



**Faculty of Engineering & Technology
Electrical & Computer Engineering Department**

**Computer Networks – ENCS3320
Project Report**

Prepared by:	ID Number:	Section:
Aya Hamayel	1221469	2
Sara Shrouf	1220481	3
Ansar Mansour	1220679	4

Date : 28/11/2024

Abstract :

Our project bridges networking principles with practical implementation, focusing on socket programming and server-client interaction. First part, aims to understand network analysis using tools like Command Prompt and Wireshark to explore packet tracking, DNS queries, and server-client communication. Next, we implement socket programming to develop a TCP-based web server that efficiently handles HTTP requests, serving HTML, CSS, and multimedia files. The server supports dynamic content, including localized pages and form handling for user input. Finally, we extend the project to a UDP-based trivia game, showcasing real-time, lightweight communication between clients and the server. This project enhances our grasp of networking while refining practical coding skills.

Table of Contents

Abstract :	1
Theory and Procedure :	5
Network Commands and Wireshark :	5
TELNET :	5
PING :	5
Nslookup :	5
Wireshark:	5
Socket Programming (TCP and UDP) :	6
TCP :	6
UDP :	7
Web server :	8
Task 1_Network Commands and Wireshark :	9
1.1 Command def:	9
1.2 Run Command:	9
1.2.1 : Run the ipconfig /all command.....	9
1.2.2 : Ping.....	10
1.2.3 : Ping discover.engineering.utoronto.ca.....	11
1.2.4 : tarcert :	12
1.2.5 : nslookup :	13
1.2.7Wireshark Interface:	15
❖Task2 – Web Server:	15
2.1 Introduction:	15
Server code and explanation in python:	16
Key Steps:	21
request wep page :	23
wep page :	25
Summary:	31
page in Arabic:	32
Supporting Material page in English.....	36

Access Log:	37
Supporting Material page in Arabic:	38
HTML and SCC CODE :	39
Css _style code :	47
Test our wep server in another device in same local network:	52
Test supporting material photo that exist in server:	54
Test fore material not exist:	55
❖Task 3 – UDP Client-Server Trivia Game Using Socket Programming : 58	
1. Server code :	58
2. Client code :	61
3)Here is an example when running the game :	62
Teamwork :	66
Conclusion:	67
References :	68

Table of figure :

Figure 1 : Requests between browser and server.....	5
Figure 2: <i>State diagram for server and client model of Socket</i>	8
<i>Figure 3:ipconfig/all.....</i>	9
<i>Figure 4: ping a device.....</i>	10
<i>Figure 5: ping a discover.engineering.utoronto.ca.....</i>	11
<i>Figure 6: Tracert Command.....</i>	12
<i>Figure 7: nslookup Command</i>	12
<i>Figure 8: Telnet Command.....</i>	13
<i>Figure 9: Wireshark</i>	14
<i>Figure 10: Basic web server.....</i>	14
<i>Figure 11: Running server.py.....</i>	15
<i>Figure 12: Manages the server's continuos</i>	16
<i>Figure 13: Part1 from handel_request function.....</i>	197
<i>Figure 14: Part2 from handel_request function (search)</i>	18
<i>Figure 15: Part2 from handel_request function.....</i>	19
<i>Figure 16: Part2 from handel_request function.....</i>	20
<i>Figure 17: Search for main page in English</i>	20
<i>Figure 18: Search for main page in English</i>	22
<i>Figure 19: Search for main page in English</i>	23
<i>Figure 20: Search for main page in English</i>	24
<i>Figure 21: Search for main page in English</i>	24
<i>Figure 22: Search for main page in English</i>	25
<i>Figure 23: Search for main page in English</i>	26
<i>Figure 24: Search for main page in English</i>	286
<i>Figure 25: Search for main page in English</i>	296
<i>Figure 26: Server log</i>	297
<i>Figure 27: Page in Arabic.....</i>	327
<i>Figure 28:Supporting Material page in English</i>	36
<i>Figure 29: Access Log</i>	37

