BIRZEIT UNIVERSITY

Faculty of Engineering & Technology
Electrical & Computer Engineering Department

ENCS3320

COMPUTER NETWORKS

(Summer Semester – 2023/2024)

Project #1

Socket Programming

Prepared by:
Suhaib Sawalha 1220251 (sec.2)
Abd Alhameed Maree 1220775 (sec.1)

Instructor: Dr. Ibrahim Nemer

Date: 12/8/2024

Abstract

The project is divided into three parts: the first part aims to understand some of the commands that can give important information about the network, routers, paths and IPs, and try the Wireshark software, the second part aims to understand the socket programming and implement it using TCP and UDP, and implement the idea of peering, the final part aims to understand how to control web server and HTTP requests, and create basic pages using HTML and CSS.

Table of Contents

List of Figures

List of Tables

Theory and Procedure

Command & Wireshark

Every system is connected to numerous different networks and systems through internal or external network channels. These network settings often run into issues and affect the system's working. Such network problems can be resolved using 'networking commands.'

These commands are specifically designed to troubleshoot network problems with minimum complexity using the windows command prompt tool [1].

The following commands are the ones which will be used in this project:

- PING

The Ping command is one of the most widely used commands in the prompt tool, as it allows the user to check the connectivity of our system to another host.

This command sends four experimental packets to the destination host to check whether it receives them successfully, if so, then, we can communicate with the destination host. But in case the packets have not been received, that means, no communication can be established with the destination host [1].

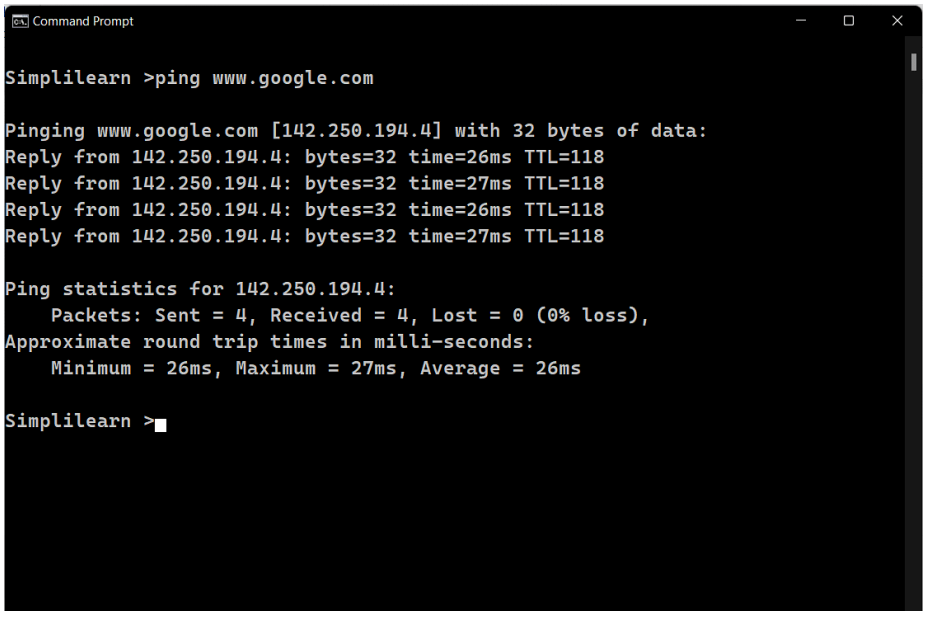

Figure 1: Ping Command.

- TRACERT

The TRACERT command is used to trace the route during the transmission of the data packet over to the destination host and also provides us with the “hop” count during transmission.

Using the number of hops and the hop IP address, we can troubleshoot network issues and identify the point of the problem during the transmission of the data packet [1].

Figure 2: Tracert Command.

- NSLOOKUP

The NSLOOKUP command is used to troubleshoot network connectivity issues in the system. Using the nslookup command, it can access the information related to the system’s DNS server, i.e., domain name and IP address [1].

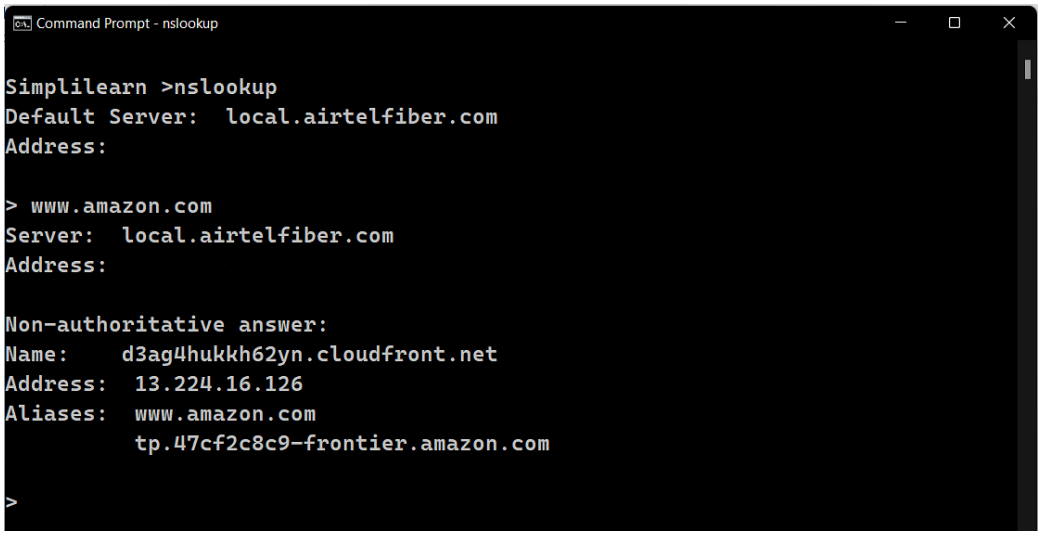

Figure 3: Nslookup Command.

- TELNET

Telnet is a network protocol, used on the Internet or Local Area Networks, that provides a bidirectional interactive text-oriented communications facility using a virtual terminal connection.

Telnet command can be used to connect to a port on a remote server to verify if the path from the computer to that server is open over that port. Telnet was originally built to remotely control and manage mainframe computers from distant terminals. It has largely been replaced by other technologies for secure remote control of computers, because Telnet sends data in plain text [2].

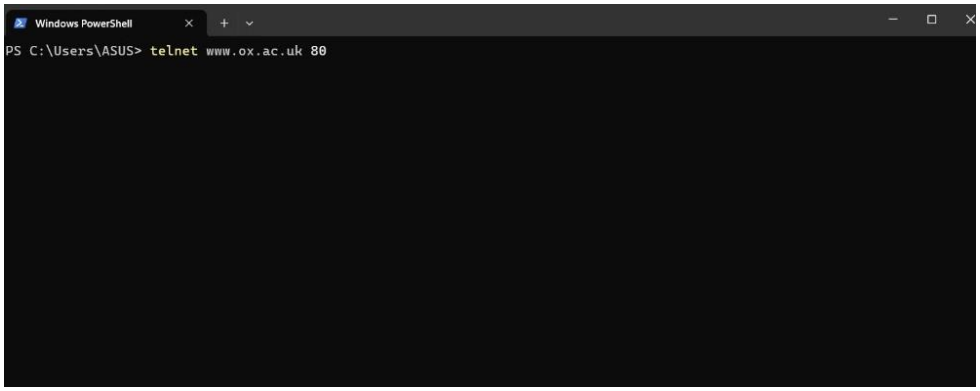

Figure 4: Telnet Command.

Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.

Network packet analyzer is a measuring device for examining what is happening inside a network cable, just like an electrician uses a voltmeter for examining what is happening inside an electric cable (but at a higher level, of course) [3].

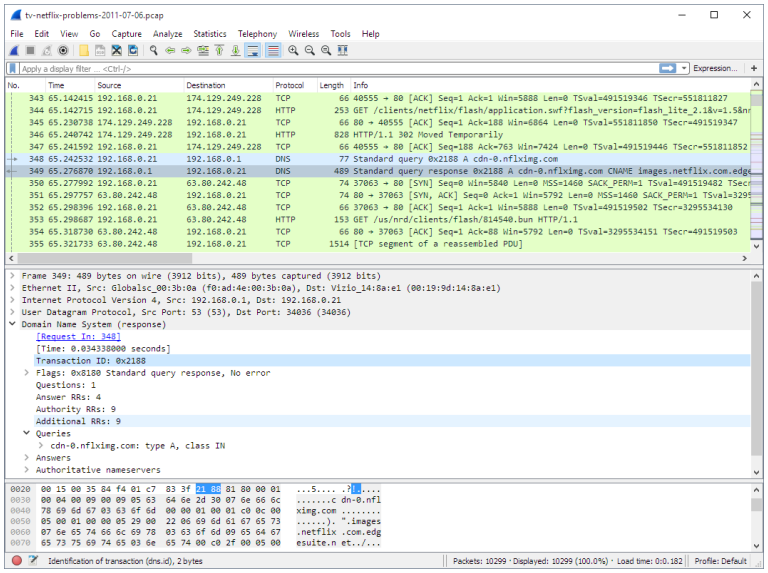

Figure 5: Wireshark Interface.

Socket Programming (TCP and UDP)

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket (node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server [4].

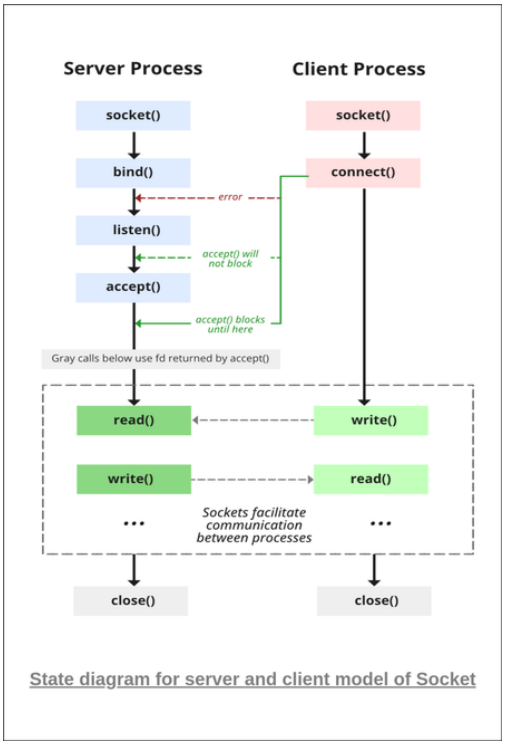

Figure 6: State Diagram for Server and Client Model.

TCP

Transmission Control Protocol (TCP) is a communications standard that enables application programs and computing devices to exchange messages over a network. It is designed to send packets across the internet and ensure the successful delivery of data and messages over networks.

TCP is one of the basic standards that define the rules of the internet and is included within the standards defined by the Internet Engineering Task Force (IETF). It is one of the most commonly used protocols within digital network communications and ensures end-to-end data delivery.

TCP organizes data so that it can be transmitted between a server and a client. It guarantees the integrity of the data being communicated over a network. Before it transmits data, TCP establishes a connection between a source and its destination, which it ensures remains live until communication begins. It then breaks large amounts of data into smaller packets, while ensuring data integrity is in place throughout the process.

As a result, high-level protocols that need to transmit data all use TCP Protocol. Examples include peer-to-peer sharing methods like File Transfer Protocol (FTP), Secure Shell (SSH), and Telnet. It is also used to send and receive email through Internet Message Access Protocol (IMAP), Post Office Protocol (POP), and Simple Mail Transfer Protocol (SMTP), and for web access through the Hypertext Transfer Protocol (HTTP).

An alternative to TCP in networking is the User Datagram Protocol (UDP), which is used to establish low-latency connections between applications and decrease transmissions time. TCP can be an expensive network tool as it includes absent or corrupted packets and protects data delivery with controls like acknowledgments, connection startup, and flow control.

UDP does not provide error connection or packet sequencing nor does it signal a destination before it delivers data, which makes it less reliable but less expensive. As such, it is a good option for time-sensitive situations, such as Domain Name System (DNS) lookup, Voice over Internet Protocol (VoIP), and streaming media [5].

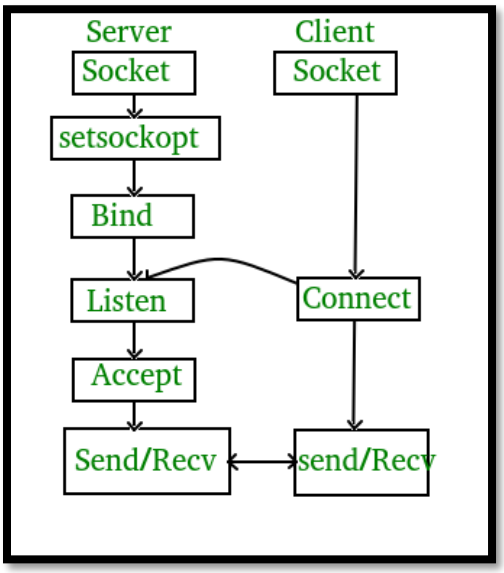

Figure 7: TCP socket programming.

UDP

User Datagram Protocol (UDP) is a communications protocol for time-sensitive applications like gaming, playing videos, or Domain Name System (DNS) lookups. UDP results in speedier communication because it does not spend time forming a firm connection with the destination before transferring the data. Because establishing the connection takes time, eliminating this step results in faster data transfer speeds.

However, UDP can also cause data packets to get lost as they go from the source to the destination. It can also make it relatively easy for a hacker to execute a distributed denial-of-service (DDoS) attack.

In many cases, particularly with Transmission Control Protocol (TCP), when data is transferred across the internet, it not only has to be sent from the destination but also the receiving end has to signal that it is ready for the data to arrive. Once both of these aspects of the communication are fulfilled, the transmission can begin. However, with UDP, the data is sent before a connection has been firmly established. This can result in problems with the data transfer, and it also presents an opportunity for hackers who seek to execute DDoS attacks [6].

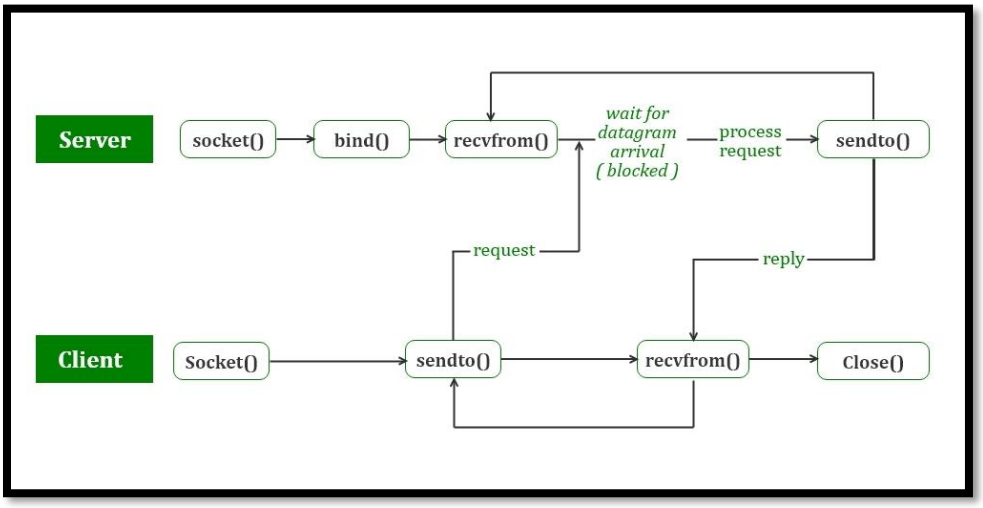

Figure 8: UDP socket programming.

The main difference between TCP (transmission control protocol) and UDP (user datagram protocol) is that TCP is a connection-based protocol and UDP is connectionless. While TCP is more reliable, it transfers data more slowly. UDP is less reliable but works more quickly. This makes each protocol suited to different types of data transfers [7].

