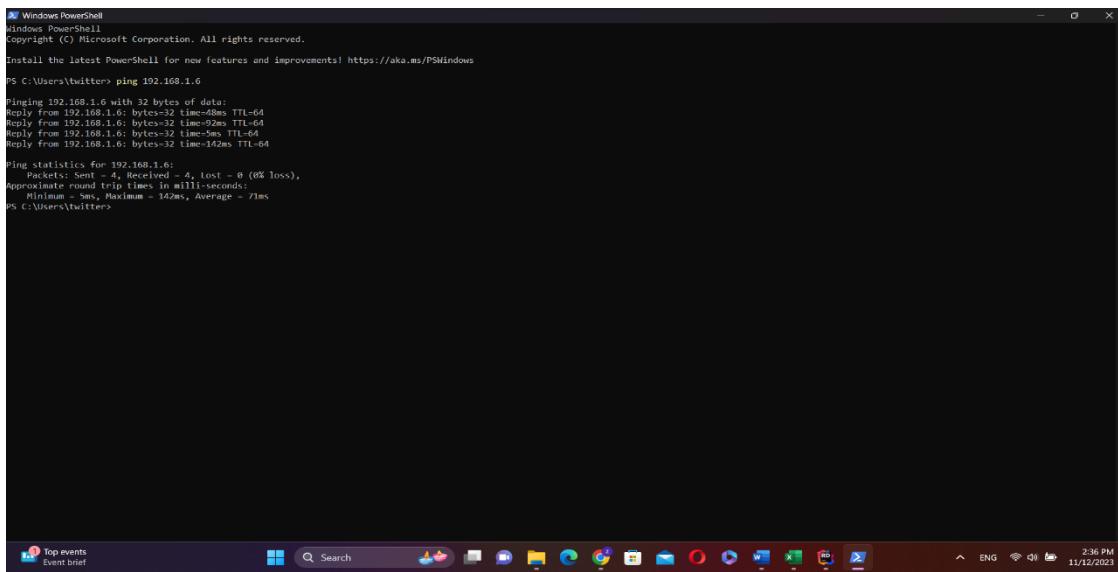
Tracert: Short for "traceroute", a command-line networking tool that is used to identify the path that network packets take as they travel from a source to a destination over an IP network.

Nslookup: Short for "name server lookup" is a command-line networking tool used to query the Domain Name System (DNS) to obtain information about domain names, IP addresses, and other related information.

Telnet: is a network protocol used for establishing a remote connection between computers over a network.

2)

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



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\twitter> ping 192.168.1.6

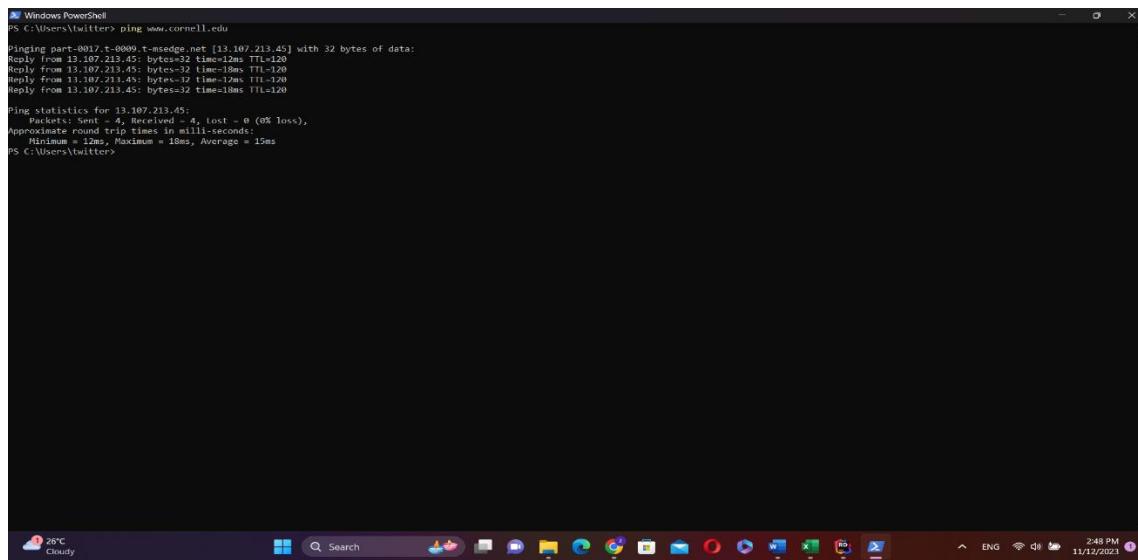
Pinging 192.168.1.6 with 32 bytes of data:
Reply from 192.168.1.6: bytes=32 time=2ms TTL=64
Reply from 192.168.1.6: bytes=32 time=2ms TTL=64
Reply from 192.168.1.6: bytes=32 time=5ms TTL=64
Reply from 192.168.1.6: bytes=32 time=142ms TTL=64

Ping statistics for 192.168.1.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milliseconds:
        Minimum = 2ms, Maximum = 142ms, Average = 71ms

PS C:\Users\twitter>
```

The previous image shows that data packets have been sent have the same TTL (time to live), and the result of each transmission is one line, all packets are received with the average 71 ms.

B) ping www.cornell.edu



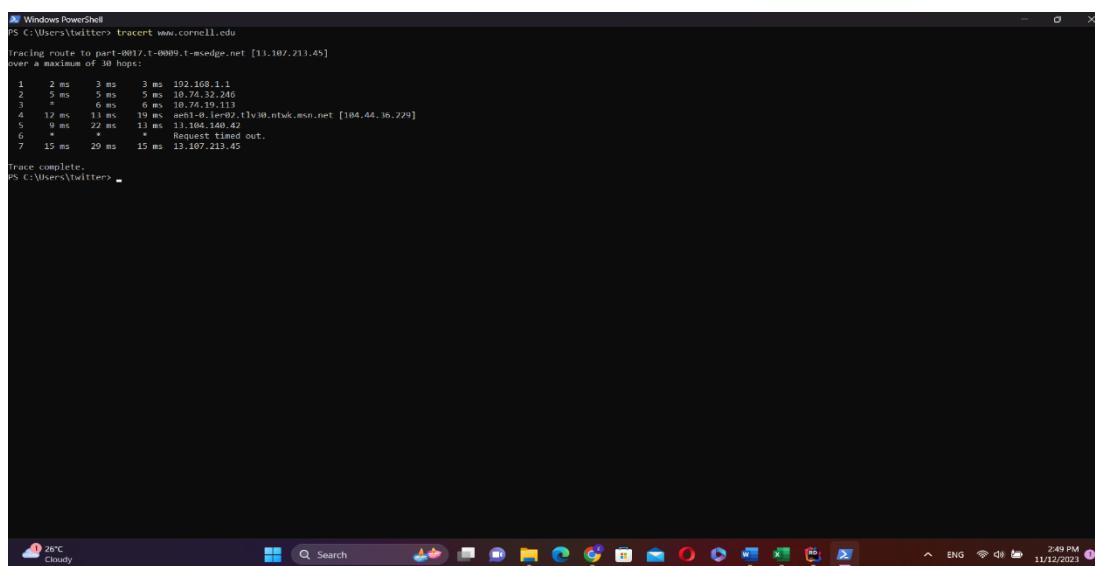
```
Windows PowerShell
PS C:\Users\twitter> ping www.cornell.edu

Pinging part-0001.t-0009.t-msedge.net [13.107.213.45] with 32 bytes of data:
Reply from 13.107.213.45: bytes=32 time=12ms TTL=120
Reply from 13.107.213.45: bytes=32 time=18ms TTL=120
Reply from 13.107.213.45: bytes=32 time=17ms TTL=120
Reply from 13.107.213.45: bytes=32 time=18ms TTL=120

Ping statistics for 13.107.213.45:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 18ms, Average = 15ms
PS C:\Users\twitter>
```

The figure below shows that we received a response from 13.107.213.45 when we sent 4 packets, each with the same TTL (time to live), and that the average delay for all packets received is 15 ms. Additionally, we can see that it takes longer to receive a response from cornell.net than it does from a smartphone connected to the same network as my laptop.

C) tracert www.cornell.com