<i>Figure 30: Supporting Material page in Arabic</i>	38
<i>Figure 31: Main in Arabic.....</i>	390
<i>Figure 32:Css_style cod</i>	47
<i>Figure 33: Test our wep server in another device.....</i>	52
<i>Figure 34: Test supporting mat photo that exist in server</i>	54
<i>Figure 35: Test for material not exist.....</i>	55
<i>Figure 36: Test for video</i>	57
<i>Figure 37: UDP server.....</i>	58
<i>Figure 38 : Handling Clients</i>	59
<i>Figure 39: gameloop</i>	59
<i>Figure 40 :game code</i>	60
<i>Figure 41:client code</i>	61
<i>Figure 42: test game1</i>	62
<i>Figure 43: test game2</i>	62
<i>Figure 44: game test3</i>	63
<i>Figure 45:game test4</i>	63
<i>Figure 46:game test5</i>	64
<i>Figure 47: game test6</i>	654
<i>Figure 48: game test7</i>	65
<i>Figure 49 :teamwork</i>	66

Theory and Procedure :

Network Commands and Wireshark :

Modern computer systems are interconnected through complex networks, relying on efficient communication channels. Troubleshooting these networks often requires specific tools and commands designed to diagnose and resolve connectivity issues. These commands are accessible via the command-line interface and are essential for network analysis and maintenance.

Some commands that will be used in this project:

TELNET :

TELNET stands for Teletype Network. It is a **client/server application protocol** that provides access to virtual terminals of remote systems on local area networks or the Internet. The local computer uses a telnet client program and the remote computers use a telnet server program.[1]

PING :

Ping is a fundamental command used to test network connectivity between a local system and a remote host. It sends ICMP echo requests to the target and waits for responses, determining the reachability and round-trip time of packets.

Nslookup :

is a useful command for getting information from the DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS-related problems.[2]

Wireshark:

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

You could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable, just like an electrician uses a voltmeter for examining what's happening inside an electric cable (but at a higher level, of course).

In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, that has changed. Wireshark is available for free, is open source, and is one of the best packet analyzers available today.[3]

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.[2]

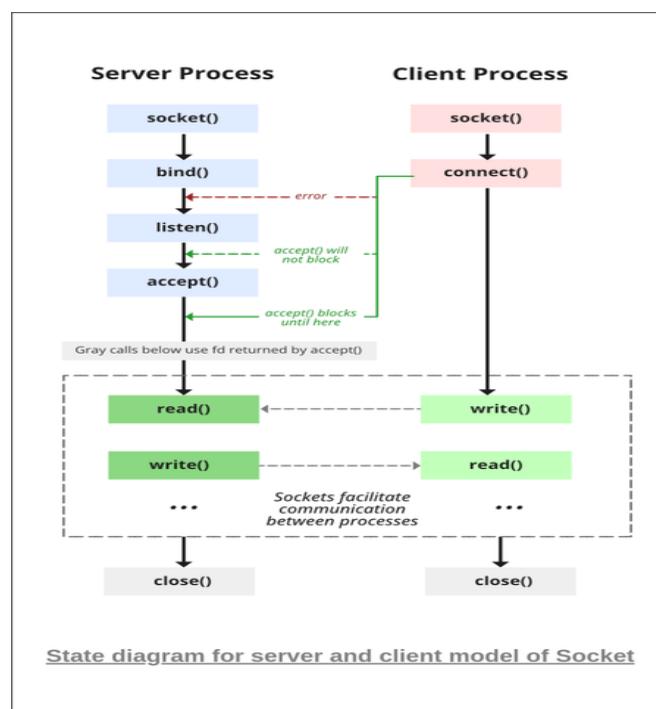


Figure 1: State diagram for server and client model of Socket

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 correction 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 [3]

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.[4]

Web server :

it's a server (no pun intended). A server is a process [*sic*] 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 [5]

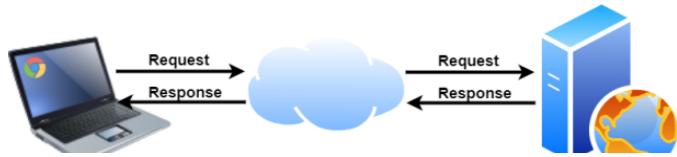


Figure 2: Requests between browser and server

HTTP (Hypertext Transfer 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, such as machine-to-machine communication, programmatic access to APIs, and more [6]

In this project we are built pages t using HTML and CSS

Part 1_Network Commands and Wireshark :

1.1 Command def:

- a) In your own words, provide a brief (two-sentence maximum) explanation of each of the following commands: (i) ipconfig, (ii) ping, (iii) tracert, (iv) telnet, and (v) nslookup.