Peer-to-Peer

A peer-to-peer network is an information technology (IT) infrastructure allowing two or more computer systems to connect and share resources without requiring a separate server or server software Workplaces may set up a P2P network by physically connecting computers into a linked system or creating a virtual network. It can be also to set up computers to be the clients and servers of their network.

A P2P network is different from a client-server network traditionally used in networking. A client-server network is a connection between a client computer and a server computer to provide the client with the server's resources.

In the P2P network, each device is considered a peer—thus "peer-to-peer"—with functions that contribute to the network. Each computer is both a client and a server and they share resources with other networked computers [8].

Python is used for socket programming in this project, as it has a ‘socket’ library which is rich of components, methods and easy to use and understand.

Table 1: Brief Explanation of some socket programming python methods.

Method	Brief Explanation
socket()	Creates a new socket object that can be used to send and receive data.
.bind()	Binds the socket to a specific network interface and port number.
.listen()	Puts the socket into listening mode, ready to accept incoming connections.
.accept()	Accepts an incoming connection, returning a new socket object and the address of the client.
.connect()	Initiates a connection to a remote socket at the specified address.
.send()	Sends data through the socket to the connected remote socket.
.recv()	Receives data from the socket, blocking until data is available.
.recvfrom()	Receives data from a socket along with the address of the sender. It is typically used with UDP sockets and returns a tuple containing the received data and the sender's address.
.close()	Closes the socket, terminating the connection and freeing the resources.
.encode()	Converts a string into bytes using a specified encoding (e.g., UTF-8), which is necessary before sending data over a socket.
.decode()	Converts bytes back into a string using a specified encoding, which is typically done after receiving data from a socket.

The methods in Table (1) are the methods that are used in this project.

Web Server

A web server is a server. A server is a process serving clients. Surprisingly or not, a server has nothing to do with hardware. It's just a regular piece of software run by an operating system. Like most other programs around, a server gets some data on its input, transforms data in accordance with some business logic, and then produces some output data. In the case of a web server, the input and output happen over the network via Hypertext Transfer Protocol (HTTP). For a web server, the input consists of HTTP requests from its clients - web browsers, mobile applications, IoT devices, or even other web services. And the output consists of HTTP responses, oftentimes in form of HTML pages, but other formats are also supported [9].

Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers, but it can also be used for other purposes. HTTP follows a classical client-server model, with a client opening a connection to make a request, then waiting until it receives a response. HTTP is a stateless protocol, meaning that the server does not keep any data (state) between two requests [10].

The pages are built using HTML and CSS.

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web [11].

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, the development of various parts of CSS specification was done synchronously, which allowed the versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, or even CSS3. There will never be a CSS3 or a CSS4; rather, everything is now just "CSS" with individual CSS modules having version numbers [12].

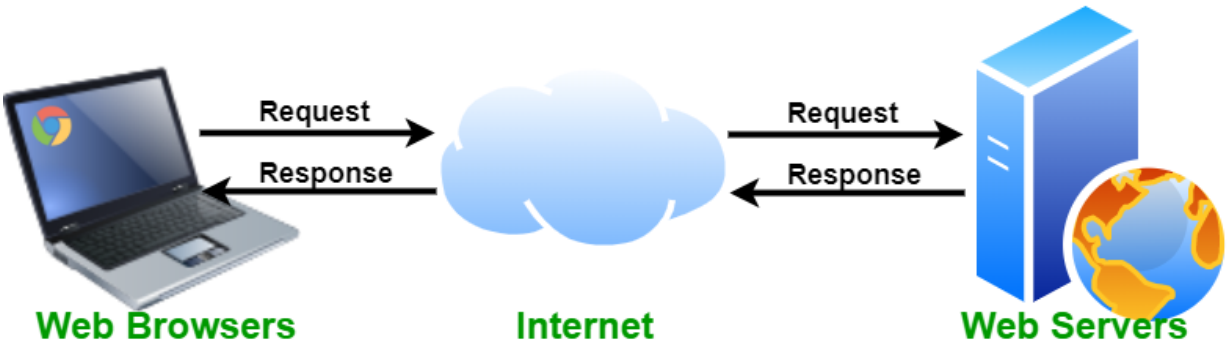

Figure 9: Requests between browser and server.

The pages and the HTTP requests are then controlled by the server sockets with the same methods that are used in the socket programming part.

Displaying pages depends on the URL (Uniform Resource Locator) of the page which is sent by HTTP request that get the response in different forms:

Table 2: Response Types for HTTP request.

Response Type	Meaning
Content-Type: text/html	HTML page that translated into GUI (Graphical User Interface) using browser.
Content-Type: text/css	CSS text file
Content-Type: text/png	PNG image
Content-Type: text/jpg	JPG image

Results and Discussions

Commands & Wireshark

1) Describing ping, tracert, nslookup, and telnet.
    - ping: sends request over the network to a specific device, the request is small packets of data to the target IP. It tests the reachability of a host; it also measures the round-trip time (RTT).
    - tracert: sends packets to the destination IP, it tracks the path of the packets; It shows the IP addresses of each router the packet passes through and the time it takes to get to each one.
    - nslookup: contacts a DNS server to resolve the domain name into its corresponding IP address or vice versa. It queries DNS servers to find the IP address of a domain name and vice versa.
    - telnet: network protocol that allows to connect to remote servers or devices over a network; It establishes a text-based session to interact with the server.

2) Trying the commands:
    a)

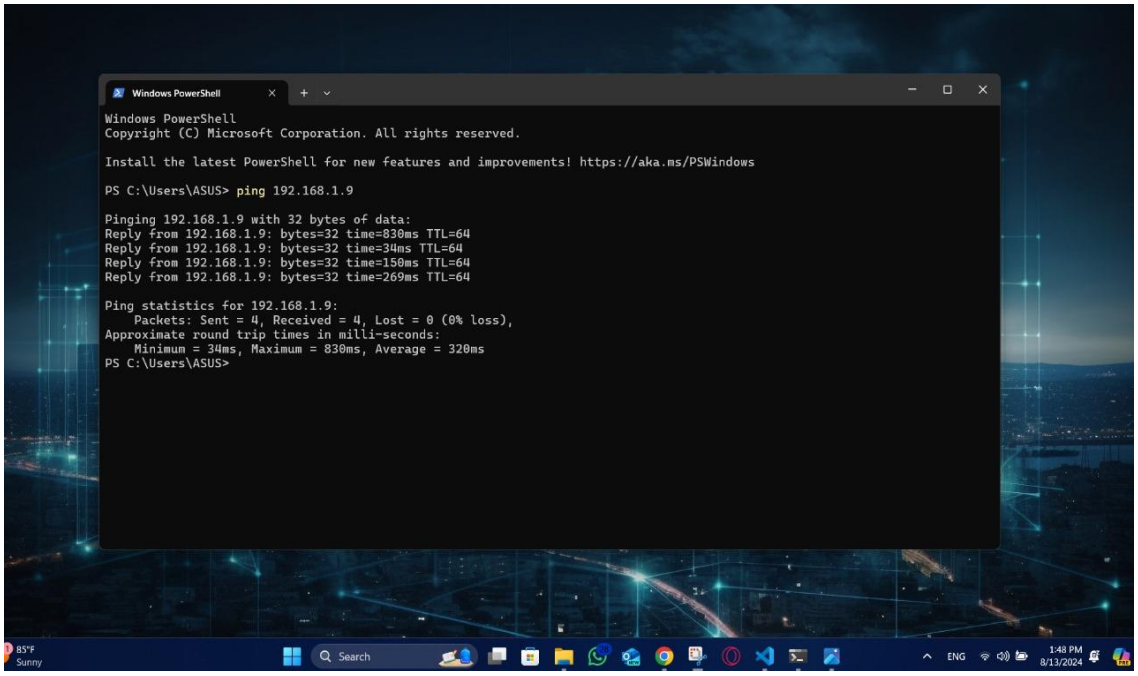

Figure 10: ping a device in the same network.

To ping a device in the same network, IP address is needed, the device in Fig (10) is another laptop, the IP address was taken from the settings and run the ping command to it. Ping showed path of the packets sent to that device and the times taken.

b)

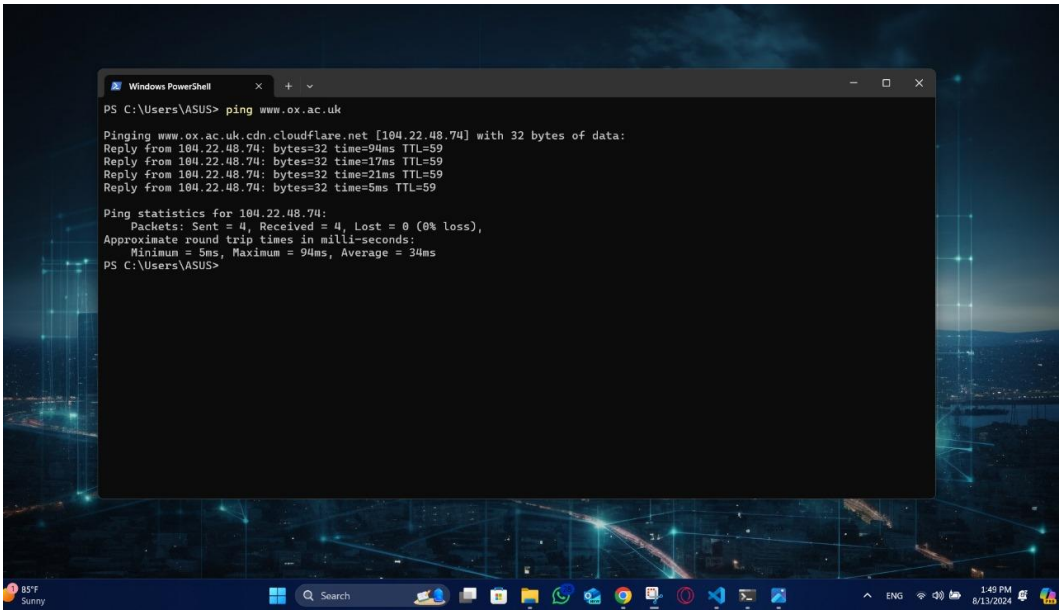

Figure 11: ping www.ox.ac.uk

The ping is used in Fig (11) to a domain name, the result shows the path passed by the packets, the IP address of the domain name and the times taken.

c) Using www.whois.com to find information about the IP address:

```
OrgName:        Internet Assigned Numbers Authority
OrgId:          IANA
Address:        12025 Waterfront Drive
Address:        Suite 300
City:           Los Angeles
StateProv:      CA
PostalCode:     90292
Country:        US
RegDate:
Updated:        2024-05-24
Ref:            https://rdap.arin.net/registry/entity/IANA
```

Figure 12: IP information for the local device.

```
OrgName:        Cloudflare, Inc.
OrgId:          CLOUD14
Address:        101 Townsend Street
City:           San Francisco
StateProv:      CA
PostalCode:     94107
Country:        US
RegDate:        2010-07-09
Updated:        2021-07-01
Ref:            https://rdap.arin.net/registry/entity/CLOUD14
```

Figure 13: IP information for www.ox.ac.uk

The IP address 192.168.1.9 is part of the private IP address range, typically used within a local area network (LAN).

The IP address 104.22.48.74 is a public IP address associated with the Oxford University website.

d)

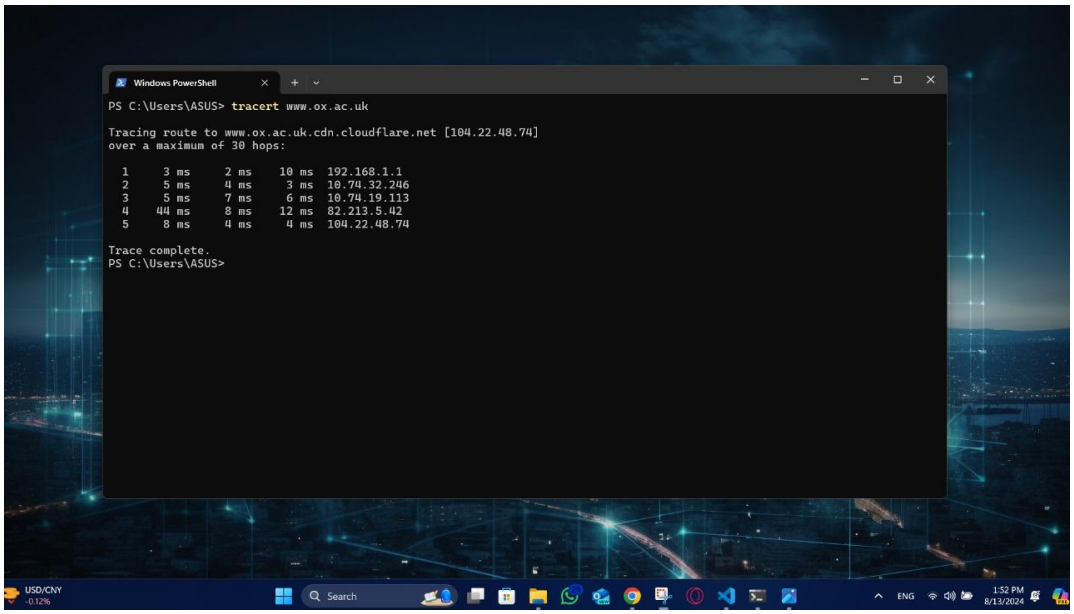

Figure 14: tracert www.ox.ac.uk

The results of the reacert in Fig (14) shows the path of the packets and the IP address for each router the packets passed to reach www.ox.ac.uk. It can be noticed that the final IP is 104.22.48.74, which is the IP for the website as it was shown in Fig (11).

e)

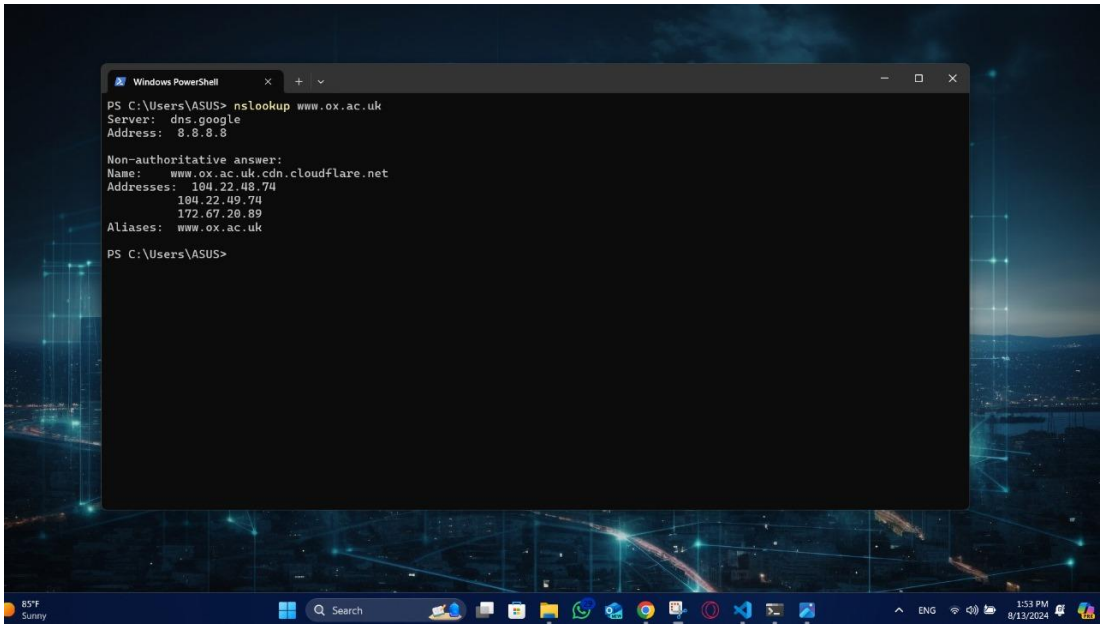

Figure 15: nslookup www.ox.ac.uk

The results of nslookup in Fig (15) shows the resolve of the domain name of www.ox.ac.uk and the IP address for the website, the server and the host. It also shows any aliases for the domain name.

f)

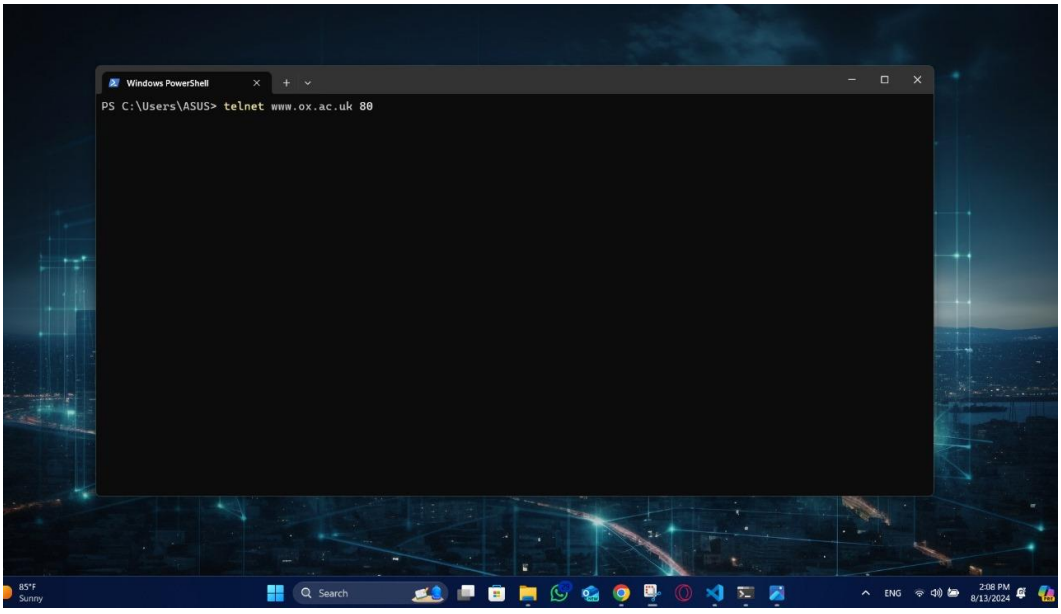

Figure 16: telnet www.ox.ac.uk

The connection for telnet in Fig (16) is running, the server is waiting for a message.