```
Windows PowerShell
PS C:\Users\twitter> tracert www.cornell.edu

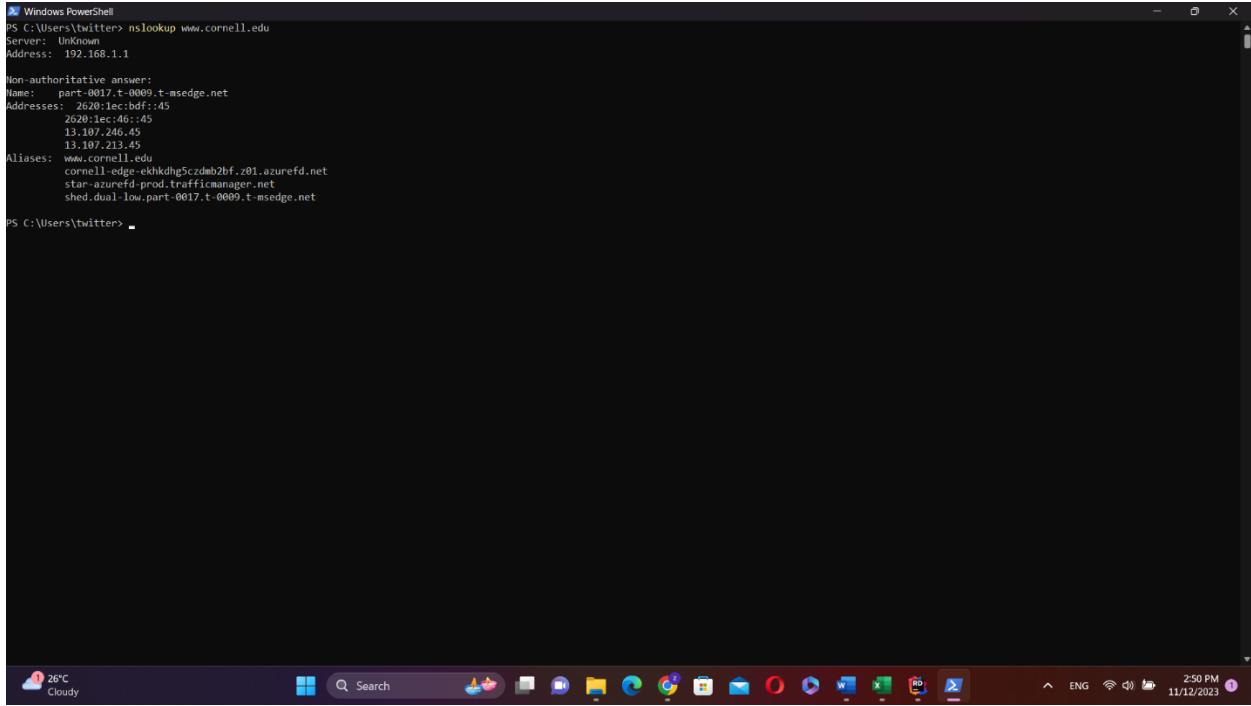
Tracing route to part-0001.t-0009.t-msedge.net [13.107.213.45]
over a maximum of 30 hops:
1  2 ms   3 ms   3 ms  102.169.1.1
2  5 ms   5 ms   5 ms  10.74.32.246
3  *       6 ms   6 ms  10.74.19.113
4  13 ms   11 ms  13 ms  13.107.213.45
5  9 ms   22 ms  13 ms  13.104.140.42
6  *       *       Request timed out.
7  15 ms   29 ms  15 ms  13.107.213.45

Trace complete.
PS C:\Users\twitter>
```

As you can see, there are several rows divided into columns on the report. Each row represents a "hop" along the route. Think of it as a check-in point where the signal gets its next set of directions. Each row is divided into five columns.

Hop #	RTT 1	RTT 2	RTT 3	Name/IP Address
-------	-------	-------	-------	-----------------

D) nslookup www.cornell.com



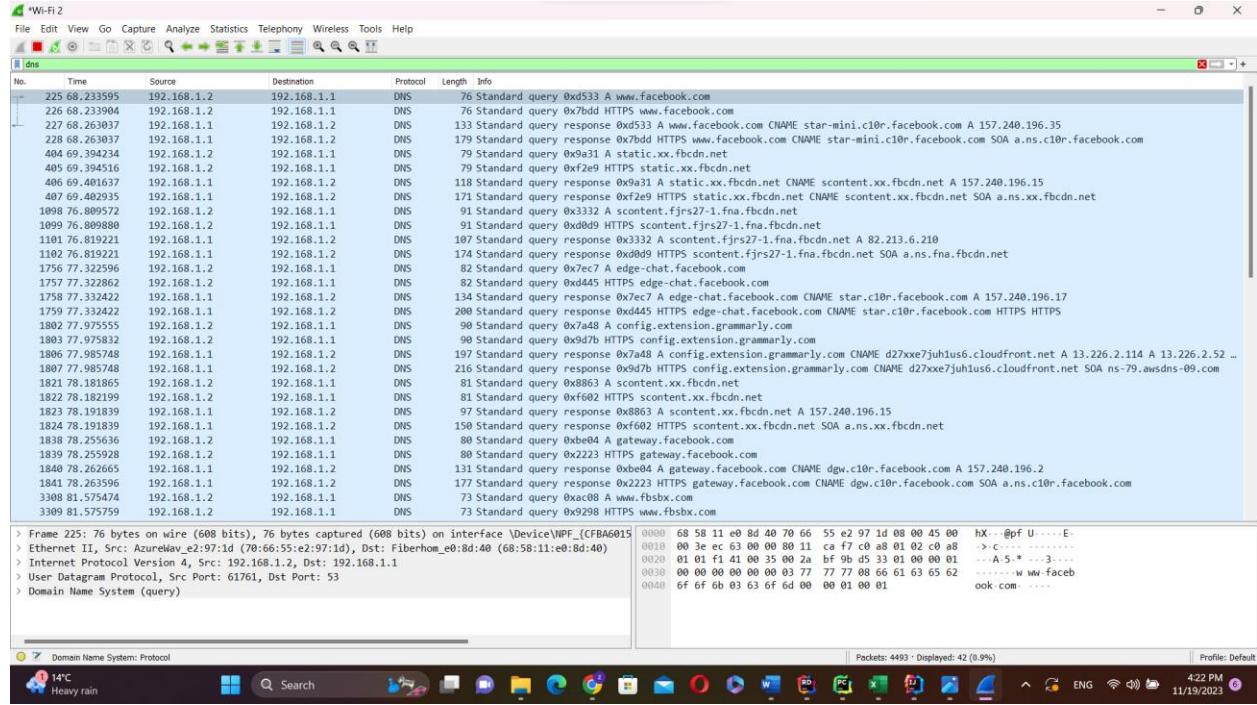
```
Windows PowerShell
PS C:\Users\twitter> nslookup www.cornell.edu
Server: UnKnown
Address: 192.168.1.1

Non-authoritative answer:
Name: part-0017.t-0009.t-msedge.net
Addresses: 2620:1ec:bdf:45
           13.197.246.45
           13.197.213.45
Aliases: www.cornell.edu
cornell-edge-ekhkdhg5czdmb2bf.z01.azurefd.net
star-azurefd-prod.trafficmanager.net
shed.dual-low.part-0017.t-0000.t-msedge.net

PS C:\Users\twitter>
```

We can see that it publishes the name and addresses of the server, which is the host that we sent a probe to, as well as the IP address that corresponds to the host, which in this case is the IP address of my laptop.

3)



This DNS analysis provides insights into the DNS communication patterns between the client and the DNS server. Understanding these patterns is crucial for maintaining network performance, security, and reliability. Ongoing monitoring and periodic reviews will contribute to a robust and efficient DNS infrastructure.

From the ping results, do you think the response you have got is from USA? Explain your answer briefly.

In the first set of pings to IP address 192.168.1.9, the times are relatively low (between 49ms and 95ms), suggesting a low-latency connection within a local network. However, the IP address 192.168.1.9 is typically associated with a local device on a private network, and it does not provide information about the geographic location.

In the second set of pings to www.cornell.edu (resolved to IP address 13.107.213.45), the response times are slightly higher (between 64ms and 86ms). While this indicates a moderate level of latency, it doesn't inherently reveal the geographic location of the server. The IP address belongs to Microsoft's content delivery network (CDN), and Microsoft has data centers distributed globally.

PART II:

TCP CLIENT CODE:

The screenshot shows the PyCharm IDE interface with the project 'network_client' open. The left sidebar displays the project structure with files 'Client.py' and 'Client.py' in the 'network_client' directory. The main editor window contains the following Python code:

```
#creat the client
import socket

c_socket=socket.socket(socket.AF_INET, socket.SOCK_STREAM)#this is an object for the client
c_socket.connect(('127.0.0.1', 9955)) #here we are not going to bind as we did in the server but we will connect it with the server using the method
#we used double brackets because this method takes only one argument so when using double brackets the tow arguments will appear as one
client_message=input('inter student id: ')#the client will enter the id
c_socket.send(client_message.encode())#client reseved a message from server// do not forget ()
fromServerToClient=c_socket.recv(1024).decode()#client reseved a message from server that OS will lock screen after 10 seconds or will not reseve
print(fromServerToClient)
c_socket.close()#it is very important to close this connection

#"C:/Program Files/Python312/python.exe" "c:/Users/HP/Desktop/cyber security/2.nd year/2.1sem/PYTHON/Client.Server/Client.py"
```

TCP SERVER CODE:

The screenshot shows the PyCharm IDE interface with the project 'network_server' open. The left sidebar displays the project structure with files 'Server.py' and 'Server.py' in the 'network_server' directory. The main editor window contains the following Python code:

```
import socket # the library that the sockets and related methods in and it is used to creat the server
import time # for the 10 seconds timer
import ctypes # the library that contains lock screen method

# defining the method that validates student id the message received will only cintain the 7 digits student id
usage
def howManyDigits(id): # this method checks if the id that was sent from client is 7 digits long
    count = len(
        str(id)) # we count the digits in many ways one of them is this built in function/count represents the number of digits
    return count == 7 # will return true if count ==7 else will return false

usage
def isValid(
    ID): # this method takes a student id as a parameter then decides if it is one of the 3 ids or not after checking if it is 7 didits
    check = howManyDigits(ID) # we implemented this method to save time
    groupIdList = [1221098, 1211047, 1202199]
    return check and ID in groupIdList # if number of digits is 7 and the ID is in the range of the three selected ids then return true else

# implementing the server
s = socket.socket(socket.AF_INET,
                  socket.SOCK_STREAM) # a socket object/socket.AF_INET specifies the address (and protocol) family, in this case, IPv4./socket.SOCK_STREAM specifies the type of socket (stream socket)
# just to know that the socket is created
print("socket created")
# listening on port
port = 9955
# binding the port the bind method taked 2 parameters the ip number and port.ip number is for local host but we can leave it empty to send an
# s.bind(('127.0.0.1',port)) # the method socket.gethostname() will get the local host ip address//the used ip adress means: the server to onl
print('socket binded to port') # to know that the socket is bound
```

```

30     # put the socket in listening mode
31     s.listen(5) # five is the number of connections the socket can handle
32     print("Server is listening")
33     # we will use an infinite loop until the student id is sent if not sent keep reseiving messages
34     while True:
35
36         c_socket, c_address = s.accept() # c:client socket , addr:client address, the server accepts the connection//.accept() returns two values
37         print(f'connected with client{c_address}')
38         receivedId = c_socket.recv(
39             1024).decode() # the method .recv(1024) reseives a msg from client and the argument declares the max amount of data reseved ,but client
40         studentId = int(receivedId)
41
42         # the method decode is to convert the msg from byte stream to character stream
43         idValidation = isValid(int(studentId)) # first convert from string to int then compare
44
45         if idValidation:
46             # these lines can be also implemented in idValidation function
47             msg = ('server will lock screen after 10 seconds') # the msg that the server will send
48             c_socket.send(
49                 msg.encode()) # this method send msg in form of bytes because of that we used encode method to convert the message
50             print(msg)
51             time.sleep(10) # the method .sleep is to make the server wait 10 seconds then continue
52             ctypes.windll.user32.LockWorkStation() # this is a built method from ctypes that locks a windows os computer
53             # close the connection but after or before shutting down the os?
54             c_socket.close()
55             s.close()
56             break
57
58         else:
59             print("Invalid student ID received.") # used print because this msg will appear on server side not client
60             c_socket.close()
61             s.close()

```

1:1 CRLF UTF-8 4 spaces Python 3.9 (network_server)

RUN CODE:

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Twitter> cd PycharmProjects
PS C:\Users\Twitter\PycharmProjects> cd network_server
PS C:\Users\Twitter\PycharmProjects\network_server> python Server.py
socket created
socket binded to port
server is listening
connected with client('127.0.0.1', 64793)

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

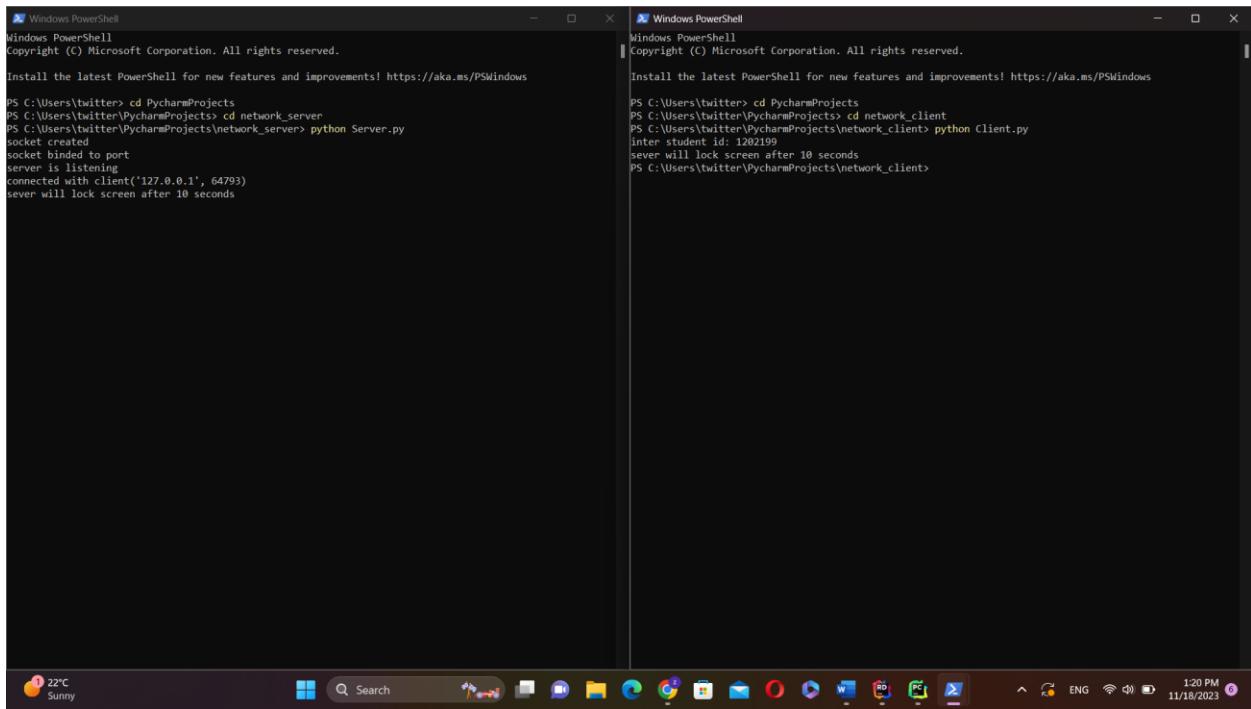
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Twitter> cd PycharmProjects
PS C:\Users\Twitter\PycharmProjects> cd network_client
PS C:\Users\Twitter\PycharmProjects\network_client> python Client.py
Enter student id: -

```

1:1 CRLF UTF-8 4 spaces Python 3.9 (network_server)

ENTER NUMBER FROM LIST:

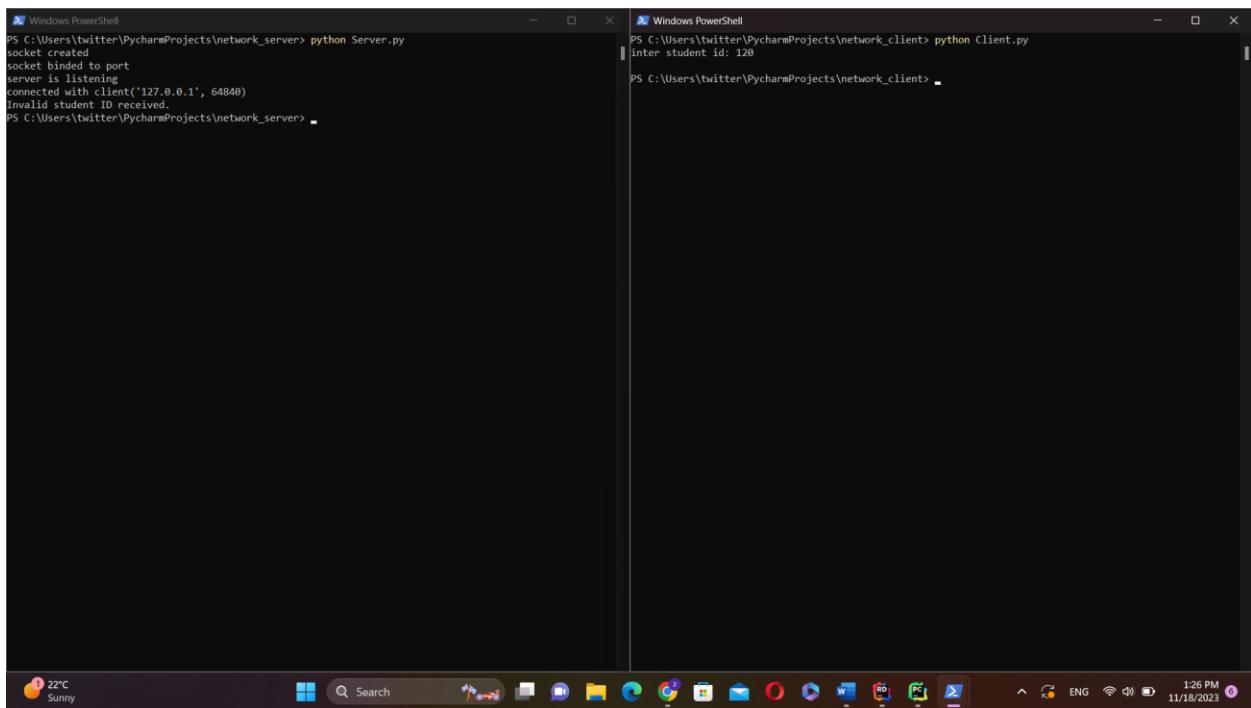


Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>
PS C:\Users\Twitter> cd PycharmProjects
PS C:\Users\Twitter\PycharmProjects> cd network_server
PS C:\Users\Twitter\PycharmProjects\network_server> python Server.py
socket created
socket binded to port
server is listening
Connected with client('127.0.0.1', 64793)
server will lock screen after 10 seconds