Answer:-

- i. **ipconfig:** is a command in Windows systems that displays the TCP/IP network configuration of network adapters, including details like IP address and subnet mask. It also allows checking the status of the Network Interface Card (NIC) to ensure proper TCP/IP configuration.
- ii. **Ping:** is a command used to test the connection between two devices on a network, measuring the time taken to send and receive data (RTT) and assessing the connection status and quality.
- iii. **Tracert:** It is a command that shows the path a data packet takes to the destination, along with measuring the response time for each router along the way.
- iv. **Telnet:** is a protocol used to connect to a remote device over a network to control it using the command line, and is used to access remote servers.
- v. **Nslookup (name server lookup):** A command that can be used to fetch the DNS records for a given domain name or IP address.

1.2 Run Command:

- b) Ensure your computer is connected to the internet, then perform the following actions:

1.2.1 : Run the ipconfig /all command

on your computer and identify the IP address, subnet mask, default gateway, and Domain Name System (DNS) server addresses for your primary network interface.

```
C:\Users\HP\man>ipconfig
'ipconfig' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\HP>ipconfig/all

Windows IP Configuration

Host Name . . . . . : AYA
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled . . . . . : No
WINS Proxy Enabled . . . . . : No

Ethernet adapter Ethernet:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . . . . :
  Description . . . . . : Realtek PCIe GBE Family Controller
  Physical Address . . . . . : 6C-02-E0-C9-C8-55
  DHCP Enabled . . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes

Ethernet adapter VEthernet (Wi-Fi):
  Connection-specific DNS Suffix . . . . . : Hyper-V Virtual Ethernet Adapter #3
  Description . . . . . : Intel(R) Dual Band Wireless-AC 7265
  Physical Address . . . . . : 00-15-5D-CB-19-02
  DHCP Enabled . . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes
  Link-local IPv6 Address . . . . . : fe80::44f0:4ed8:72e9:f6f5%41(Preferred)
  IPv4 Address . . . . . : 172.20.80.1(PREFERRED)
  Subnet Mask . . . . . : 255.255.240.0
  Default Gateway . . . . . :
  DNS Servers . . . . . : 687871325
  DHCPv6 IAID . . . . . : 00-01-00-28-28-42-E1-6C-02-E0-C9-C8-55
  DHCPv6 Client DUID . . . . . : fe80:0:0:ffff:1%1
  DNS Servers . . . . . : fe80:0:0:ffff:2%1
  DNS Servers . . . . . : fe80:0:0:ffff:3%1
  NetBIOS over Tcpip. . . . . : Enabled

Wireless LAN adapter Local Area Connection 9:
  Media State . . . . . : Media disconnected
```

Figure 3 :ipconfig/all

❖ Explanation :

In our testing , we utilized the Command Prompt (CMD) to execute the ipconfig /all command

IP Address (IPv4): 172.20.80.1

Subnet Mask: 255.255.240.0

Default Gateway: None is listed (it would be the router's IP address if specified)

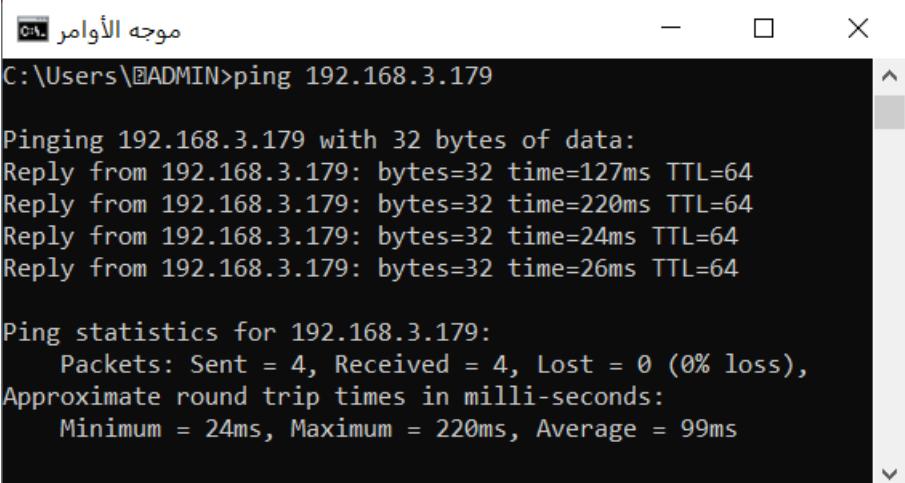
DNS Server Addresses: fec0:0:0:ffff::1%1