3) some details about autonomous system (AS) number, number of IPs, prefixes, peers, name of Tier1-ISP of www.ox.ac.uk.

Table 3: Information about www.ox.ac.uk

Announced prefix	129.67.0.0/16
Prefix Name	UNIV-OF-OXFORD-LINR
Prefix Description	Oxford University
ASN	As786
ASN Description	JANET
ASN Name	Jisc Services Limited
number of IPs	65,536

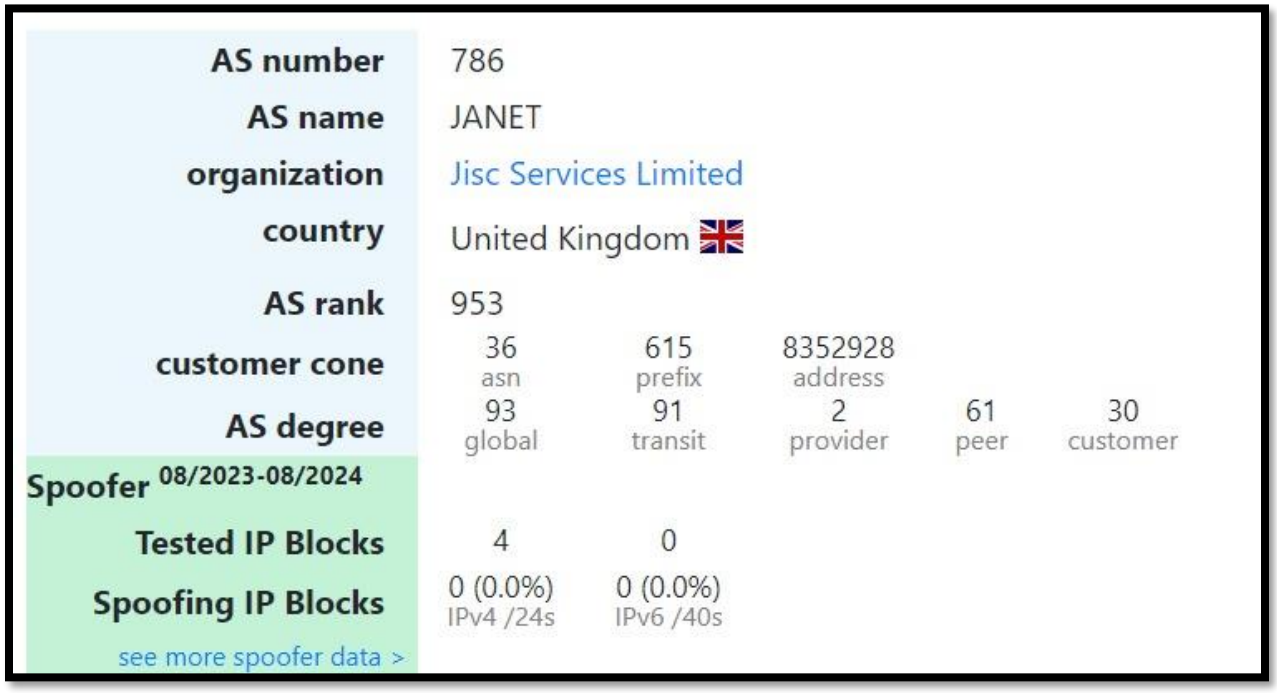

Figure 17: information of the domain www.ox.ac.uk

4) Using Wireshark to capture some DNS messages.

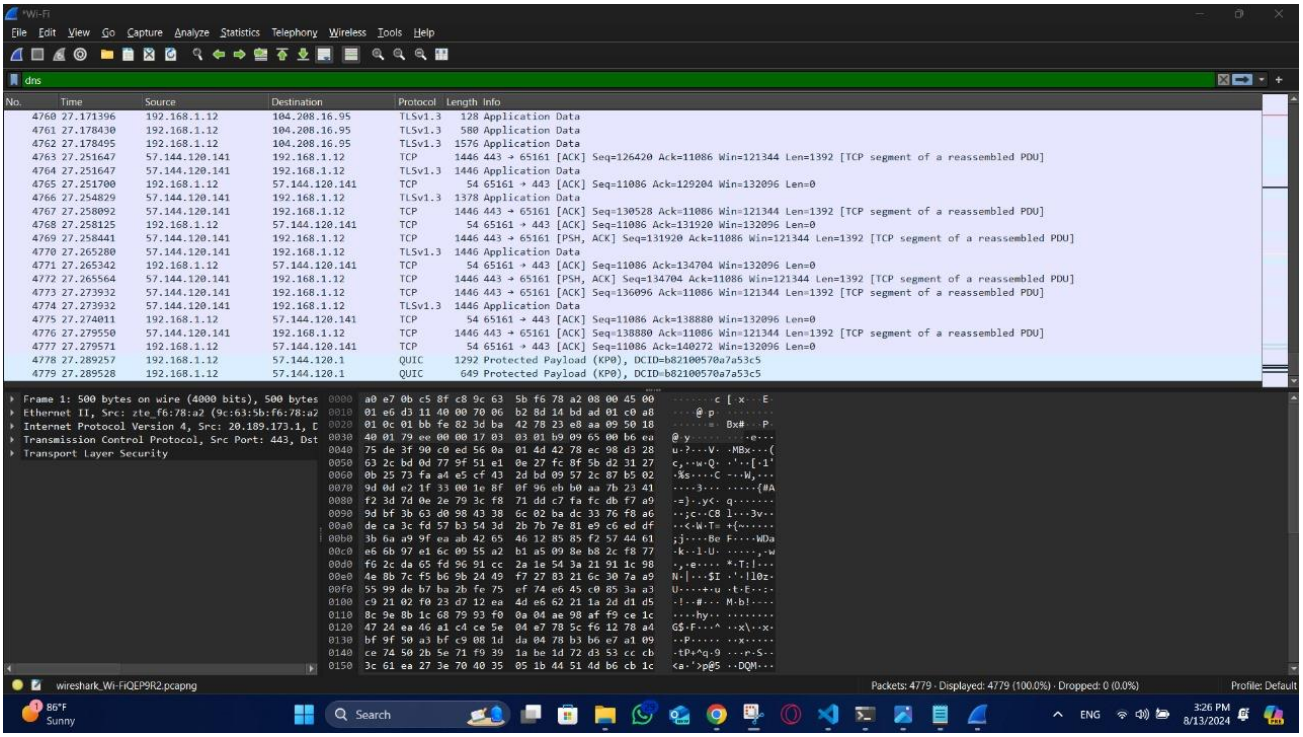

Figure 18: capture some DNS messages using Wireshark.

Fig (18) shows the output of the Wireshark software when capturing some DNS messages.

Socket Programming (TCP and UDP)

1) TCP client server

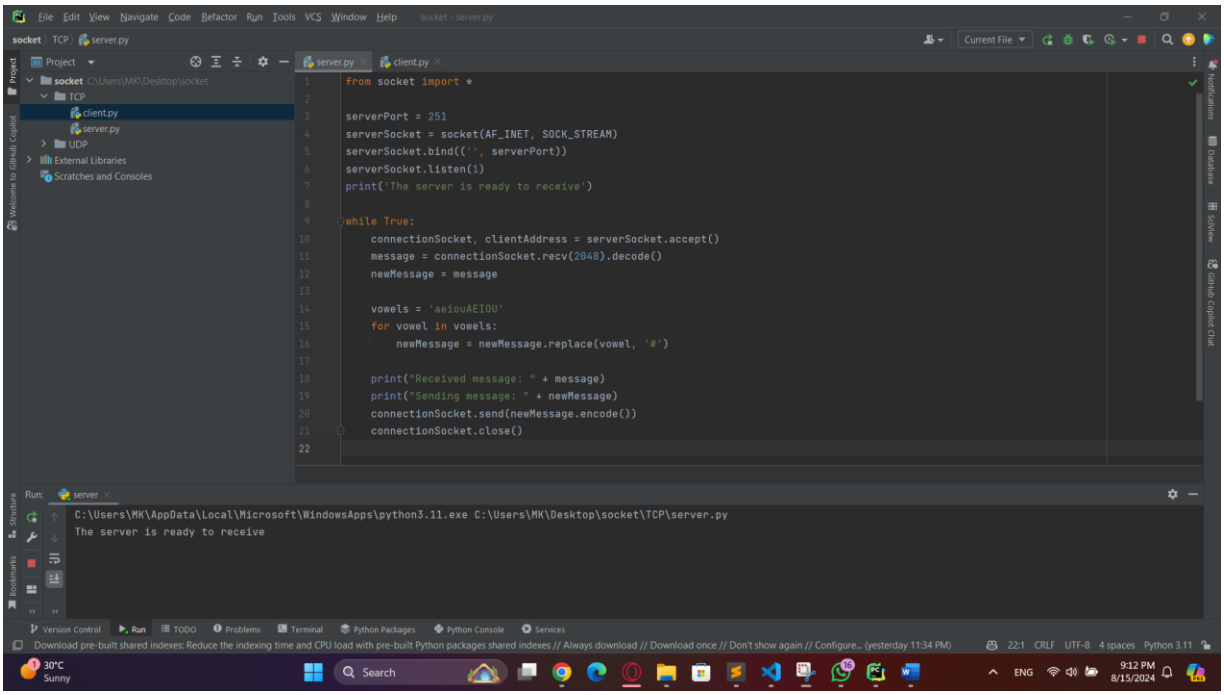

Figure 19: TCP Server.

The TCP server is built in Fig (19). The socket library is used, server port identified to 251 using Suhaib's ID 1220251, the server socket is defined TCP using (SOCK_STREAM), the socket is bound to the port, the server starts to listen, a message identifying that the server is ready is print. The Infinity loop is written to receive messages from clients, when the message arrives, the server socket will accept it and decode it, the new message is done by replacing the vowels by '#', the message and the new message are displayed, finally the message is encoded and sent back to the client and the connect close.

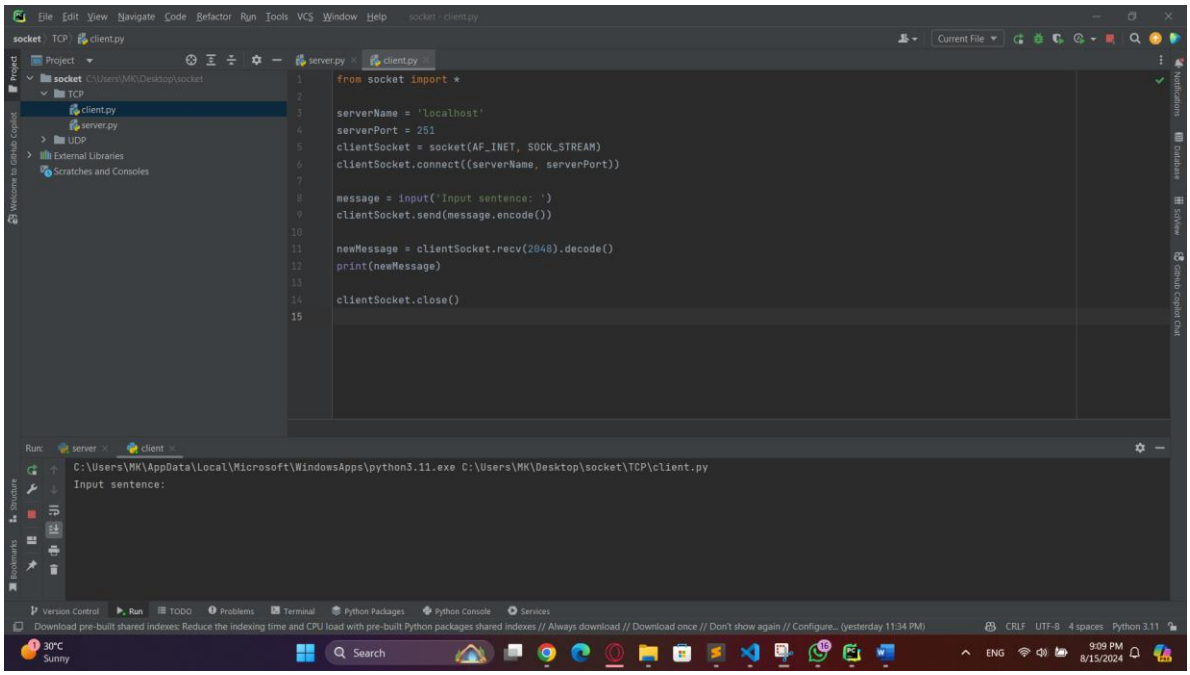

Figure 20: TCP Client.

The TCP server is built in Fig (20). The socket library is used, server port identified to 251 using Suhaib's ID 1220251 as the server in Fig (19) used, the client socket is defined TCP using (SOCK_STREAM), the connection is created. The message is taken from the input then sent to the server encoded, the new message comes from the server and then decoded, finally the client socket is closed.