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>
PS C:\Users\Twitter> cd PycharmProjects
PS C:\Users\Twitter\PycharmProjects> cd network_client
PS C:\Users\Twitter\PycharmProjects\network_client> python Client.py
inter student id: 1202199
sever will lock screen after 10 seconds
PS C:\Users\Twitter\PycharmProjects\network_client>

After print this message the server will look the screen

ENTER NUMBER NOT IN LIST:



Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
PS C:\Users\Twitter\PycharmProjects\network_server> python Server.py
socket created
socket binded to port
server is listening
connected with client('127.0.0.1', 64840)
Invalid student ID received.
PS C:\Users\Twitter\PycharmProjects\network_server>

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
PS C:\Users\Twitter\PycharmProjects\network_client> python Client.py
inter student id: 1202199
PS C:\Users\Twitter\PycharmProjects\network_client>

PART III:

Python Code...

```
from socket import * # it is not same as import socket, if you use import socket you need to add a prefix to the function you want to use (socket.function)
host = '0.0.0.0' # (no particular address place holder)special ip address that means "any" or "all available interfaces," and also means I want to listen on all available IP
# A network interface is a connection point for connecting the computer to a network, and a machine can have multiple network interfaces (such as Ethernet, Wi-Fi, loopback, ...
# interface: a point of connection between computer and network
# 127.0.0.1(loopback address) it is used to connect to a server running on a local machine and accept connection from local machine. it only listen to loopback interface
serverPort = 9996 # Port number
serverSocket = socket(AF_INET, SOCK_STREAM) # TCP socket for incoming request
serverSocket.bind(("", serverPort)) # the server port number "serverPort" with this socket//empty ip address has the same meaning as 0.0.0.0
serverSocket.listen(1) # The server listen for TCP connection requests with 1 queued connections,//1 is the number of connections the socket can handle#bind=address is created
print("The web server is ready to receive") # Print a message to tell server is ready.

while True:
    connectionSocket, addr = serverSocket.accept() # When a client sends a TCP connection request create "connectionSocket"
    sentence = connectionSocket.recv(1024).decode() # the method .recv(1024) receives a msg from client and the argument declares the max amount of data reserved ,but client can send less than that
    print(addr) # client address and it is made of two parts:client ip address, port number used by the client for this connection
    print(sentence) # the msg received from client
    # how did you split it?
    ip = addr[0] # ip will be the client ip address
    port = addr[1] # this is the port number that the client is using for connection
    # how did you know that the object will be at [1]?
    object = sentence.split()[1] # Get the requested object from client//the object requested is the second word of the sentence
    print("\nThe HTTP request is:\n", object) # print the HTTP request//this message will appear on server side only

    if (
        object == '/' or object == '/index.html' or object == '/main_en.html' or object == '/en'):
        # if statement checks what is the requested object
        while True : elif (object.endswith('.jpg')):
```

```
object == '/' or object == '/index.html' or object == '/main_en.html' or object == '/en'): # if statement checks what is the requested object
# this is an http status line it is a common response when a requested resource is found, and the server is able to successfully fulfill the request
# Status codes are three-digit numbers that indicate the result of the HTTP request
connectionSocket.send(
    "HTTP/1.1 200 OK \r\n".encode()) # 1.1 is used http protocol vs //it sends an HTTP response with a "200 OK" which is a status code that means:received, understood
# http msg header that sets the content type to HTML with a character set of UTF-8// character set of UTF-8 covers almost all characters and symbols in the world
connectionSocket.send(
    "Content-Type: text/html; charset=UTF-8\r\n".encode()) # send the HTML file// \r\n is used as a line ending sequence and is used to delimit lines in HTTP message
connectionSocket.send(
    "\r\n".encode()) # end an additional \r\n sequence indicates the end of the headers section in the HTTP response
file1 = open("main_en.html",
            "rb") # server should send main_en.html file//this function will open the html file in binary mode so the content will be read in bytes
connectionSocket.send(
    file1.read()) # read the file that was open//read the file then send its content in a binary stream

elif (object == '/en'):
    # If object '/en', it will send the same HTTP status code and content type,
    # but will serve the contents of the file "main_en.html" instead. --> (sending a similar HTTP response with a different HTML file as the body).
    connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
    connectionSocket.send("Content-Type: text/html \r\n".encode())
    connectionSocket.send("\r\n".encode())
    file2 = open("main_en.html", "rb")
    connectionSocket.send(file2.read())

elif (object.endswith('.html')):
    # If the object ends with '.html', the server sends an HTTP response with a 200 OK status and a Content-Type
    # The server then opens the file "li.html" and sends its contents as the response body.
    connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
    connectionSocket.send("Content-Type: text/html; charset=UTF-8 \r\n".encode())
    connectionSocket.send("\r\n".encode())
    file3 = open("li.html", "rb")
    connectionSocket.send(file3.read())

while True : elif (object.endswith('.jpg')):
```

```
file3 = open("l1.html", "rb")
connectionSocket.send(file3.read())

...  
elif (object.endswith('.css')): # If the object ends with '.css', the server sends a similar HTTP response with
# the server then opens the file "styles.css" and sends its contents as the response body.
connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
connectionSocket.send("Content-Type: text/css; charset=UTF-8 \r\n".encode())
connectionSocket.send("\r\n".encode())
file4 = open("style.css", "rb")
connectionSocket.send(file4.read())

...  
elif (object == "/images/p1.png"):
connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
connectionSocket.send("Content-Type: image/png; charset=utf-8\r\n".encode())
connectionSocket.send("\r\n".encode())
f1 = open("Images/p1.png", "rb") # this is the way used to open non text files like imgs and html files
data = f1.read()
connectionSocket.send(data)

...  
elif (object == "/images/p2.jpg"):
connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
connectionSocket.send("Content-type: image/jpeg; charset=utf-8\r\n".encode())
connectionSocket.send("\r\n".encode())
f1 = open("Images/p2.jpg", "rb")
data = f1.read()
connectionSocket.send(data)

...  
elif (object == "/images/flag.png"):
connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
...  
while True : elif (object.endswith('.jpg')):  
    connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
```

```
connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
connectionSocket.send("Content-Type: image/png; charset=utf-8\r\n".encode())
connectionSocket.send("\r\n".encode())
f1 = open("Images/flag.png", "rb")
data = f1.read()
connectionSocket.send(data)

...  
elif (object.endswith('.png')): # files with the extensions '.png'
connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
connectionSocket.send("Content-Type: image/png \r\n".encode())
connectionSocket.send("\r\n".encode())
file5 = open("Images/brexit.png", "rb")
connectionSocket.send(file5.read())

...  
elif (object.endswith('.jpg')): # The same process occurs for '.jpg' files,
connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
connectionSocket.send("Content-Type: image/jpeg \r\n".encode())
connectionSocket.send("\r\n".encode())
file6 = open("Images/easy.jpg", "rb")
connectionSocket.send(file6.read())

...  
elif (object == '/cr'): # status code 307 Temporary Redirect: If the request is for '/cr',
# the Location header set to cornell
connectionSocket.send(
    "HTTP/1.1 307 Temporary Redirect \r\n".encode()) # http respons status line/307: represent a temporary redirect , telling the client that the requested resource has been temporarily moved
connectionSocket.send("Content-Type: text/html \r\n".encode())
connectionSocket.send("Location: https://cornell.edu \r\n".encode())
connectionSocket.send("\r\n".encode())

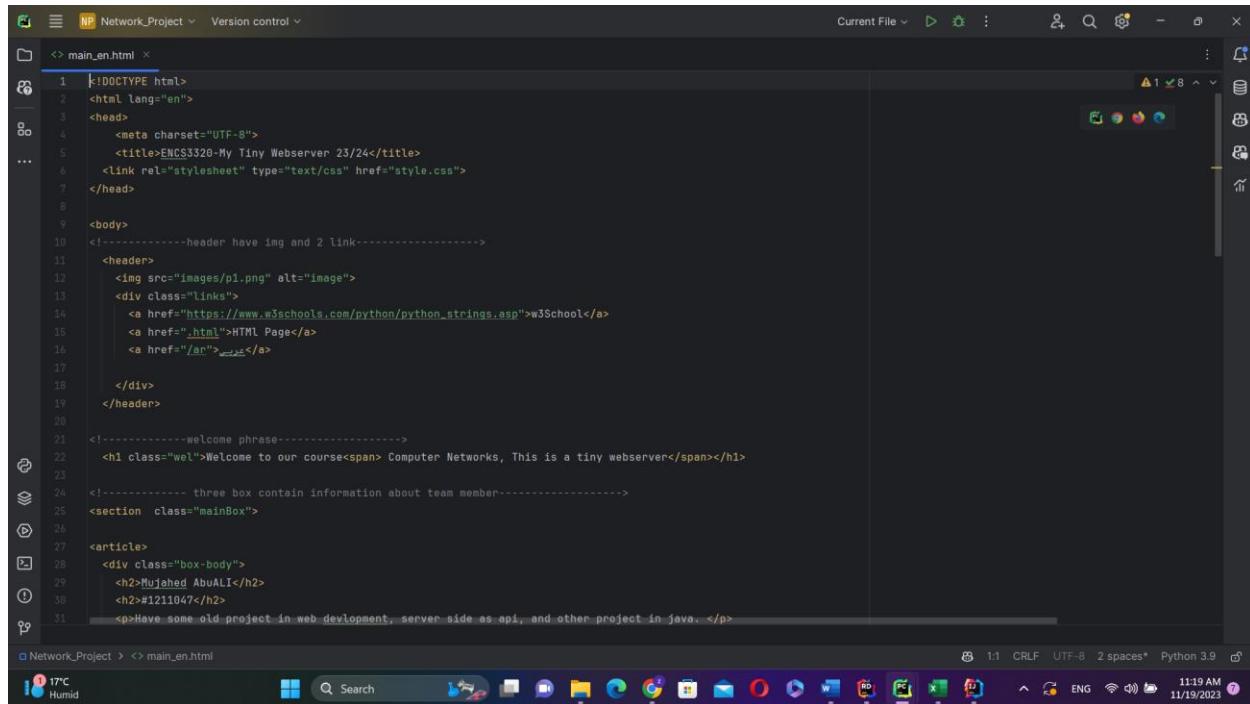
...  
elif (object == '/so'): # if the request is for '/so', the server sends a 307 Temporary Redirect
# instructing the client to make a new request to Stack Overflow.
...  
while True : elif (object.endswith('.jpg')):  
    connectionSocket.send("HTTP/1.1 200 OK \r\n".encode())
```

```
121     # instructing the client to make a new request to Stack Overflow.
122     connectionSocket.send("HTTP/1.1 307 Temporary Redirect \r\n".encode())
123     connectionSocket.send("Content-Type: text/html \r\n".encode())
124     connectionSocket.send("Location: https://stackoverflow.com \r\n".encode())
125     connectionSocket.send("\r\n".encode())
...
126
127 elif (object == '/rt'): # HTTP request that includes the string "/bzu" in the URL. If this string is present.
128     connectionSocket.send(
129         "HTTP/1.1 307 Temporary Redirect \r\n".encode()) # HTTP response to the client with a "307 Temporary Redirect" status code
130     connectionSocket.send("Content-Type: text/html \r\n".encode())
131     connectionSocket.send(
132         "Location: https://ritaj.birzeit.edu \r\n".encode()) # HTTP response with the location header set to birzeit.edu
133     connectionSocket.send("\r\n".encode())
134
135
136 else: # where a requested resource is not found .
137     connectionSocket.send(
138         "HTTP/1.1 404 Not Found \r\n".encode()) # the server sends a 404 Not Found HTTP response to the client
139     connectionSocket.send("Content-Type: text/html \r\n".encode()) # type text HTML
140     connectionSocket.send("\r\n".encode())
141     notFoundHtmlString = "<html> \
142         <head> \
143             <title>Error 404</title> \
144         </head> \
145         <body style='background-color: white;'> \
146             <div> \
147                 <p style='font-size: 30px; background-color: black; color:white; text-align: center; border-style: ridge; border-color: white; border-width: thin; font-family: georgia; text-align:center; color:Red;'> \
148                     Sorry The request is WRONG !!!!</p> \
149                     <p style='font-size: 45px; font-family: georgia; text-align:center; color:Red;'> \
150                         <strong> \
151                             The file is not found </strong> </p> \
152                         <pre style='font-size: 25px; font-family: arial ;text-align:center; color:Black'> <br> \
153                             <cb> Name: Mujahed AbuAli #122107 <br/> \
154                             <Name: Ziad magalma #1202199<br/><br/> \
155                             <Name: manal nidal #1221098<br/><br/> \
156                         </pre> \
157                         <pre style='font-size: 40px; font-family: arial;text-align:center; color:Black'><br> \
158                             f"IP: {ip}" \
159                             f"Port: {port}" \
160                         </pre> \
161                         </div> \
162                         </body> \
163                         </html>" \
164
165     # if the request is wrong or the file doesnt exist the server should return a simple HTML webpage that contains with
166     # some design and color needed
167     notFoundHtmlBytes = bytes(notFoundHtmlString, "UTF-8")
168     connectionSocket.send(notFoundHtmlBytes)
169
170
171 connectionSocket.close() # close the connection
172
...
173
174 while True : elif (object.endswith('jpg'))
```