fec0:0:0:ffff::2%1

fec0:0:0:ffff::3%1

1.2.2 : Ping

a device within your local network (e.g., from your laptop to a smartphone on the same Wi-Fi network).



```
C:\Users\ADMIN>ping 192.168.3.179

Pinging 192.168.3.179 with 32 bytes of data:
Reply from 192.168.3.179: bytes=32 time=127ms TTL=64
Reply from 192.168.3.179: bytes=32 time=220ms TTL=64
Reply from 192.168.3.179: bytes=32 time=24ms TTL=64
Reply from 192.168.3.179: bytes=32 time=26ms TTL=64

Ping statistics for 192.168.3.179:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 24ms, Maximum = 220ms, Average = 99ms
```

Figure 4: ping a device

❖ Explanation:

I performed a Ping test on my mobile device using the IP address 192.168.3.179. The results were as follows:

4 packets were sent, and 4 were received with no loss (0% packet loss).

The “time” is the round trip time (RTT): which is the time it takes for a packet to travel from client to server and then back to the client.

Minimum time: 24ms (fastest reply)

Maximum time: 220ms (slowest reply)

Average time: 99ms $(127+220+24+26)/4=99$ ms

Bytes = 32 because the default behavior of ping command is to send 32 bytes of data to the destination.

(TTL) stands for “time to live”: it determines when a resource should be removed from a cache.

1.2.3 : Ping discover.engineering.utoronto.ca.

Based on the results, briefly explain whether you believe the response originates from Canada.

```
C:\Users\u>ping discover.engineering.utoronto.ca

Pinging discover.engineering.utoronto.ca [23.185.0.2] with 32 bytes of data:
Reply from 23.185.0.2: bytes=32 time=53ms TTL=251
Reply from 23.185.0.2: bytes=32 time=53ms TTL=251
Reply from 23.185.0.2: bytes=32 time=59ms TTL=251
Reply from 23.185.0.2: bytes=32 time=54ms TTL=251

Ping statistics for 23.185.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 53ms, Maximum = 59ms, Average = 54ms
```

Figure 5: ping a discover.engineering.utoronto.ca

❖ Explanation

The domain discover.engineering.utoronto.ca resolves to the IP address 23.185.0.2, as seen in the ping results. This indicates that the DNS name resolution is functioning correctly.

packets were sent, and all 4 packets were received successfully, indicating that the connection to the server is stable with no packet loss.

Response Times (RTT)

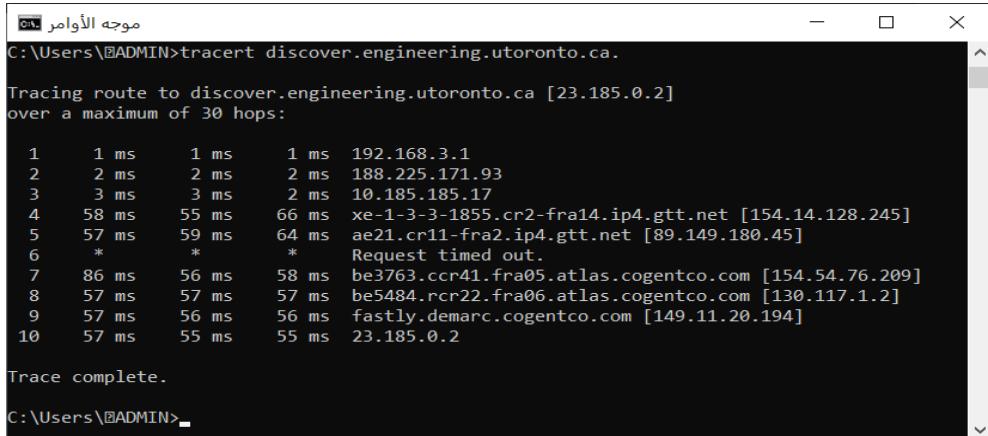
Minimum Response Time: 53ms

Maximum Response Time: 59ms

Average Response Time: 54ms $(53+53+59+54)/4=54$ ms

1.2.4 : tracert :