Example:

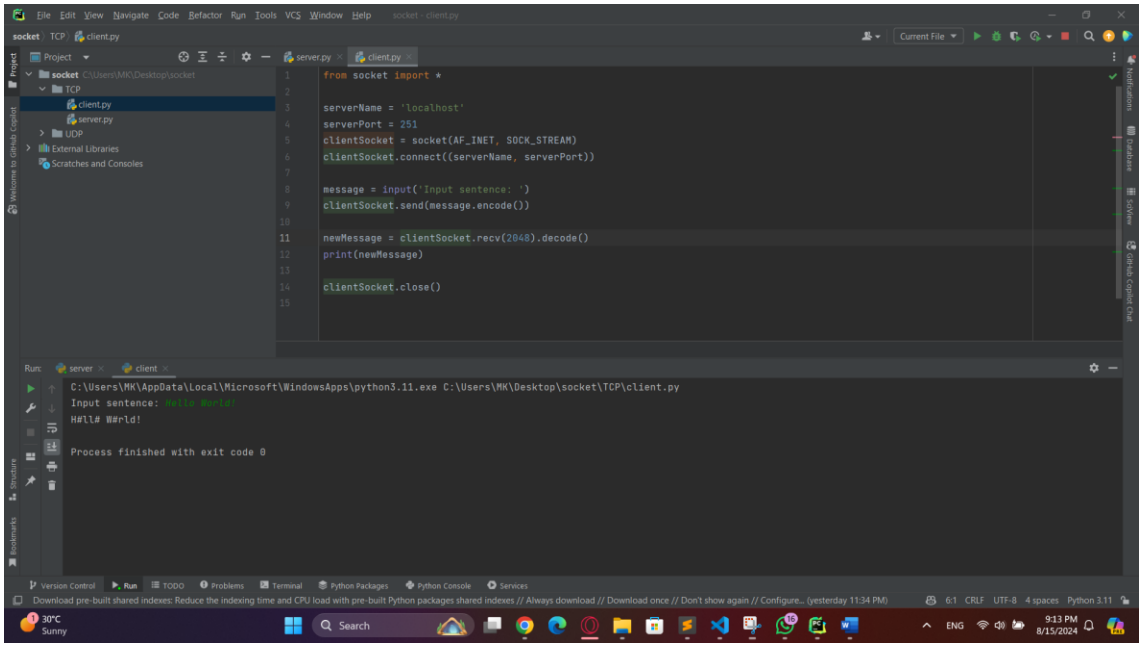

Figure 21: Client TCP connection example.

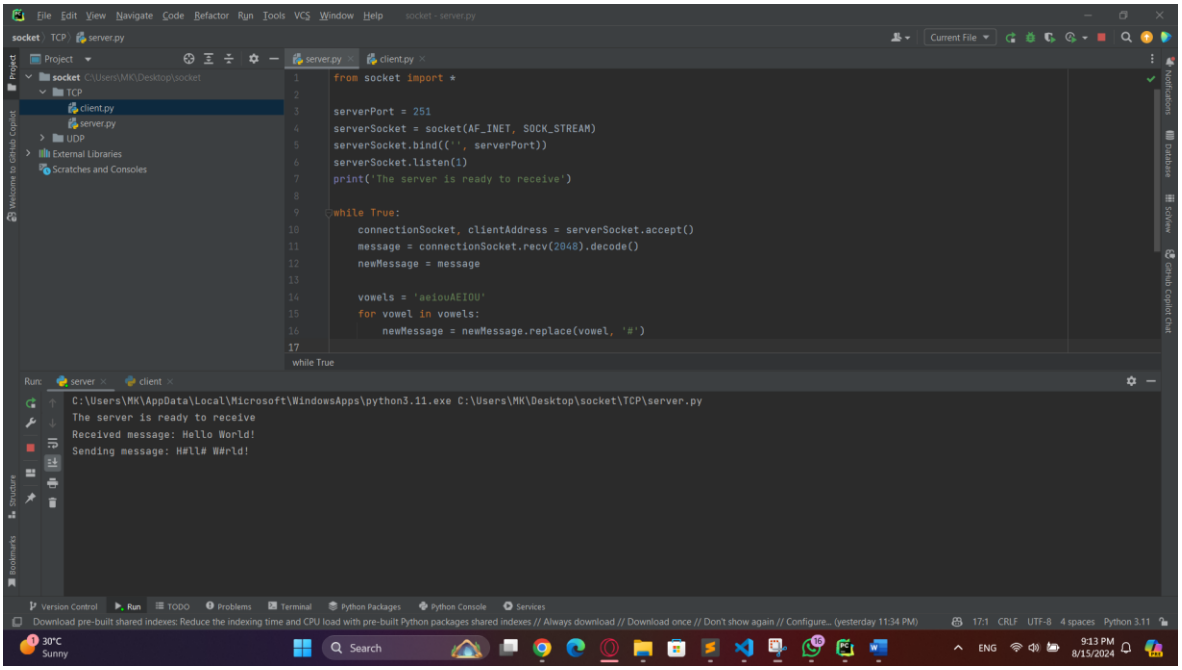

Figure 22: Server TCP connection example.

The example is done by sending the message “Hello World!” as in Fig (21), the server then receives the message and send the new message “H#ll# W#rld!” as in Fig (22), the client then receives the new message and display it as in Fig (21).

2) UDP clients and server

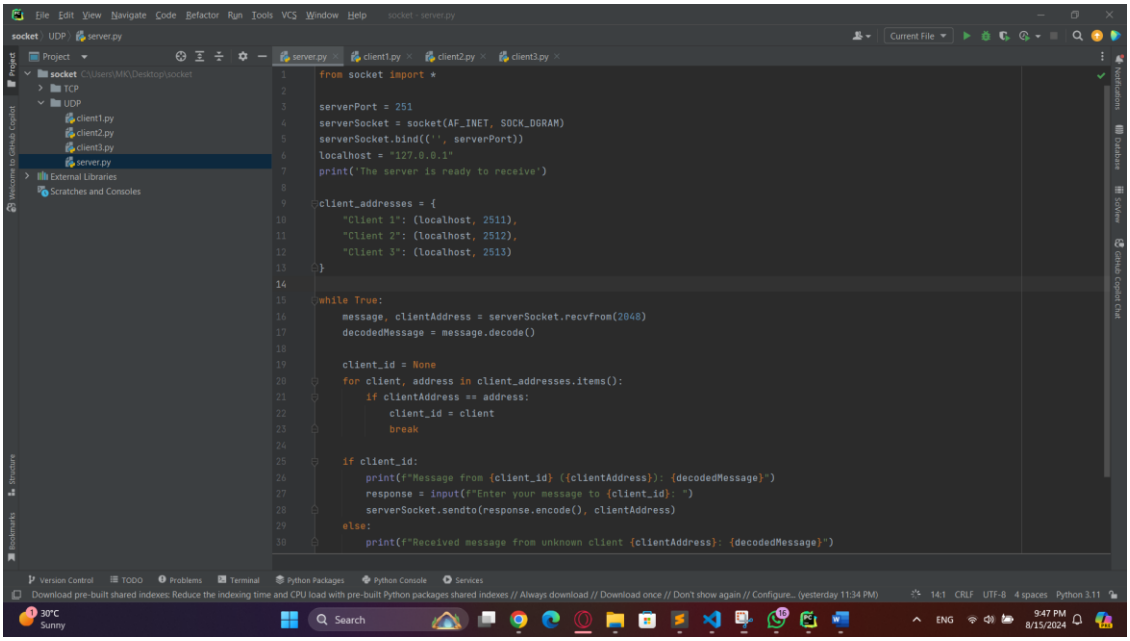

Figure 23: UDP server.

The UDP server is built in Fig (23). The socket library is used, server port identified to 251 using Suhaib's ID 1220251, the server socket is defined UDP using (SOCK_DGRAM), the socket is bound to the port. The port of each client is given, that is how the server would know which client sent the message. The Infinity for listening, when a message is arrived, the server would receive it and decode it, the server then will loop over the clients to find which one sent the message, if the server identified the client, the server will reply, else it will display that the client is unknow.

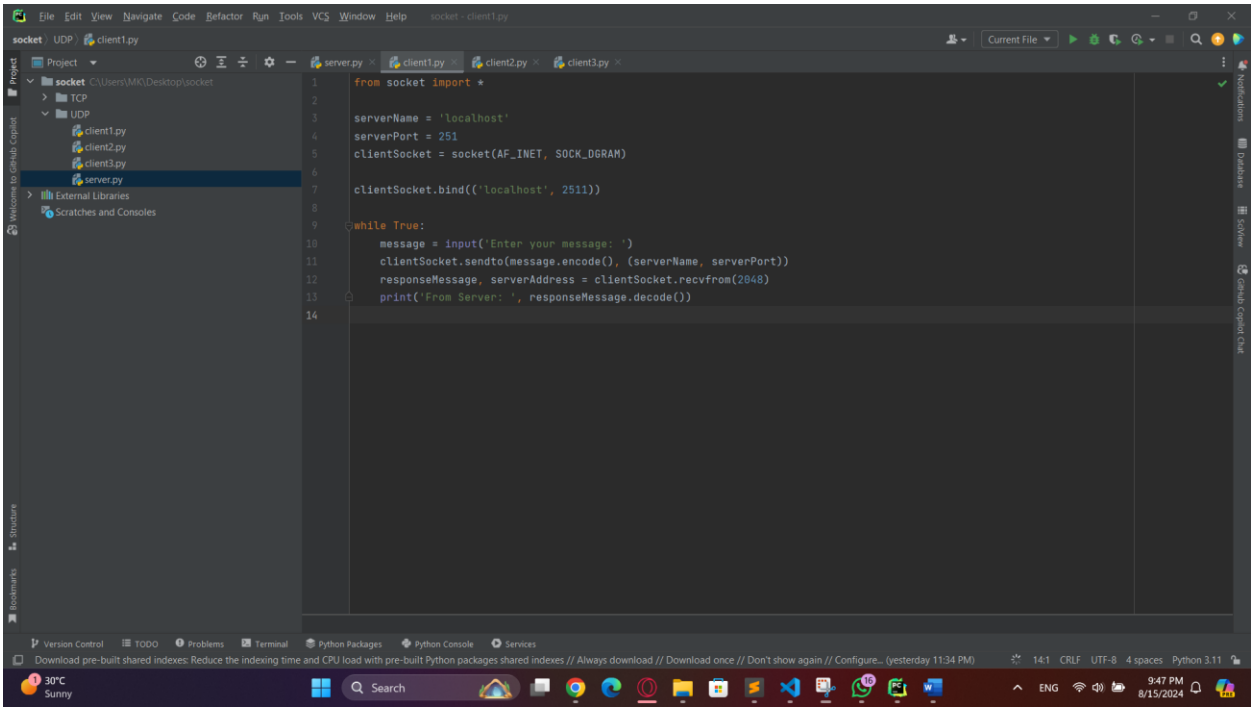

Figure 24: UDP Client 1.

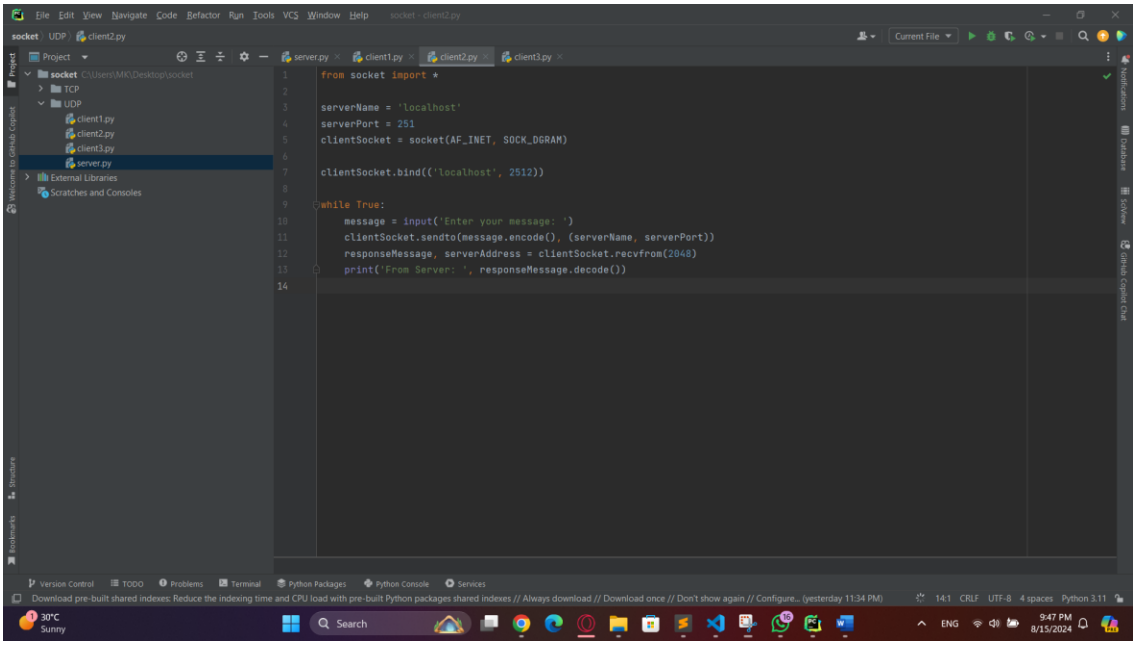

Figure 25: UDP Client 2.

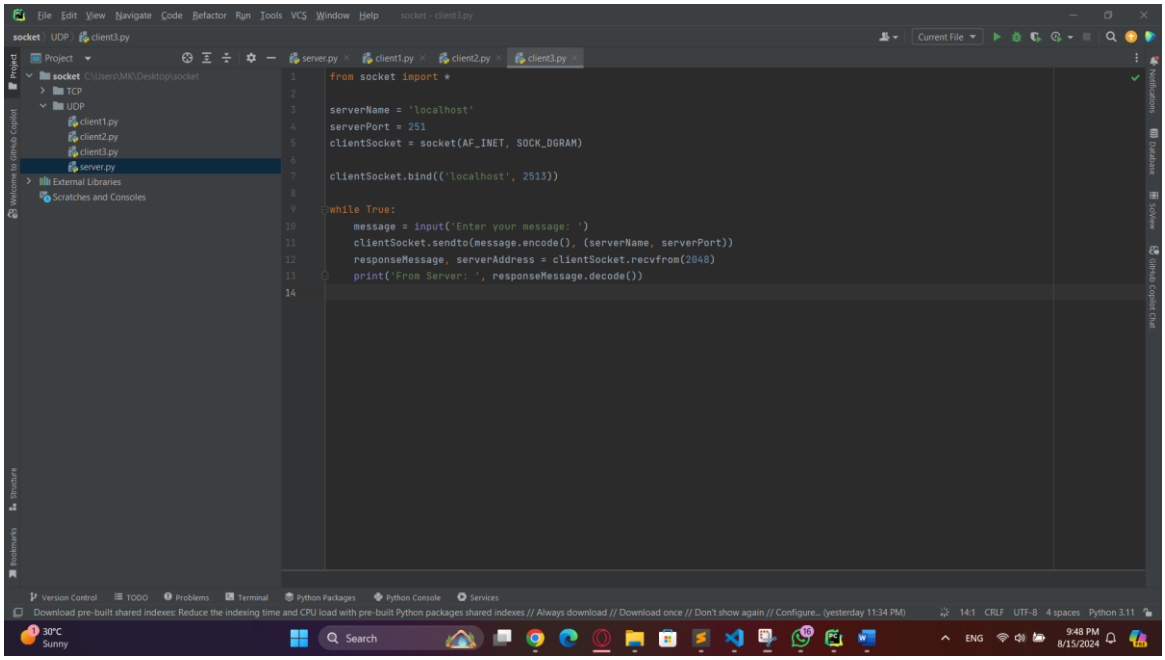

Figure 26: UDP Client 3.

Each one of the three UDP clients Fig (24, 25, 26), is implemented in the same way as the TCP clients, except two main differences: SOCK_DGRAM is used to identify that the socket is UDP, each client socket is bound to a port “251” + “client number” to give each client unique port so that the server can identify them.