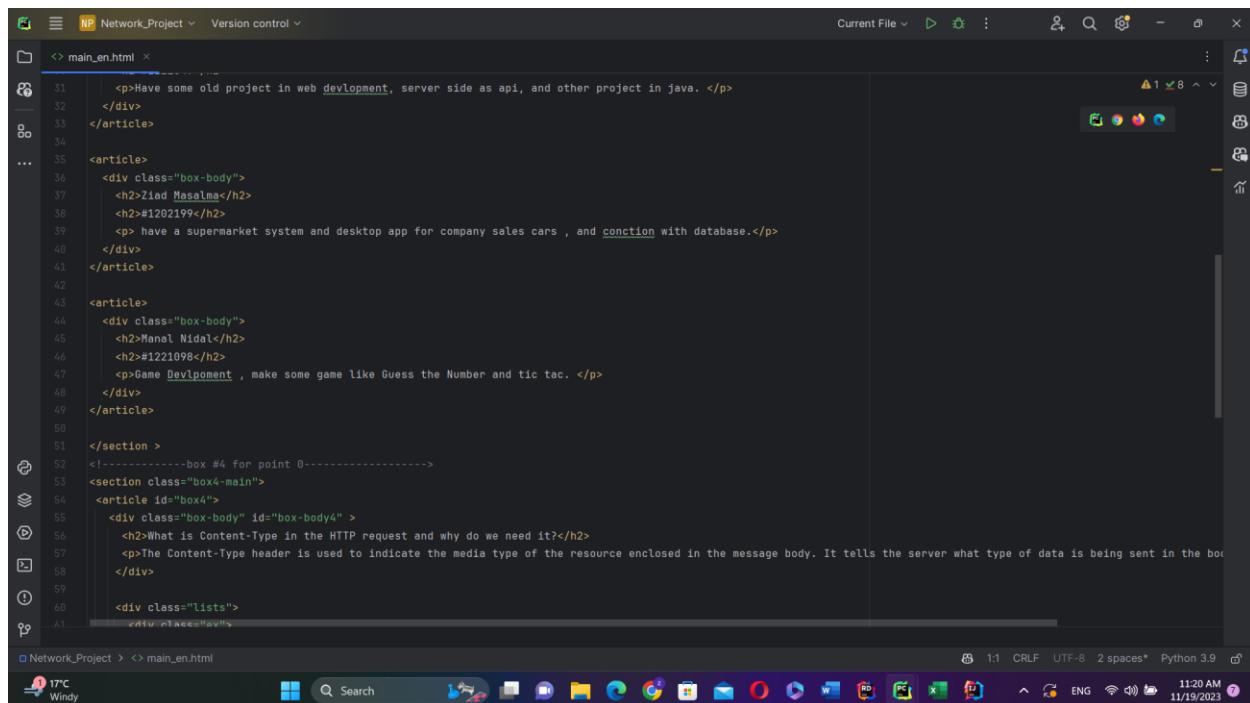
```
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
...
173
174 while True : elif (object.endswith('jpg'))
```

HTML CODE:

main_en.html:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>ENC33320-My Tiny Webserver 23/24</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
<!--header have img and 2 link-->
<header>
    
    <div class="links">
        <a href="https://www.w3schools.com/python/python_strings.asp">w3School</a>
        <a href="#_html">HTML Page</a>
        <a href="/en">English</a>
    </div>
</header>
<!--welcome phrase-->
<h1 class="wel">Welcome to our course<span> Computer Networks, This is a tiny webserver</span></h1>
<!-- three box contain information about team member-->
<section class="mainBox">
<article>
    <div class="box-body">
        <h2>MuJahed AbuAli</h2>
        <h2>#1211047</h2>
        <p>Have some old project in web development, server side as api, and other project in java. </p>
    </div>
</article>
<article>
    <div class="box-body">
        <h2>Ziad Maslme</h2>
        <h2>#1202199</h2>
        <p> have a supermarket system and desktop app for company sales cars , and connection with database.</p>
    </div>
</article>
<article>
    <div class="box-body">
        <h2>Manal Nidal</h2>
        <h2>#1221098</h2>
        <p>Game Development , make some game like Guess the Number and tic tac. </p>
    </div>
</article>
</section >
<!--box #4 for point 0-->
<section class="box4-main">
<article id="box4">
    <div class="box-body" id="box-body4" >
        <h2>What is Content-Type in the HTTP request and why do we need it?</h2>
        <p>The Content-Type header is used to indicate the media type of the resource enclosed in the message body. It tells the server what type of data is being sent in the body</p>
    </div>
</article>
</section >
```