Run tracert on discover.engineering.utoronto.ca



```
C:\Users\ADMIN>tracert discover.engineering.utoronto.ca.

Tracing route to discover.engineering.utoronto.ca [23.185.0.2]
over a maximum of 30 hops:

  1   1 ms    1 ms    1 ms  192.168.3.1
  2   2 ms    2 ms    2 ms  188.225.171.93
  3   3 ms    3 ms    2 ms  10.185.185.17
  4   58 ms   55 ms   66 ms  xe-1-3-3-1855.cr2-fra14.ip4.gtt.net [154.14.128.245]
  5   57 ms   59 ms   64 ms  ae21.cr11-fra2.ip4.gtt.net [89.149.180.45]
  6   *        *        *      Request timed out.
  7   86 ms   56 ms   58 ms  be3763.ccr41.fra05.atlas.cogentco.com [154.54.76.209]
  8   57 ms   57 ms   57 ms  be5484.rcr22.fra06.atlas.cogentco.com [130.117.1.2]
  9   57 ms   56 ms   56 ms  fastly.demarc.cogentco.com [149.11.20.194]
 10   57 ms   55 ms   55 ms  23.185.0.2

Trace complete.

C:\Users\ADMIN>
```

Figure 6: Tracert Command

❖ Explanation:

The data packets will find its way to the **destination** **discover.engineering.utoronto.ca** and each time it reaches a router on its path it will report back information about that router (The IP address and the time it took between each hop)

The message "Over a maximum of 30 hops" indicates that the tracert command is designed to trace up to 30 hops by default. If the route requires passing through more than 30 routers, the command will stop at the 30th hop unless the maximum limit is manually adjusted.

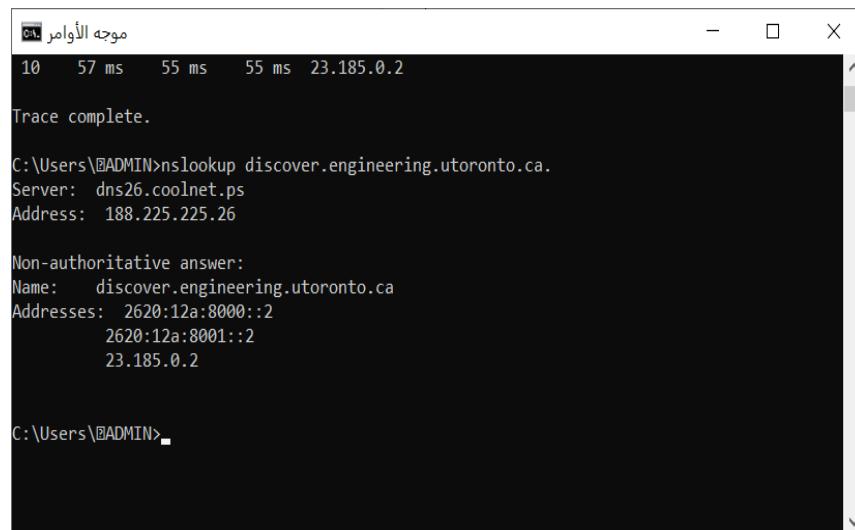
The first column on the left shows the hop number, indicating each router along the path to the destination. In this case, it took 10 hops for the packets to reach the destination **discover.engineering.utoronto.ca**.

The tracert command sends three separate packets to each router, and the second, third, and fourth columns display the Round-Trip Times (RTT) for these packets.

The final column shows the IP address of each router, and sometimes the domain name is also included if available. For example, hop 4 includes `xe-1-3-3-1855.cr2-fra14.ip4.gtt.net`. At hop 6, an asterisk appears, which means no response was received from that router. This can happen if the router is configured to block tracert requests or is temporarily unresponsive.

1.2.5 : nslookup :

Use nslookup to retrieve the DNS information for discover.engineering.utoronto.ca



```
茅جه الأوامر
10 57 ms 55 ms 55 ms 23.185.0.2
Trace complete.

C:\Users\@ADMIN>nslookup discover.engineering.utoronto.ca.
Server: dns26.coolnet.ps
Address: 188.225.225.26

Non-authoritative answer:
Name: discover.engineering.utoronto.ca
Addresses: 2620:12a:8000::2
           2620:12a:8001::2
           23.185.0.2

C:\Users\@ADMIN>
```