Example:

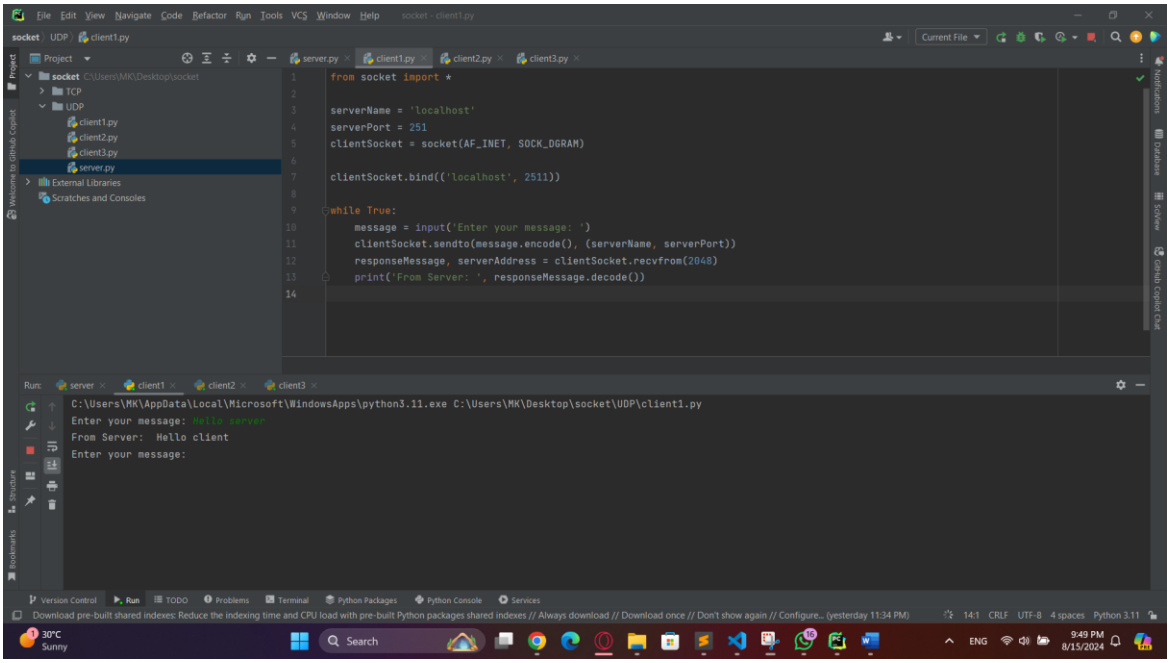

Figure 27: UDP Client 1 example.

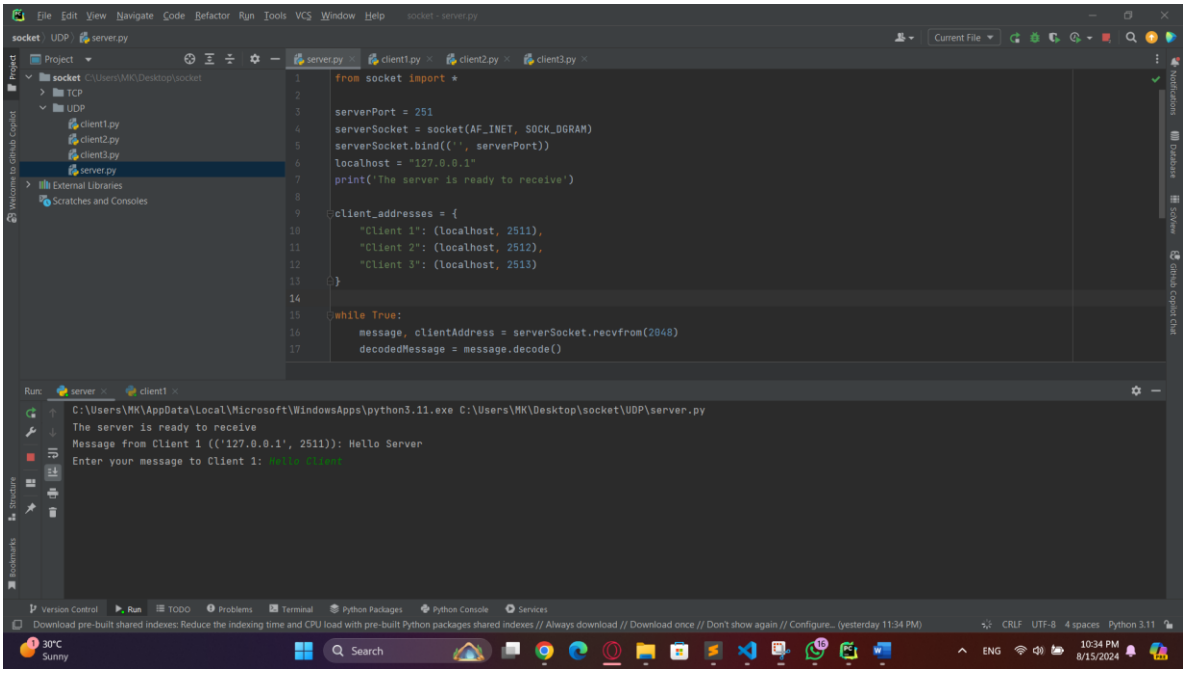

Figure 28: UDP Server for Client 1 example.

The example in Fig (27, 28) for client 1 sending message to the server, the server receives the message, identify that client 1 sent this message and then reply to it.

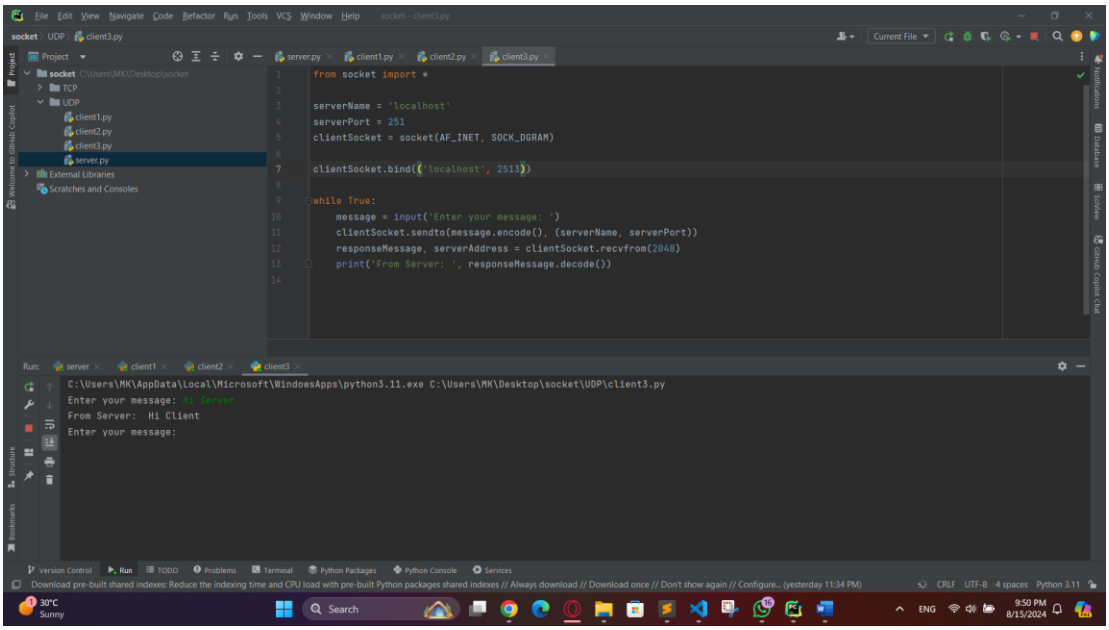

Figure 29: UDP Client 3 example.

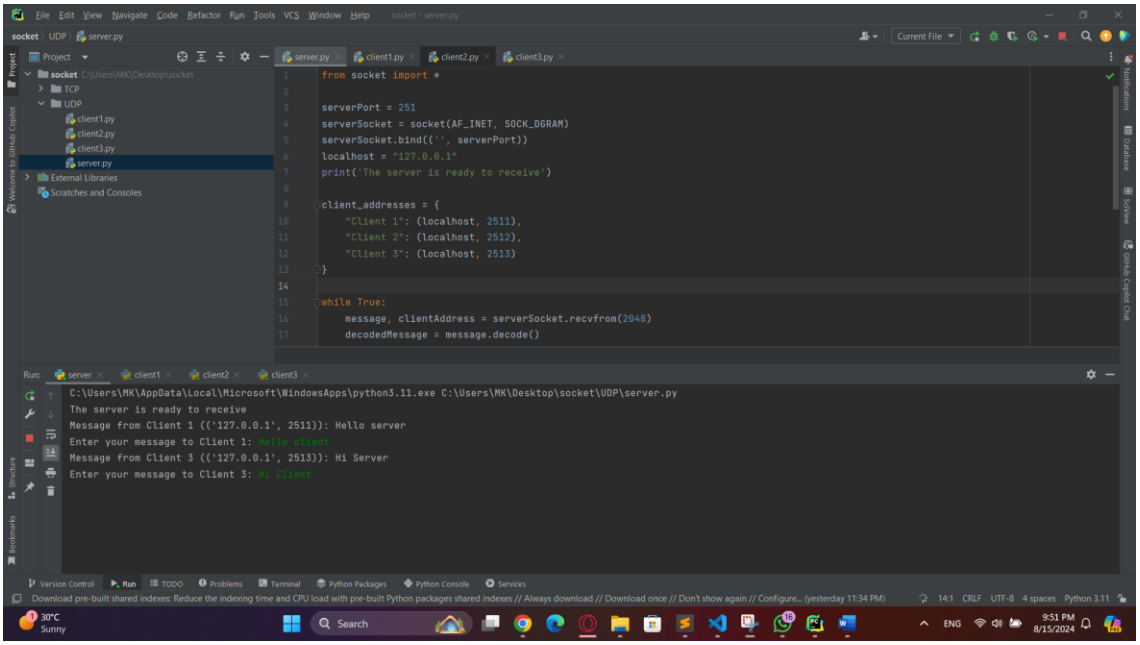

Figure 30: UDP Server for Client 3 example.

The example in Fig (29, 30) for client 3 sending message to the server, the server receives the message, identify that client 2 sent this message and then reply to it. Even though client 3 sent a message before client 2, the server identified it.

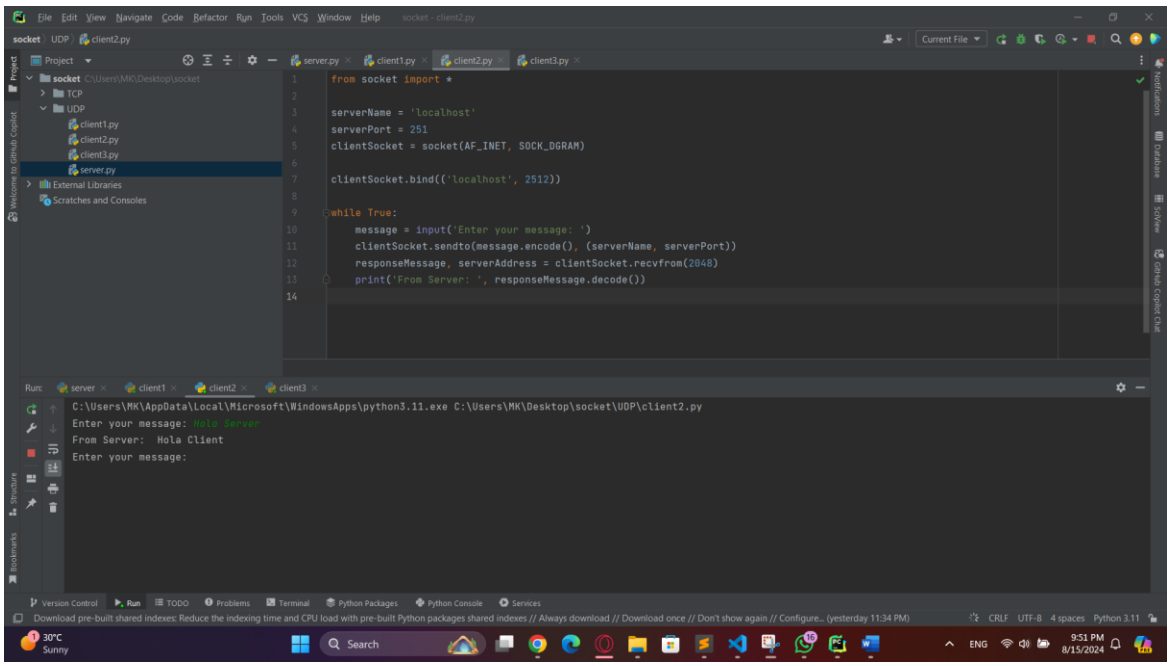

Figure 31: UDP Client 2 example.

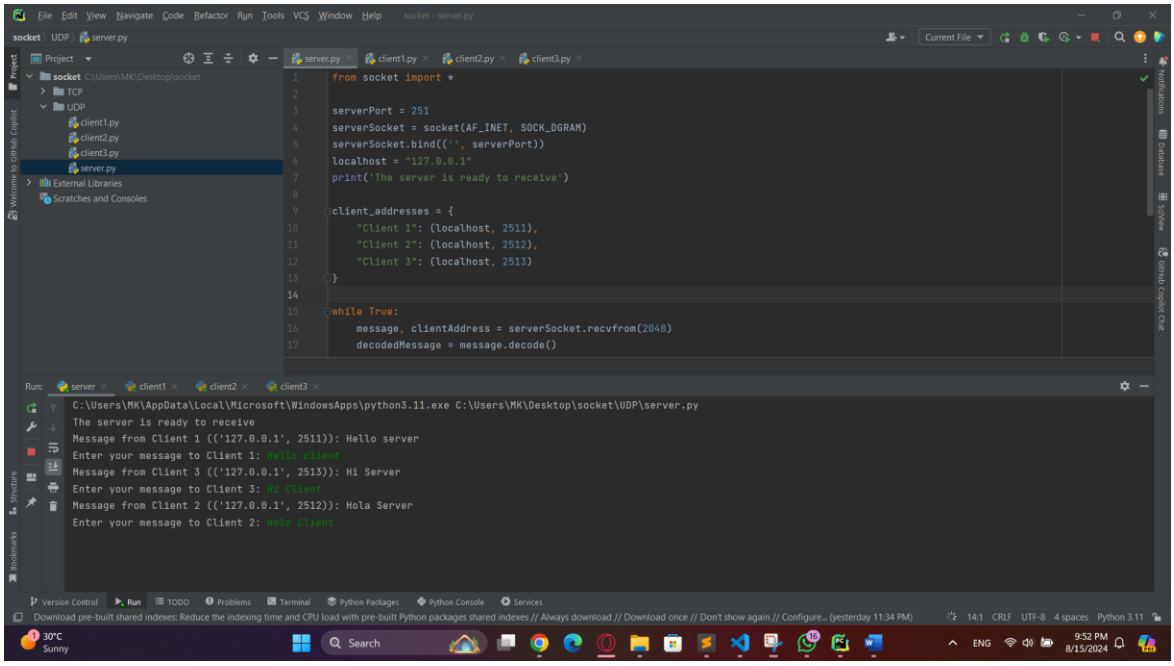

Figure 32: UDP Server for Client 2 example.

The example in Fig (31, 32) for client 2 sending message to the server, the server receives the message, identify that client 2 sent this message and then reply to it.

Web Server

The main page is:

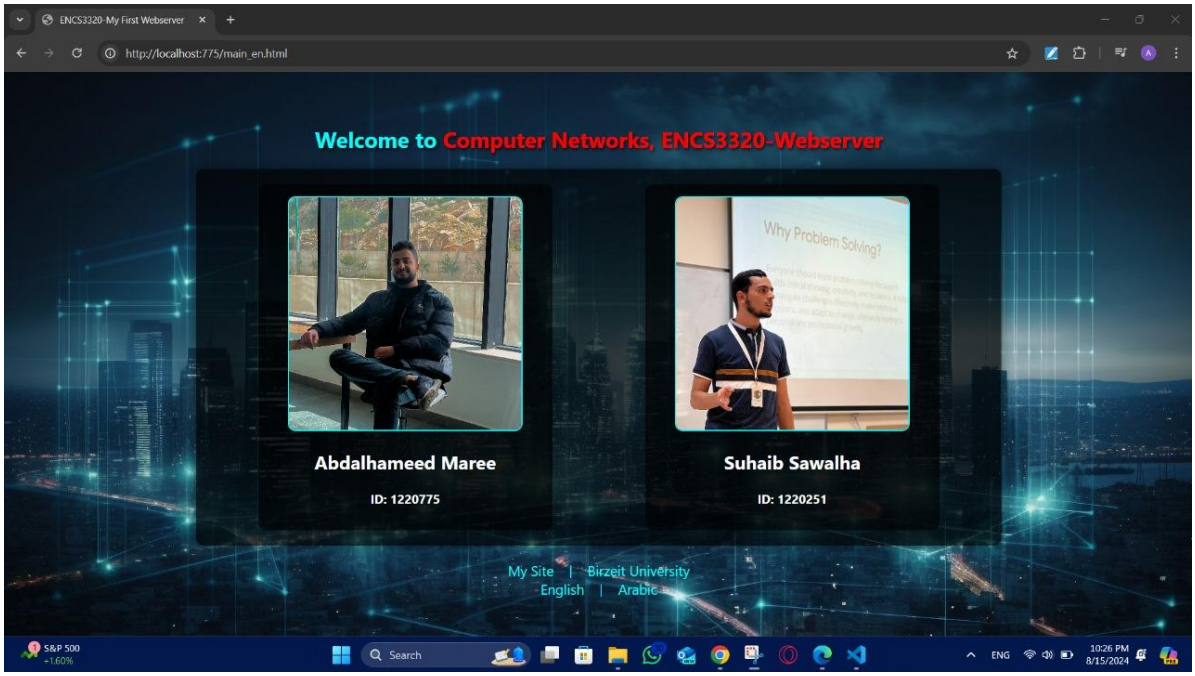

Figure 33: Main HTML page from localhost.