```
<p>Have some old project in web development, server side as api, and other project in java. </p>
</div>
</article>
<article>
    <div class="box-body">
        <h2>Ziad Maslme</h2>
        <h2>#1202199</h2>
        <p> have a supermarket system and desktop app for company sales cars , and connection with database.</p>
    </div>
</article>
<article>
    <div class="box-body">
        <h2>Manal Nidal</h2>
        <h2>#1221098</h2>
        <p>Game Development , make some game like Guess the Number and tic tac. </p>
    </div>
</article>
</section >
<!--box #4 for point 0-->
<section class="box4-main">
<article id="box4">
    <div class="box-body" id="box-body4" >
        <h2>What is Content-Type in the HTTP request and why do we need it?</h2>
        <p>The Content-Type header is used to indicate the media type of the resource enclosed in the message body. It tells the server what type of data is being sent in the body</p>
    </div>
</article>
</section >
```

Network_Project Network_Project Version control

main_en.html

```

61     <div class="ex">
62         <p>For example:</p>
63         <ul>
64             <li>text/plain: Plain text data.</li>
65             <li>text/html: HTML document.</li>
66             <li>application/json: JSON data.</li>
67             <li>image/jpeg: JPEG image.</li>
68         </ul>
69     </div>
70
71     <div class="ex">
72         <p>Its used for:</p>
73         <ul>
74             <li>Server Processing.</li>
75             <li>Interpretation by Recipient.</li>
76             <li>Security Measures.</li>
77             <li>Content Negotiation.</li>
78         </ul>
79     </div>
80 </div>
81
82 </article>
83 </section>
84 <!-------jpg image----->
85 <div class="center-img">
86     
87 </div>
88 <!-------footer----->
89 <footer>
90     <p>Copyright 2025 ©</p>
91 </footer>

```

Network_Project main_en.html

Top events Event brief

1:1 CRLF UTF-8 2 spaces* Python 3.9

11:21 AM 11/19/2023

main_ar.html:

Network_Project Network_Project Version control

main_ar.html

```

1 <!DOCTYPE html>
2 <html lang="ar">
3     <head>
4         <meta charset="UTF-8">
5         <title>ENCS3320-My Tiny Webserver 23/24</title>
6         <link rel="stylesheet" type="text/css" href="style.css">
7     </head>
8
9     <body>
10    <!-------header have img and 2 link----->
11    <header>
12        
13        <div class="links">
14            <a href="https://www.w3schools.com/python/python_strings.asp">w3School</a>
15            <a href="#">HTML Page</a>
16            <a href="#">English</a>
17        </div>
18    </header>
19
20    <!-------welcome phrase----->
21    <h1 class="wel">مرحباً بكم في مدرسة<span>مكتبة</span></h1>
22
23    <!------- three box contain information about team member----->
24    <section class="mainBox">
25
26        <article>
27            <div class="box-body">
28                <h2>ابراهيم</h2>
29                <h2>#12110647</h2>
30                <p>يعمل المعاذري في تطوير الويب، وحانس الخادم كواجهة برمجة التطبيقات، ويشغله آخر في جناد</p>
31            </div>

```

Network_Project main_ar.html

17C Windy

1:1 CRLF UTF-8 2 spaces* Python 3.9

11:23 AM 11/19/2023

Screenshot of a code editor showing the main_ar.html file. The code is in Arabic and discusses various types of data and processing steps. The editor interface includes tabs for Network_Project and main_ar.html, a search bar, and a status bar at the bottom.

```
</div>
</article>
<article>
<div class="box-body">
<h2>بيانات</h2>
<h2>#1202199</h2>
<p>بيانات ملخصة، وبيانات مفصلة، وبيانات معمقة، وبيانات متقدمة</p>
</div>
</article>
<article>
<div class="box-body">
<h2>بيانات</h2>
<h2>#1221098</h2>
<p>بيانات المدخلات، اصنف بعض الالعاب مثل لعبة اغنية الفقير وبيانات</p>
</div>
</article>
</section >
<!-------box #4 for point 0----->
<section class="box4-main">
<article id="box4">
<div class="box-body" id="box-body4" >
<h2 lang="ar">ما هو نوع المحتوى في طلب</h2>
<p>يمكن استخدام رأس نوع المحتوى للإشارة إلى نوع البيانات المطلوبة في طلب</p>
<div class="lists">
<div class="ex">
<p>بيانات</p>

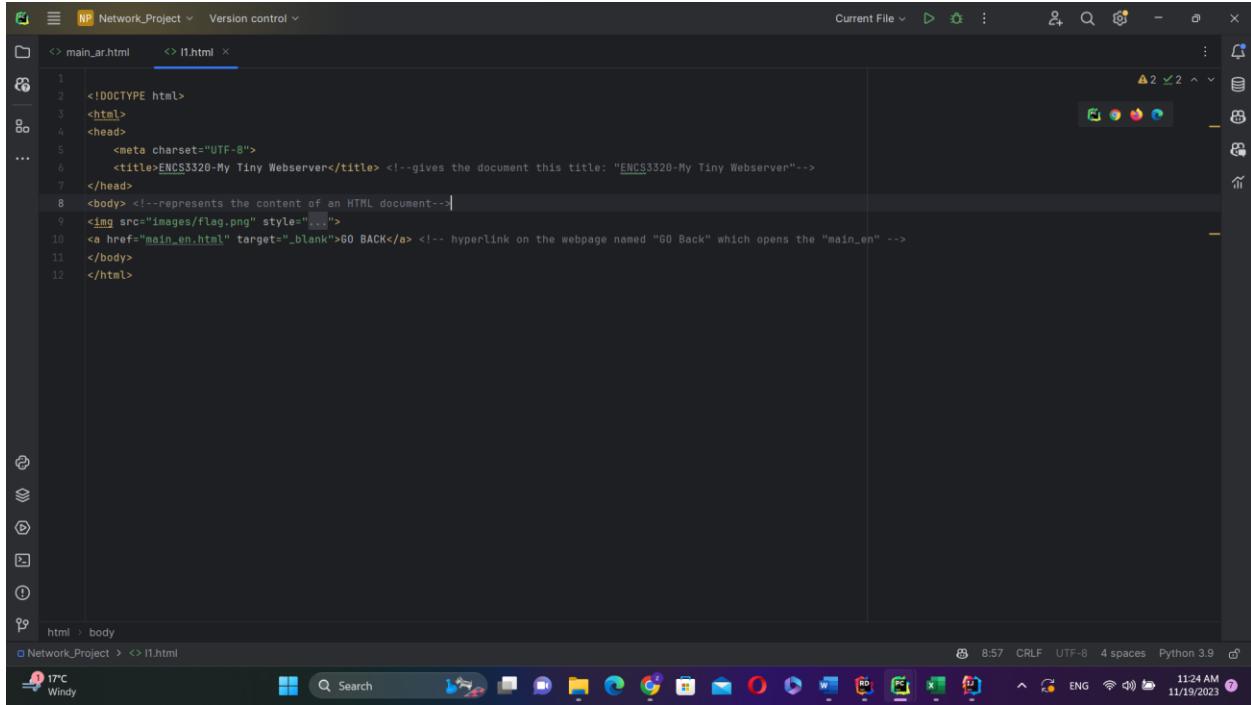
```

Screenshot of a code editor showing the main_ar.html file. The code continues from the previous snippet, focusing on data types and processing. The editor interface is identical to the first screenshot.

```
<ul>
<li>text/plain: Plain text data.</li>
<li>text/html: HTML document.</li>
<li>application/json: JSON data.</li>
<li>image/jpeg: JPEG image.</li>
</ul>
</div>
<div class="ex">
<p lang="ar">رسالة</p>
<ul>
<li>Server Processing.</li>
<li>Interpretation by Recipient.</li>
<li>Security Measures.</li>
<li>Content Negotiation.</li>
</ul>
</div>
</div>
</article>
</section>
<!-------jpg image----->
<div class="center-img">

</div>
<!-------footer----->
<footer>
<p>Copyright 2023 @</p>
</footer>
```

L1.html:



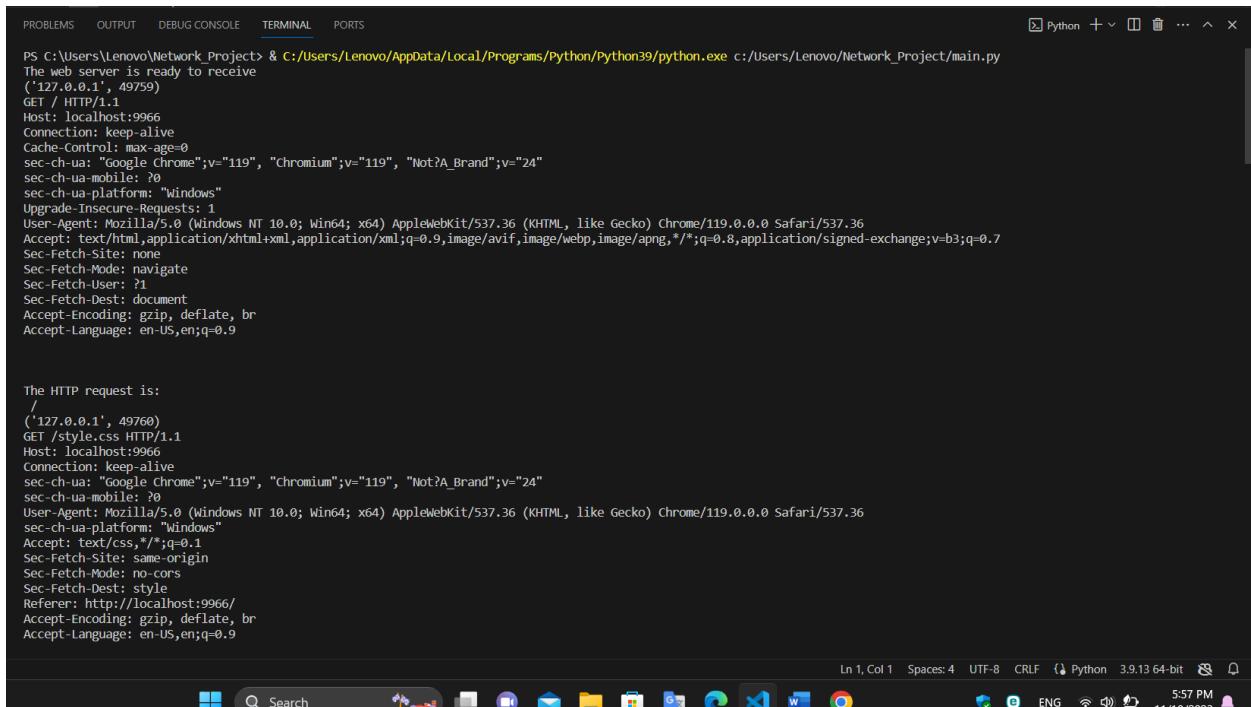
The screenshot shows a code editor window with the file `L1.html` open. The code is as follows:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>ENCS3320-My Tiny Webserver</title> <!--gives the document this title: "ENCS3320-My Tiny Webserver"-->
</head>
<body> <!--represents the content of an HTML document-->

<a href="#main_en.html" target="_blank">GO BACK</a> <!-- hyperlink on the webpage named "GO Back" which opens the "main_en" -->
</body>
</html>
```

The code includes a meta tag for character encoding, a title, an image reference, and a link to another page.

Screenshots in the Terminal for all requests:



The screenshot shows a terminal window with several lines of text representing HTTP requests. The requests are from different clients, each with a unique User-Agent string. The terminal also shows the server's response to these requests.

```
PS C:\Users\Lenovo\Network Project & C:/Users/lenovo/AppData/Local/Programs/Python/Python39/python.exe c:/Users/lenovo/Network Project/main.py
The web server is ready to receive
('127.0.0.1', 49759)
GET / HTTP/1.1
Host: localhost:9966
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Google Chrome";v="119", "Chromium";v="119", "Not?A_Brand";v="24"
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/119.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
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

The HTTP request is:
/
('127.0.0.1', 49760)
GET /style.css HTTP/1.1
Host: localhost:9966
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="119", "Chromium";v="119", "Not?A_Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:9966/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

Screenshots in the browser for all requests:

localhost:9966 or localhost:9966/index.html or /main_en.html

The screenshot shows a web browser window with the URL `localhost:9966/main_en.html`. The page features a green molecular structure logo at the top left. The main content area has a light blue background with the heading "Welcome to our course Computer Networks, This is a tiny webserver". Below this, there are three grey boxes containing student profiles:

- Mujahed AbuALI**
#1211047
Have some old project in web development, server side as api, and other project in java.
- Ziad Masalma**
#1202199
have a supermarket system and desktop app for company sales cars , and conction with database.
- Manal Nidal**
#1221098
Game Devlpoment , make some game like Guess the Number and tic tac.

In the center of the page is a purple callout box with the question "What is Content-Type in the HTTP request and why do we need it?". Below it, a smaller text states: "The Content-Type header is used to indicate the media type of the resource enclosed in the message body. It tells the server what type of data is being sent in the body of the request." The browser's taskbar at the bottom shows various pinned icons and the date/time as 11/18/2023.

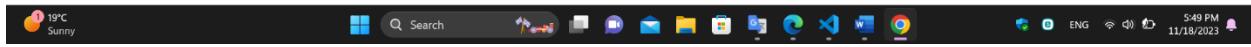
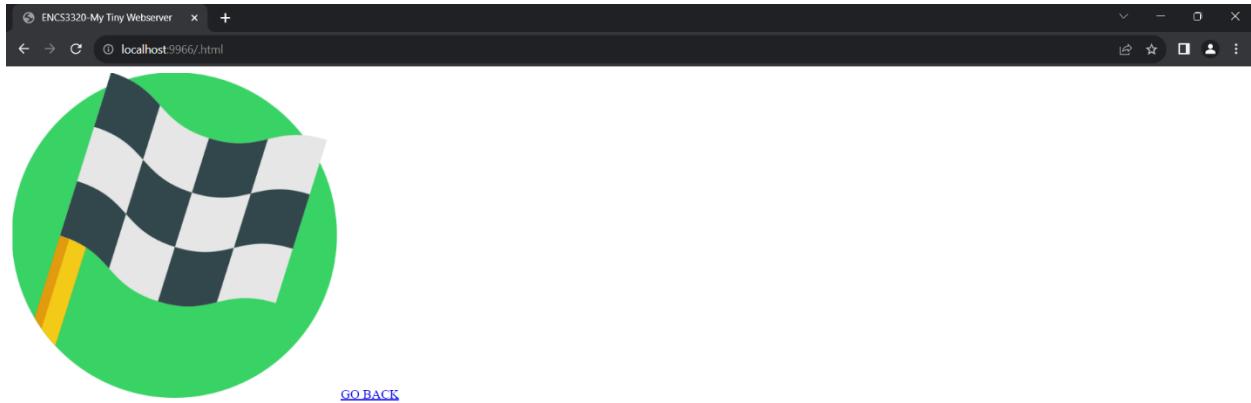
localhost:9966/ar

Enter to Arabic page.

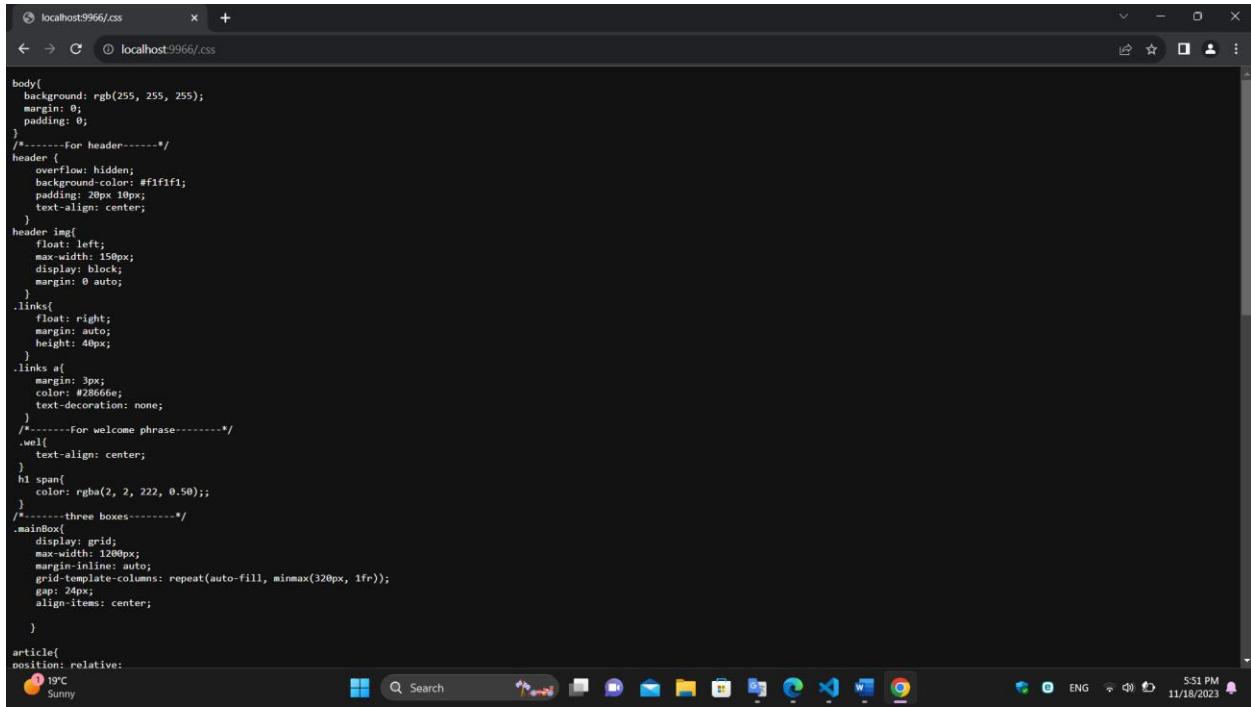
The screenshot shows a web browser window with the URL `localhost:9966/ar`. The page layout is identical to the English version, featuring the green molecular structure logo and the same three student profiles. The central callout box also contains the same question and explanation about the Content-Type header.

Below the profiles, the central callout box now asks "ما هو نوع المحتوى في طلب HTML ولماذا نحتاج اليه؟" (What is the content type in an HTML request and why do we need it?). The explanatory text remains the same: "يتم استخدام رأس نوع المحتوى لإشارة إلى نوع الوسائط للمورد الموجود في نفس الرسالة. يخبر الخادم نوع البيانات التي يتم إرسالها في نفس الطلب". The browser's taskbar at the bottom shows various pinned icons and the date/time as 11/18/2023.

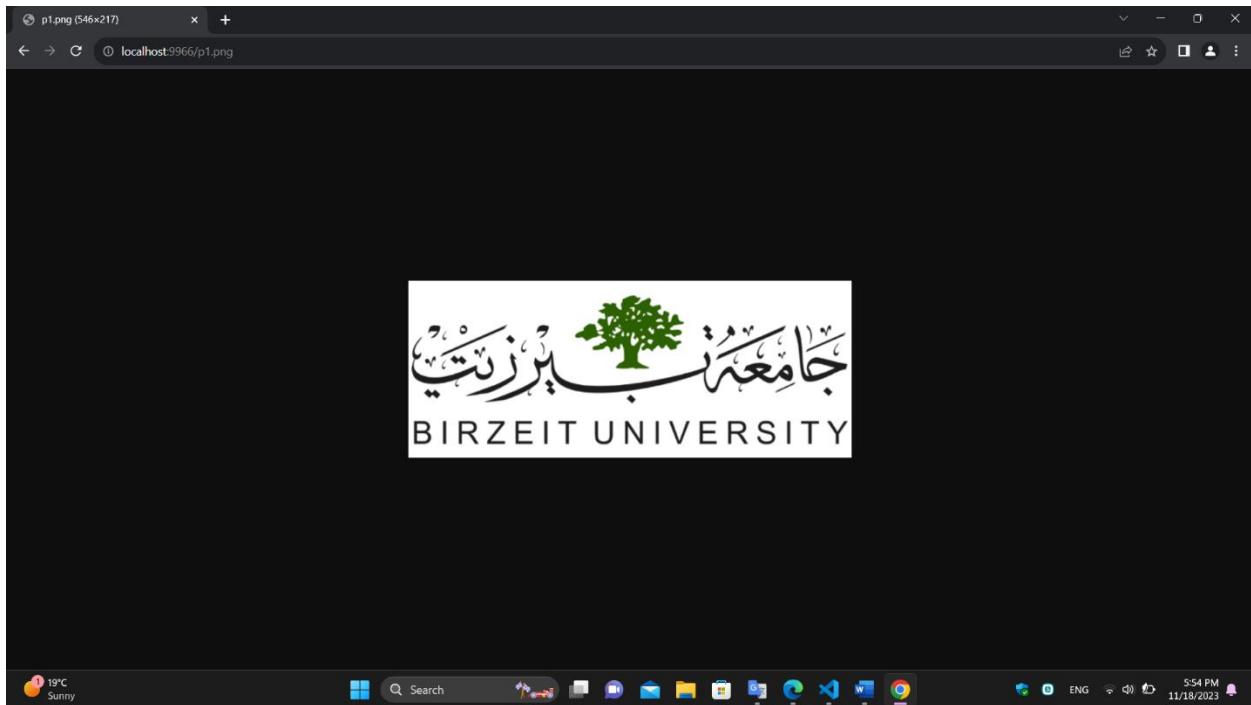
Localhost:9966/l1.html



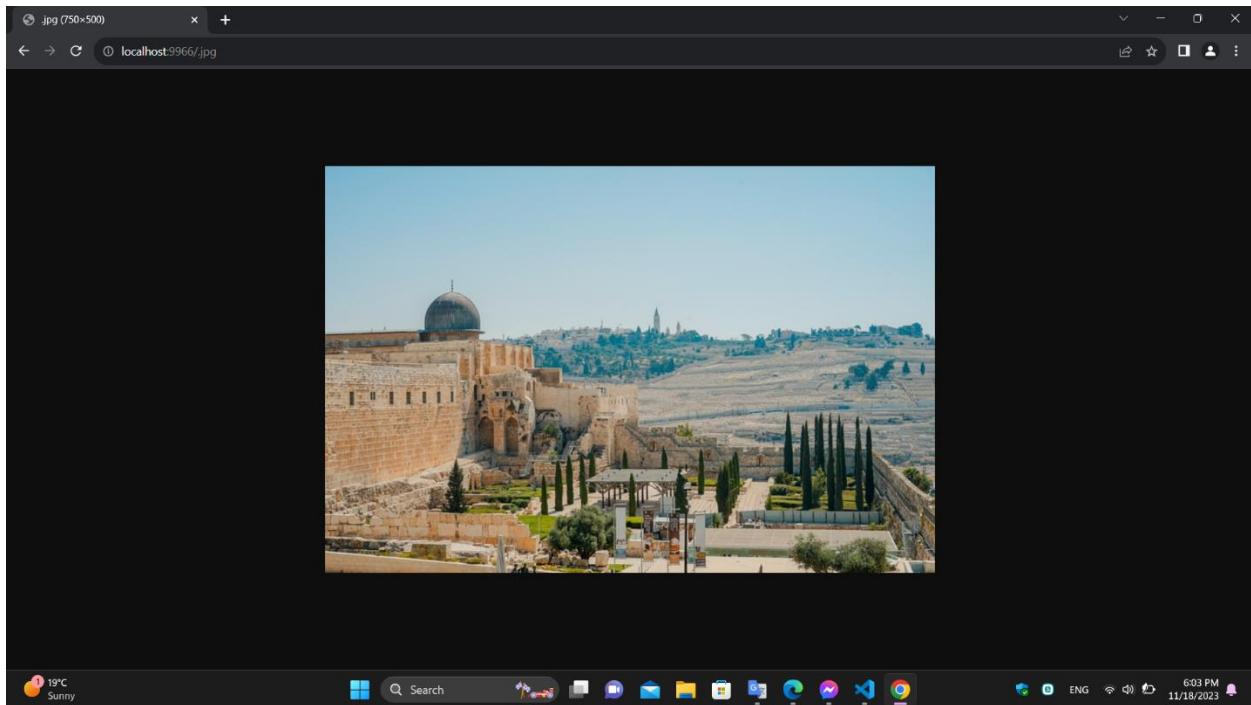
Localhost:9966/.css



Localhost:9966/p1.png



Localhost:9966/aqsa.jpg



localhost:9966/error

Wrong request



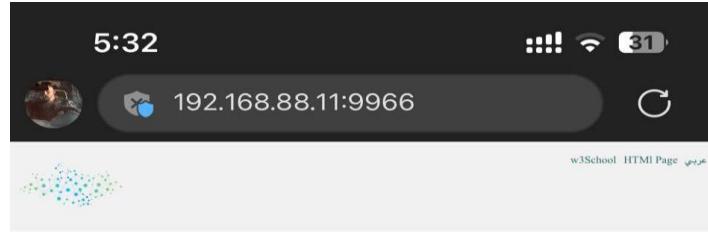
The file is not found

Name: Mujahed Abuali #121107
Name: Ziad masalma #1202199
Name: manal nidal #1221098

IP: 127.0.0.1 Port: 49707



Screenshots from phone:



Welcome to our course Computer Networks, This is a tiny webserver

A screenshot of a mobile device displaying a list of three students. Each student has a profile picture, name, ID number, and a brief description of their projects.

Mujahed AbuALI

#1211047

Have some old project in web development, server side as api, and other project in java.

Ziad Masalma

#1202199

have a supermarket system and desktop app for company sales cars , and conction with database.

Manal Nidal

#12221098

Game Devlopment , make some game like Guess the Number and tic tac.

What is Content-Type in the HTTP request and why do we need it?

The Content-Type header is used to indicate the media type of the resource enclosed in the message body. It tells the server what type of data is being sent in the body of the request.

For example:

- text/plain: Plain text data.
- text/html: HTML document.
- application/json: JSON data.
- image/jpeg: JPEG image.

Its used for:

- Server Processing.
- Interpretation by Recipient.
- Security Measures.
- Content Negotiation.