The page in Fig (33) is opened from the localhost and the main laptop that was implemented on. By IP address, the website can be opened from any device in the network.

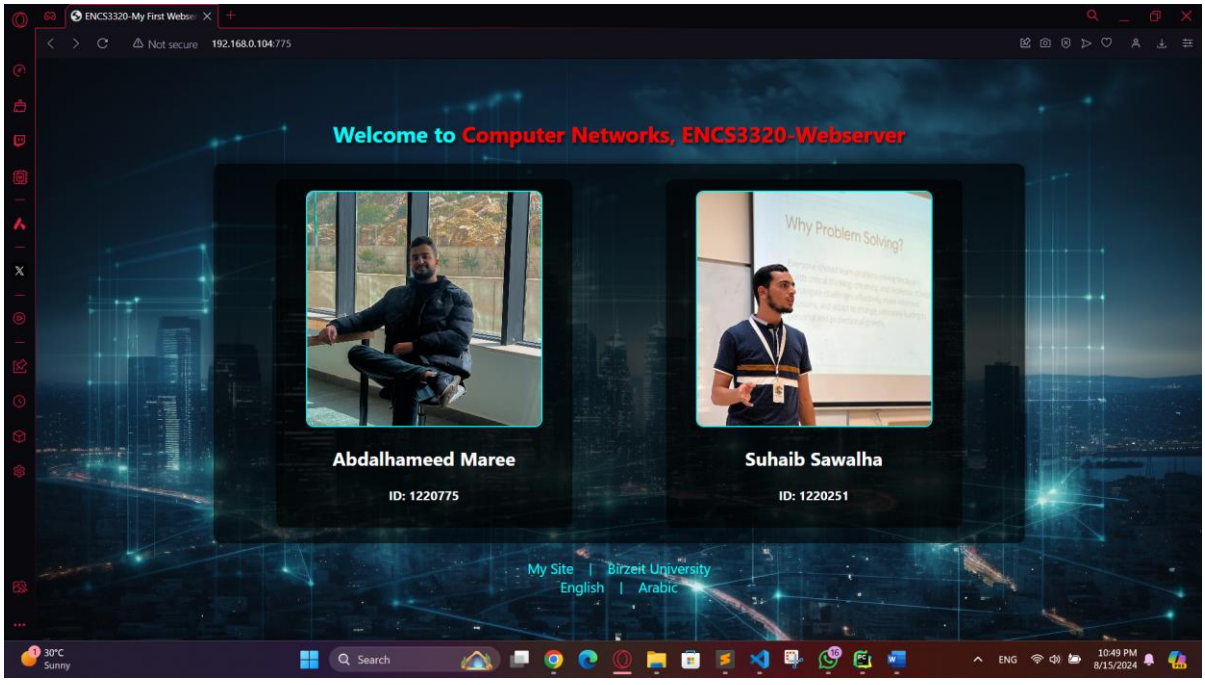

Figure 34: Main HTML page from another laptop in the network.

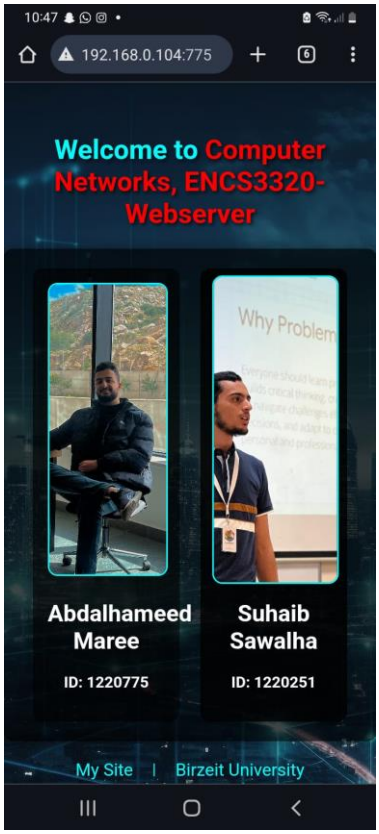

Figure 35: Main HTML page opened from mobile in the same network.

1) The request ‘/’, ‘/index.html’, ‘/main_en.html’ or ‘/en’ would return the main html page.

Figure 36: the main page '/'

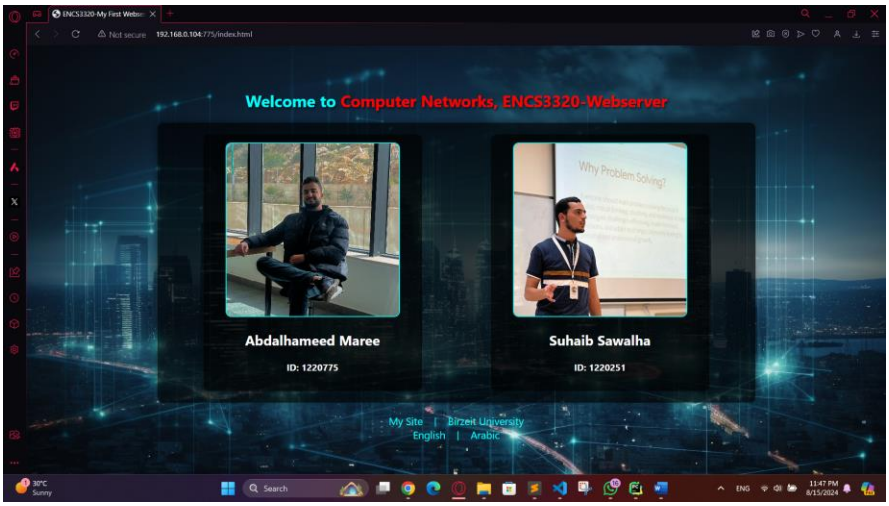

Figure 37: the main page '/index.html'

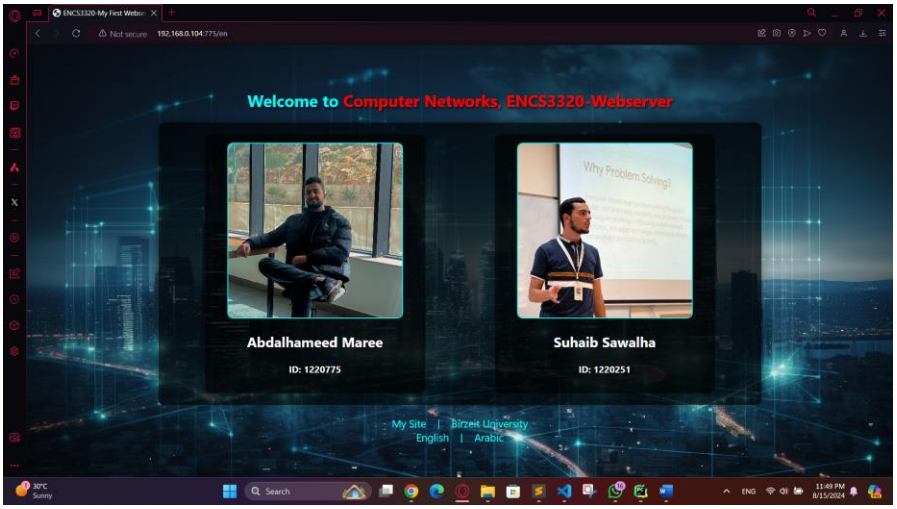

Figure 38: the main page '/en'

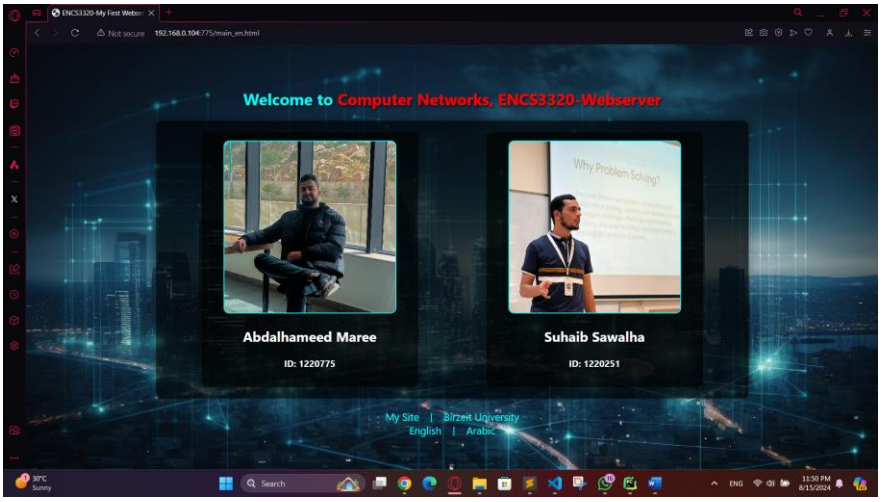

Figure 39: the main page 'main_en.html'

All requests in Fig (36, 37, 38, 39) redirects to the same (main) page.

When press on the image of one of the partners, it redirects the user to a page contains the CV of the person who was clicked one.

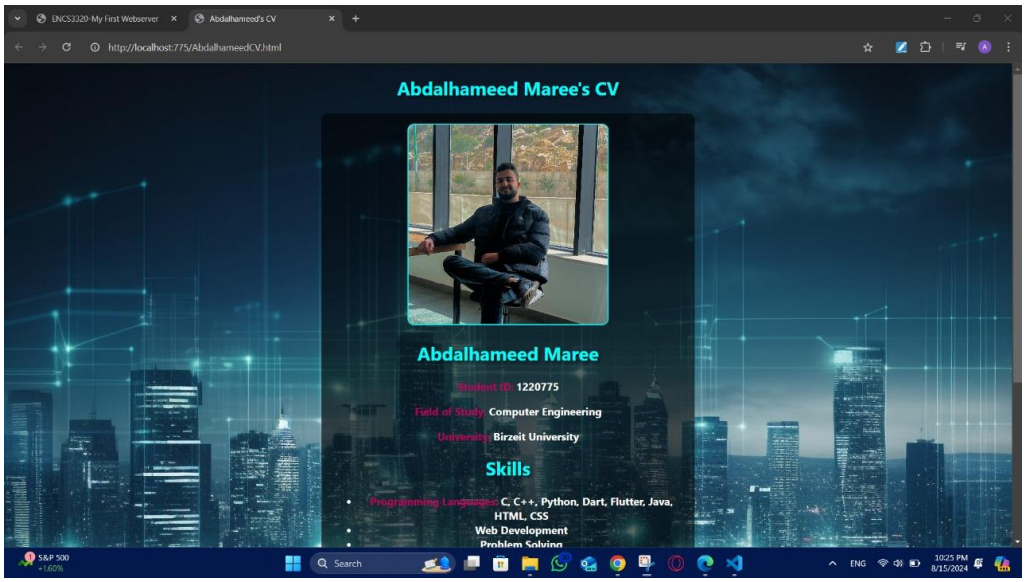

Figure 40: Abd Alhameed CV in English.

2) If the request is /ar then the server response with main_ar.html which is an Arabic version of main_en.html.

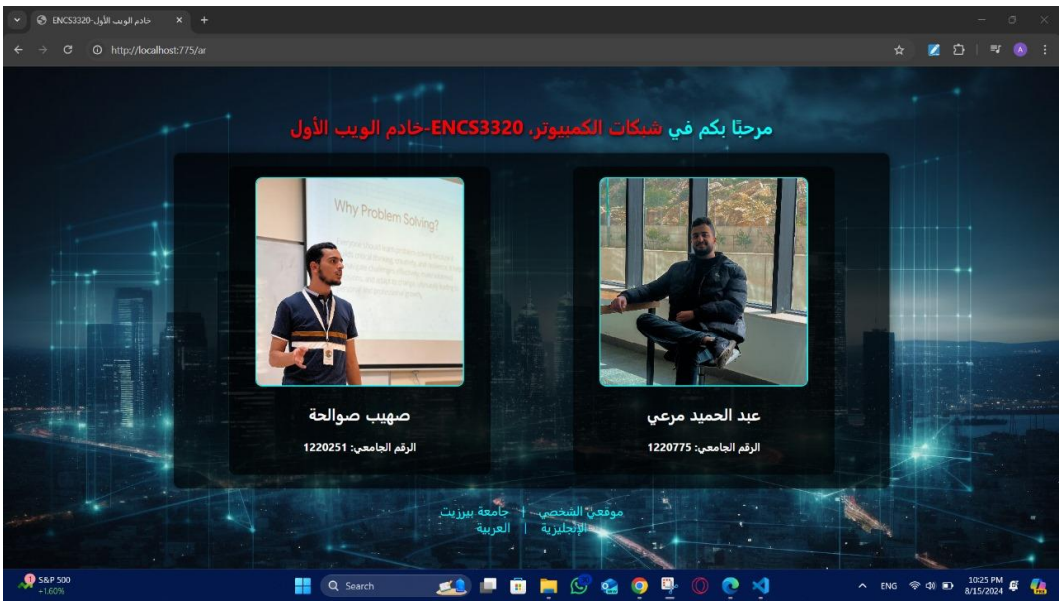

Figure 41: the main page in Arabic '/ar'

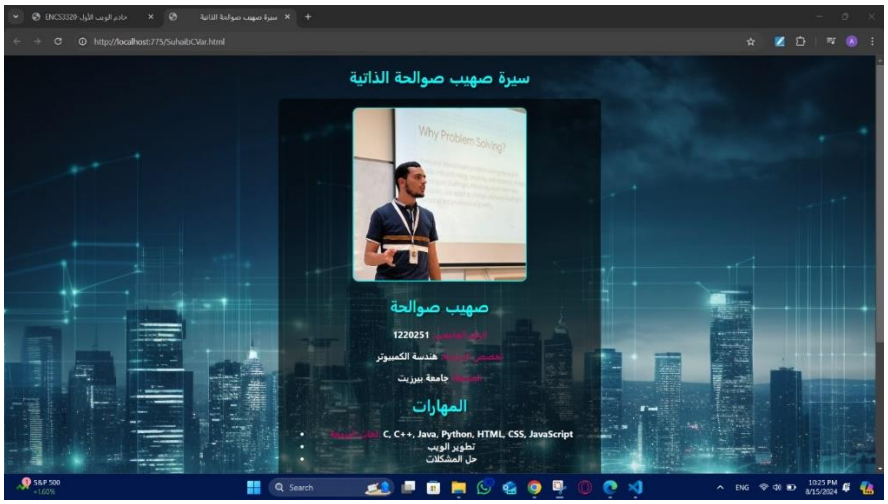

Figure 42: Suhaib CV in Arabic.

3) Every .html file is sent with 'Content-Type: text/html' and displayed in the browser.

   For example, the CV pages are sent with 'Content-Type: text/html', so the page is displayed in the browser as translation of the HTML code.

4) Every .css file is sent with 'Content-Type: text/css' and displayed in the browser.

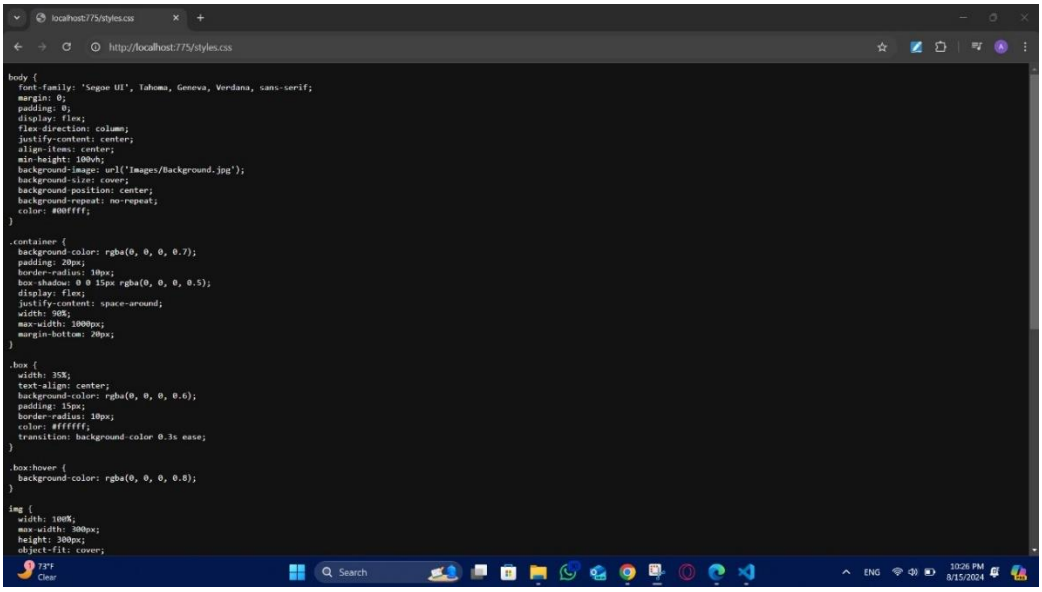

Figure 43: content-type: text/css request.

A request for style.css is sent, the request is sent with 'Content-Type: text/css' and the content of the file is displayed.

5) Every .png file is sent with ‘Content-Type: text/png’ and displayed in the browser.

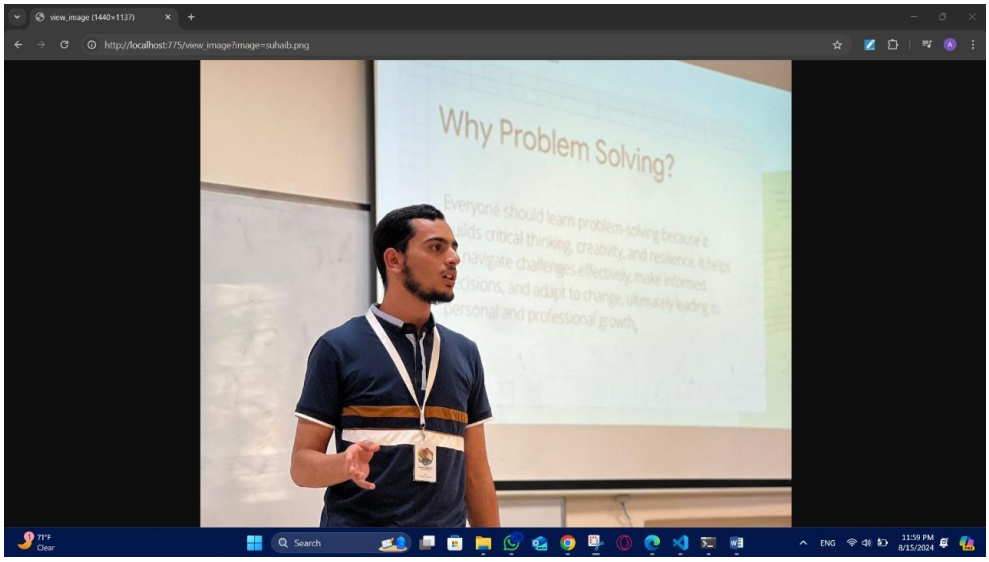

Figure 44: content-type: text/png request.

A request for suhaib.png is sent, the request is sent with ‘Content-Type: text/png and the image is displayed.

6) Every .jpg file is sent with ‘Content-Type: text/jpeg’ and displayed in the browser.

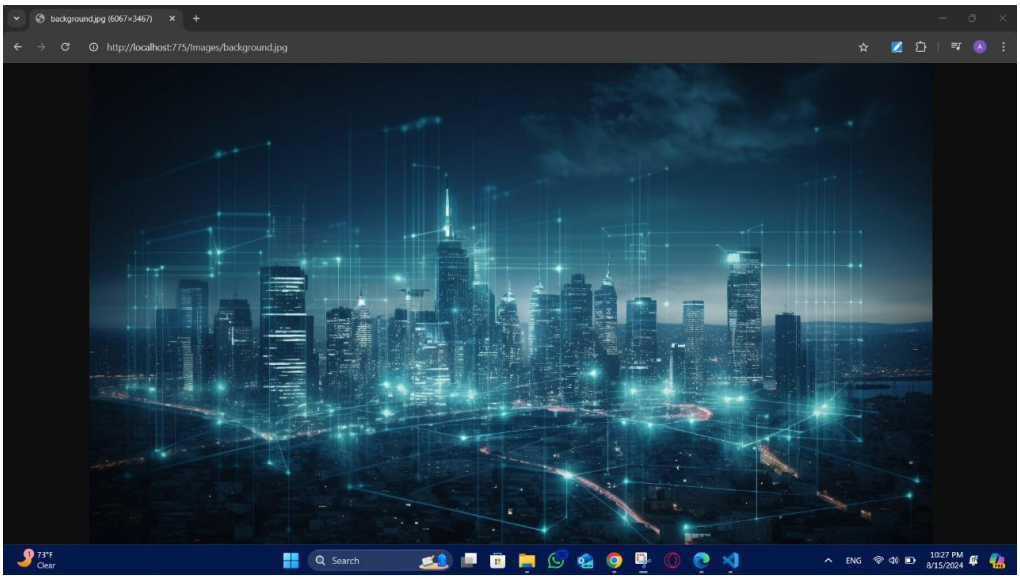

Figure 45: content-type: text/jpg request.

A request for background.jpg is sent, the request is sent with ‘Content-Type: text/jpg and the image is displayed.

7) All images are stored in a folder named “Images”.

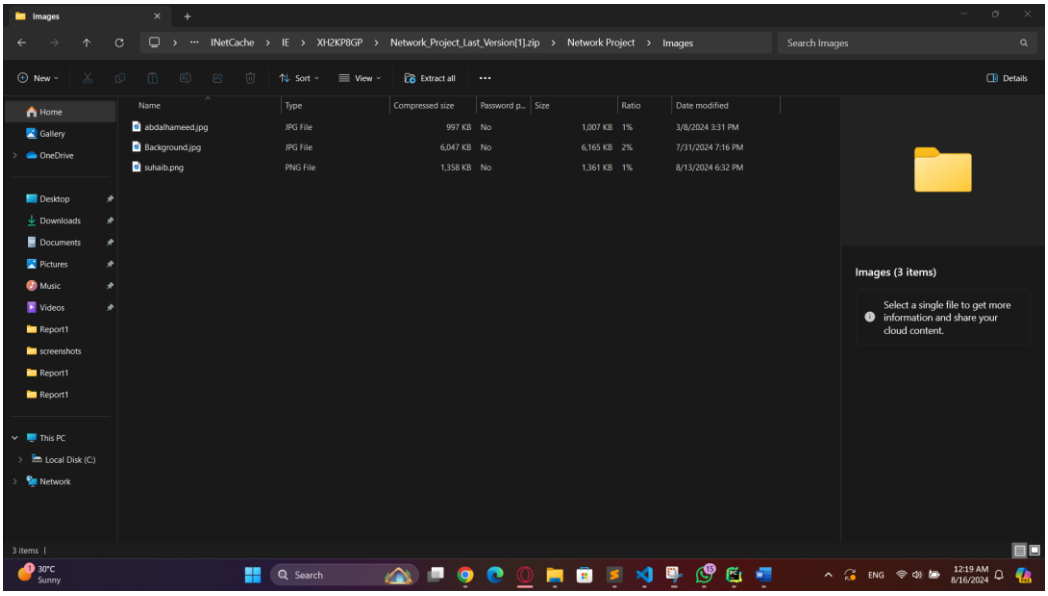

Figure 46: Images folder.

8) Get Images by their name.

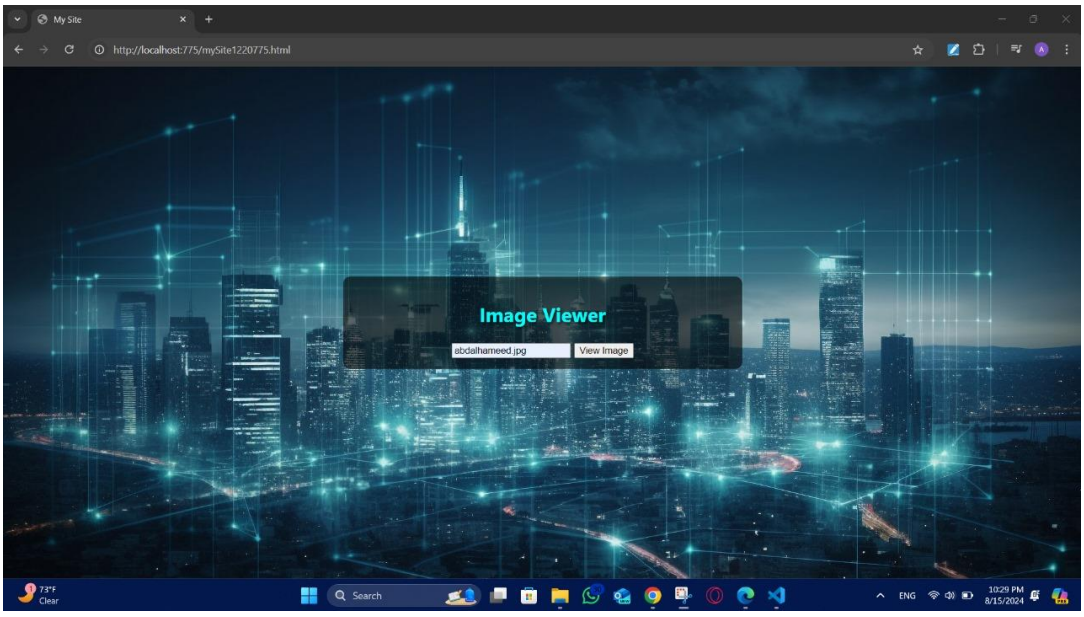

Figure 47: mySite1220775.html

When press ‘View Image’ button, a request for abdalhameed.jpg is sent, the image exists so the request will succeed and the page with ‘Content-Type: text/jpg’ is sent.

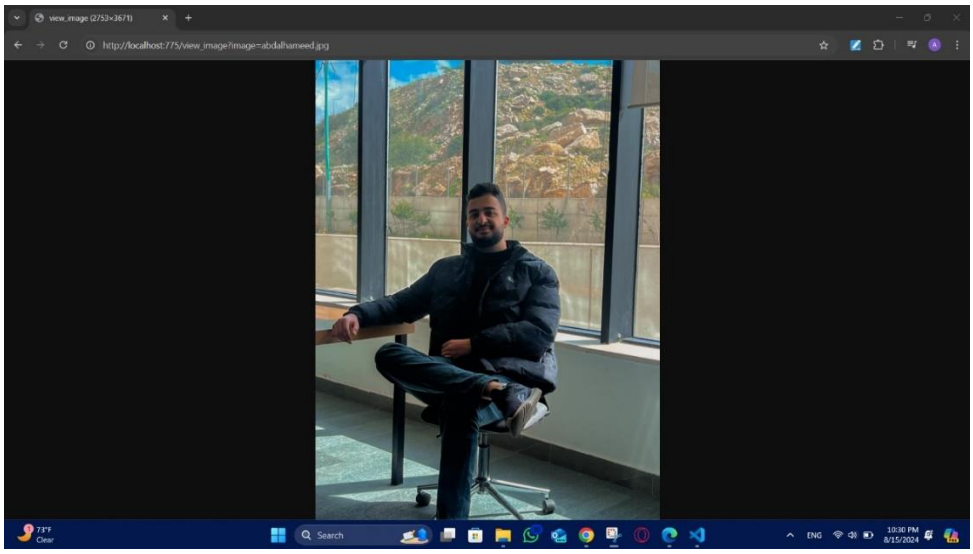

Figure 48: result of search abdalhameed.jpg in mySite775.html

9) A status code 307 Temporary Redirect is given when redirect to the following:

   a) ‘/so’ would redirect to ‘stackoverflow.com’

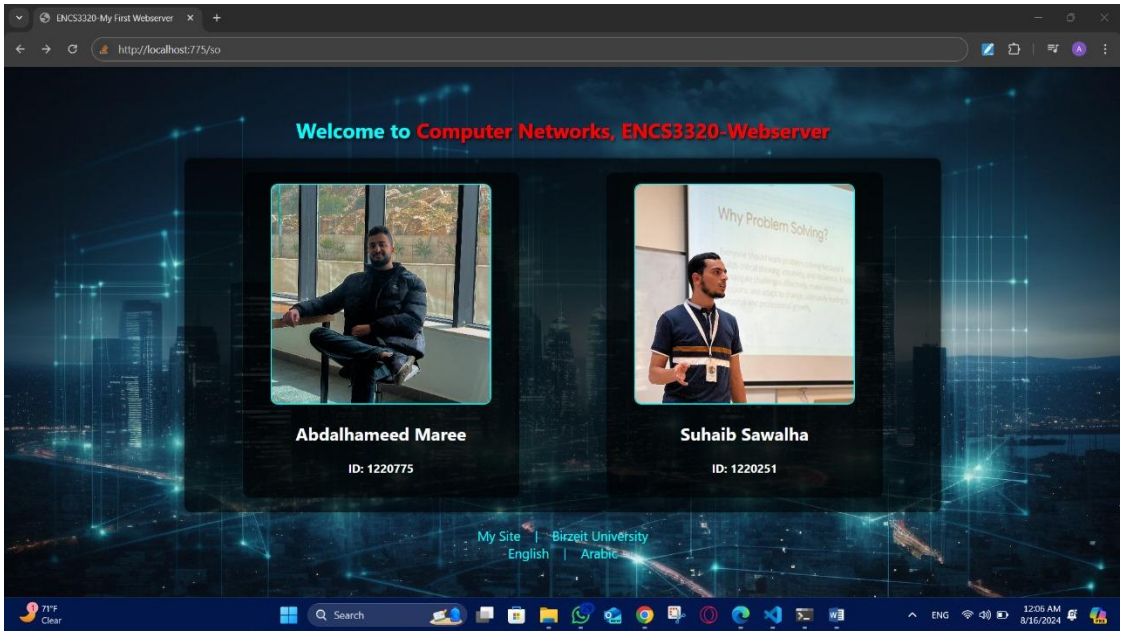

Figure 49: redirect to stackoverflow.com '/so'

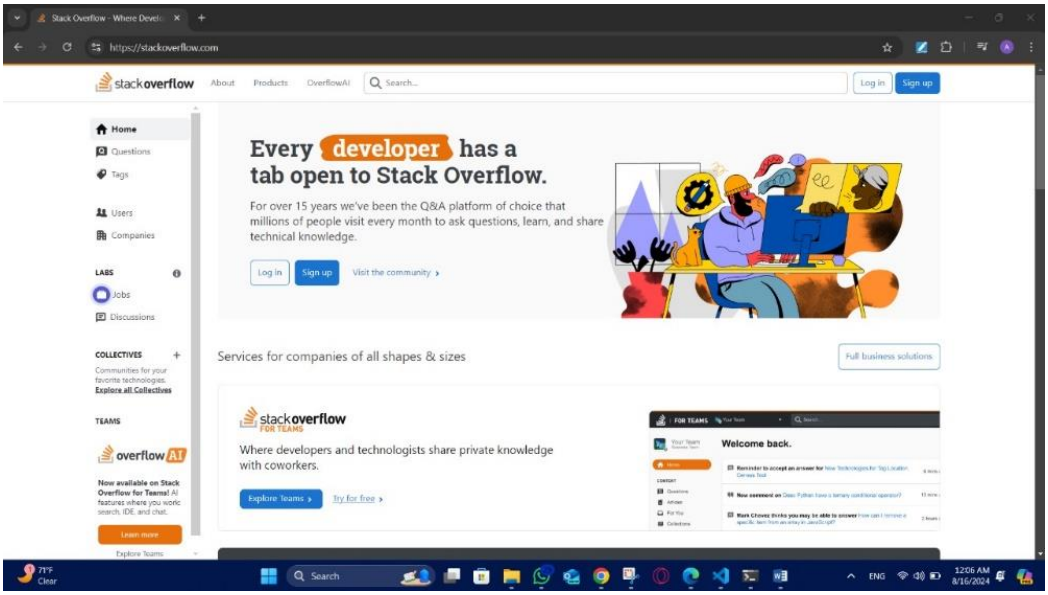

Figure 50: stackoverflow.com

b) '/itc' would redirect to 'itc.birzeit.edu'

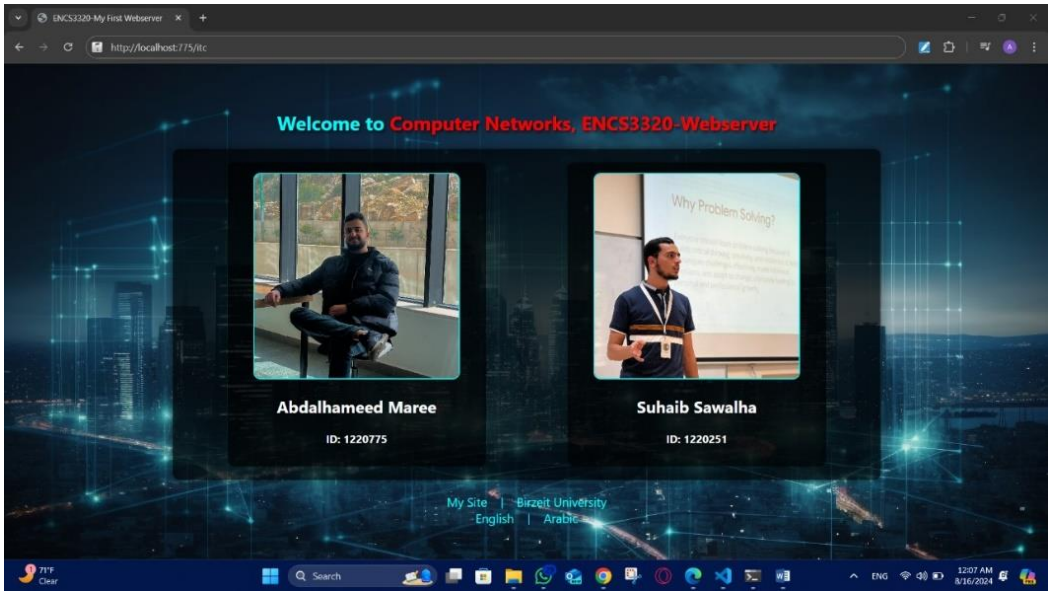

Figure 51: redirect to itc.birzeit.edu '/itc'

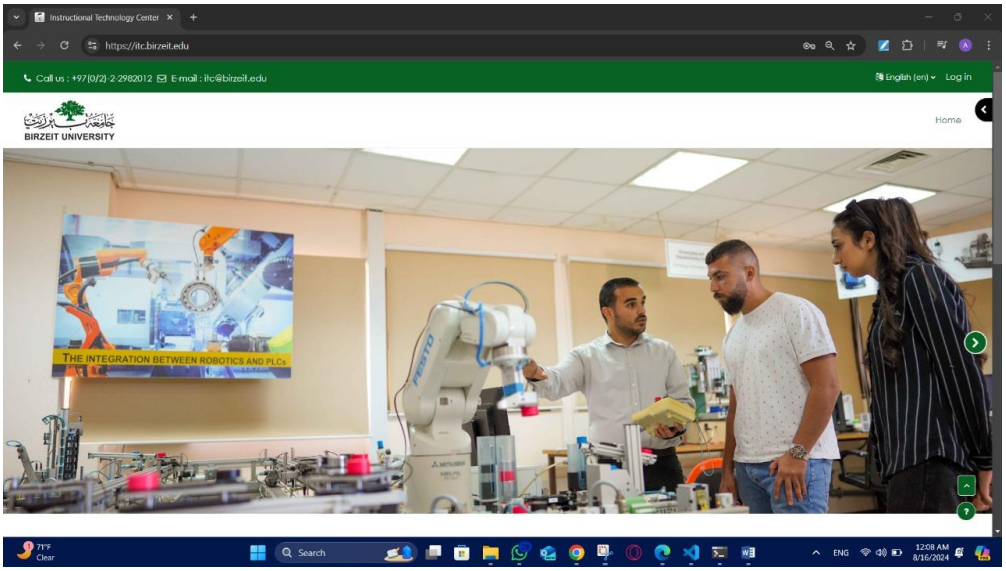

Figure 52: itc.birzeit.edu

10) Error Page

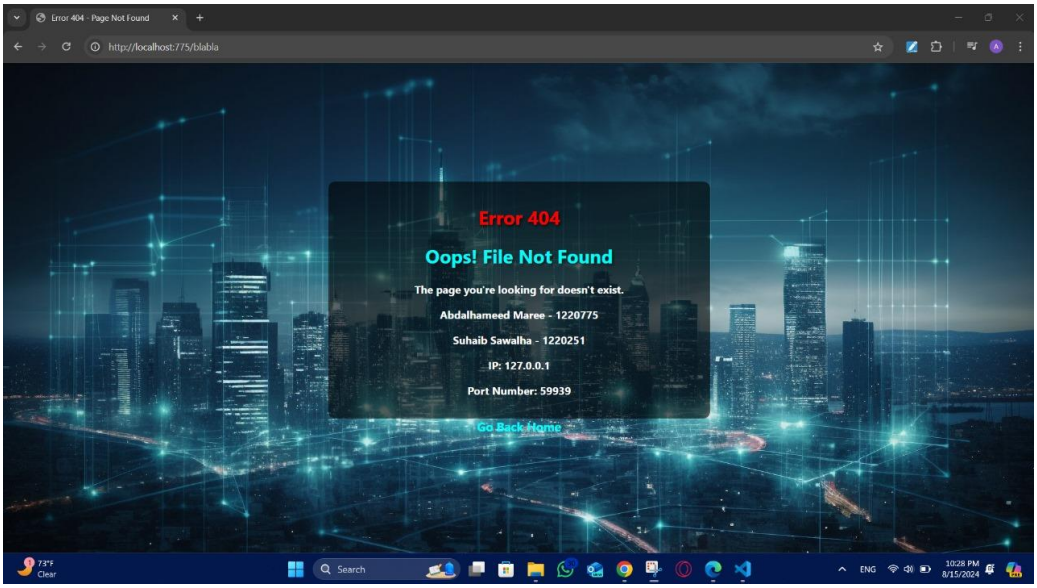

Figure 53: Error Page.

Error page appears when requesting a bad request or a file that does not exist. The IP and the port number are displayed on the screen.

11) HTTP requests

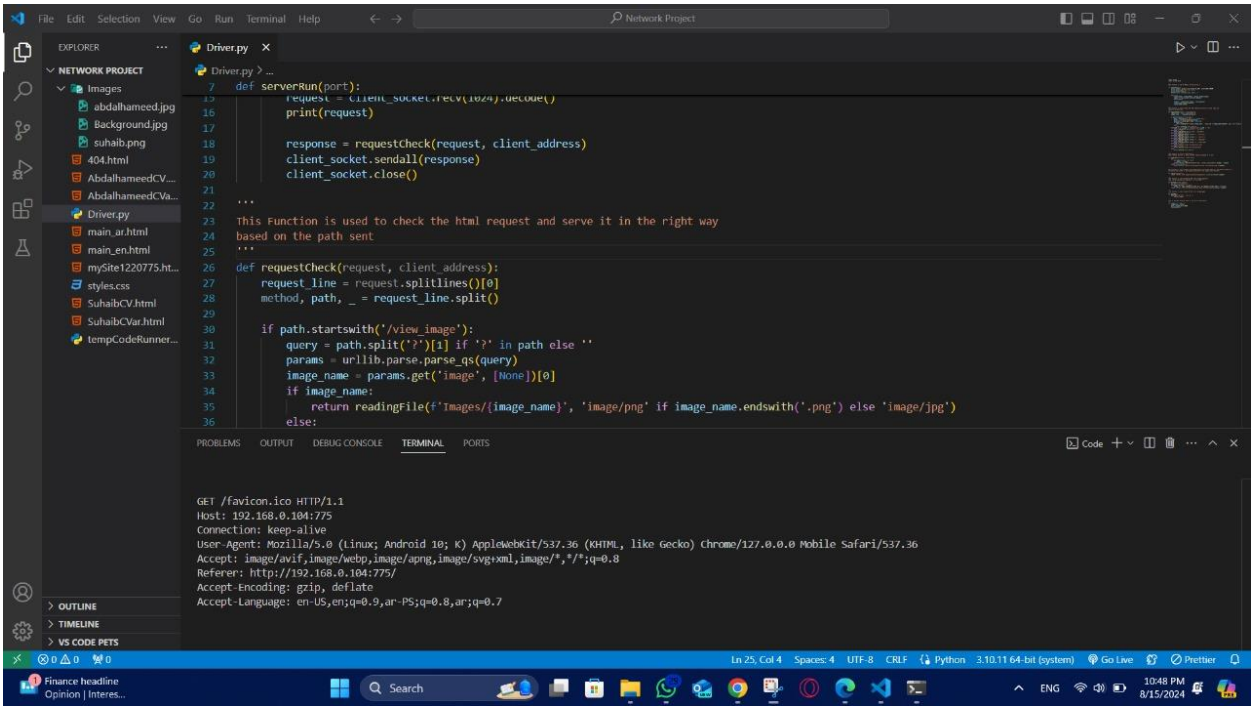

Figure 54: HTTP requests.

HTTP requests that are sent for each move are displayed on the terminal.

Alternatives Solutions, Issues and Limitations

Commands & Wireshark

At first, telnet command was not available in the machine, after install it and run “telnet www.ox.ac.uk” an error occurred, the solution was to specify the port to 80 instead of the default, which is 25.

There is an issue with www.bgpview.io to find the peers and 1st ISP, the site is not opening, and after hours of searching for an alternative, could not find any.

Socket Programming (TCP and UDP)

The peering part was challenging in this part, identifying the client who sent a message to the server was a hard thing to figure. At first, the code was written in a way that the client sends the message to the server telling what its number, the problem here is that client 1 can mock the server by telling it that its client 2, so an alternative way must be done.

The second approach was to store the client and give it a number and store its address in the server, this solution solved the mocking problem, but caused a problem that the order of clients would be missed, if client 2 sent a message before client 1, then the server will swap their numbers.

The best solution was to manually give each client a port and tell the server (manually) which port belongs to which client, this approach solved the problem of mocking and the problem of ordering. This approach also has it downs, which is that ports need to be handled manually, but as the requirements of the project tells that there are three clients, this solution was the best.

Web Server

The limitation of not using any library except socket library was challenging especially with searching for images part, the displayed image is found by the URL, parsing the URL is easy using a library called “urllib”, but as a result of this limitation, the function was implemented manually, which was challenging.

Redirecting to another web page automatically was also challenging, it required a lot of searches to find a solution.

Teamwork

The project was done by two students: Suhaib Sawalha and Abd Alhameed Maree. Each part of the project was distributed between them and each one of them helped the other in his part, but mainly the distribution was as follows:

- Command & Wireshark: Abd Alhameed Maree.
- Socket Programming (TCP and UDP): Suhaib Sawalha.
- Web Server: Abd Alhameed Maree.
- Report: Suhaib Sawalha.

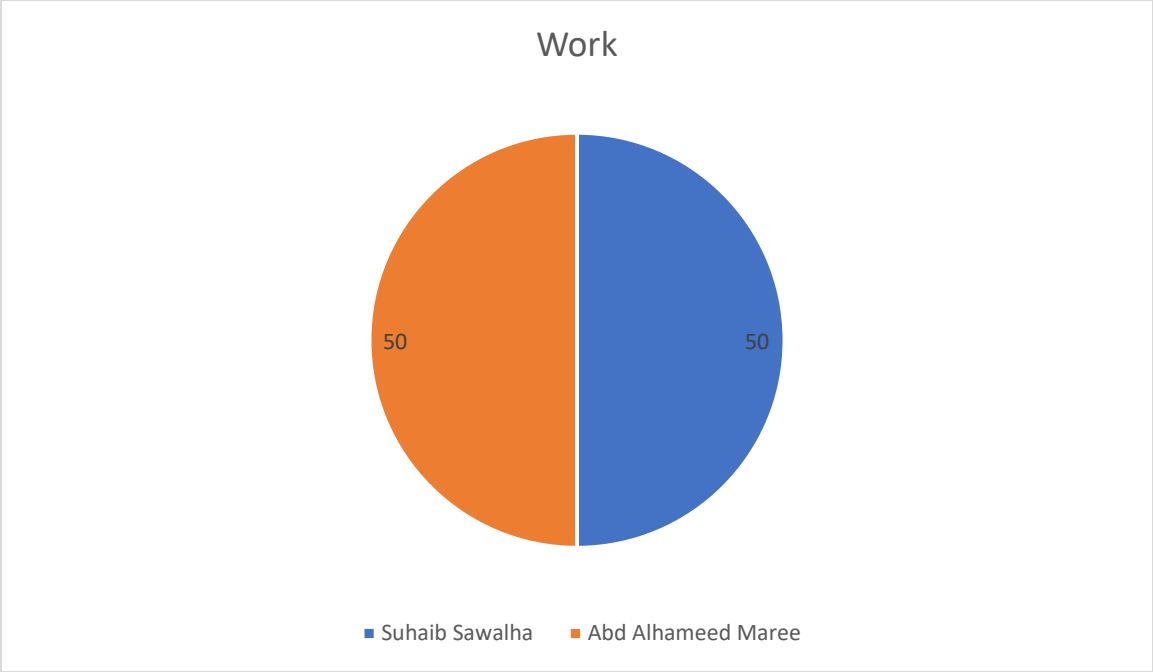

Figure 55: Teamwork.

References

[1]: https://www.simplilearn.com/tutorials/cyber-security-tutorial/understanding-the-networking-commands [Accessed on 16 Aug 2024].

[2]: https://support.n4l.co.nz/s/article/How-to-use-Telnet-to-Check-the-Status-of-Ports [Accessed on 16 Aug 2024].

[3]: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html [Accessed on 16 Aug 2024].

[4]: https://www.geeksforgeeks.org/socket-programming-cc/ [Accessed on 16 Aug 2024].

[5]: https://www.fortinet.com/resources/cyberglossary/tcp-ip [Accessed on 16 Aug 2024].

[6] https://www.fortinet.com/resources/cyberglossary/user-datagram-protocol-udp [Accessed on 16 Aug 2024].

[7]: https://www.avast.com/c-tcp-vs-udp-difference [Accessed on 16 Aug 2024].

[8]: www.indeed.com/career-advice/career-development/what-is-a-peer-to-peer-network [Accessed on 16 Aug 2024].

[9]: https://iximiuz.com/en/posts/writing-web-server-in-python-sockets/ [Accessed on 16 Aug 2024].

[10]: https://developer.mozilla.org/en-US/docs/Web/HTTP [Accessed on 16 Aug 2024].

[11]: https://developer.mozilla.org/en-US/docs/Web/HTML [Accessed on 16 Aug 2024].

[12]: https://developer.mozilla.org/en-US/docs/Web/CSS [Accessed on 16 Aug 2024].