Figure 7: nslookup Command

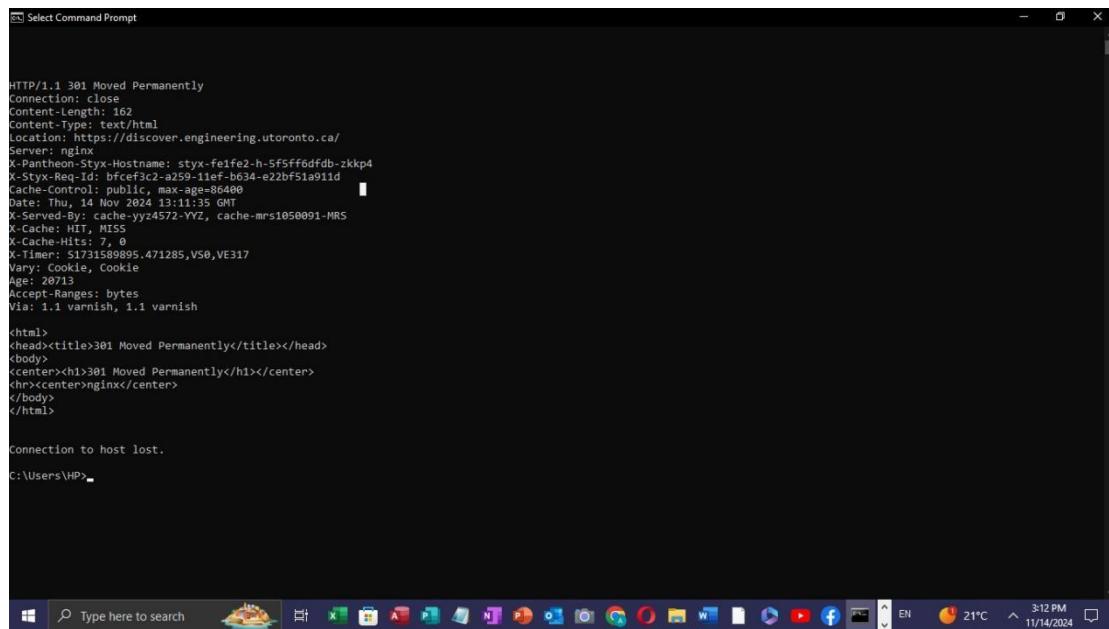
❖ Explanation:

The server I'm communicating with is dns26.coolnet.ps, as shown by the IP address 188.225.225.26.

The nslookup command sends a query to resolve the domain discover.engineering.utoronto.ca, and the server returns the associated IP addresses. The result includes both IPv6 and IPv4 addresses, indicating support for both types.

1.2.6 :Telnet :

Attempt to connect with telnet to discover.engineering.utoronto.ca.



```
HTTP/1.1 301 Moved Permanently
Connection: close
Content-Length: 162
Content-Type: text/html
Location: https://discover.engineering.utoronto.ca/
Server: nginx
X-Pantheon-Styx-Hostname: styx-feife2-h-5f5ff6fdb-zkkp4
X-Styx-Req-Id: bfcef3c2-a259-11ef-b634-e22bf51a911d
Cache-Control: public, max-age=86400
Date: Thu, 14 Nov 2024 13:11:35 GMT
X-Served-By: cache-yzz4572-YZ, cache-mrs1050091-MRS
X-Cache: HIT, MISS
X-Cache-Hits: 7, 0
X-Timer: S1731589895.471285,V50,VE317
Vary: Cookie, Cookie
Age: 20713
Accept-Ranges: bytes
Via: 1.1 varnish, 1.1 varnish

<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<br><center>nginx</center>
</body>
</html>

Connection to host lost.

C:\Users\HP>
```

Figure 8: Telnet Command

❖ Explanation:

When I used **Telnet** to connect to the server, the server responded with a "301 Moved Permanently" message, which means the link I was trying to access has been permanently moved to a different location. The message included the new location to visit, which is <https://discover.engineering.utoronto.ca>

It also indicated that the server sent an HTML page with a size of 162 bytes. Additionally, there were some other details, such as the time the response was sent, which was 13:11:35 GMT, and caching settings that allow storing the content for 86400 seconds.

Finally, after sending the response, the server closed the connection.

1.2.7 Wireshark Interface:

- c. Use the Wireshark packet analyzer to capture a DNS query and reply for any hostname of your choice. Note: You can download the Wireshark packet analyzer from the following link:
<https://www.wireshark.org/download.html>.

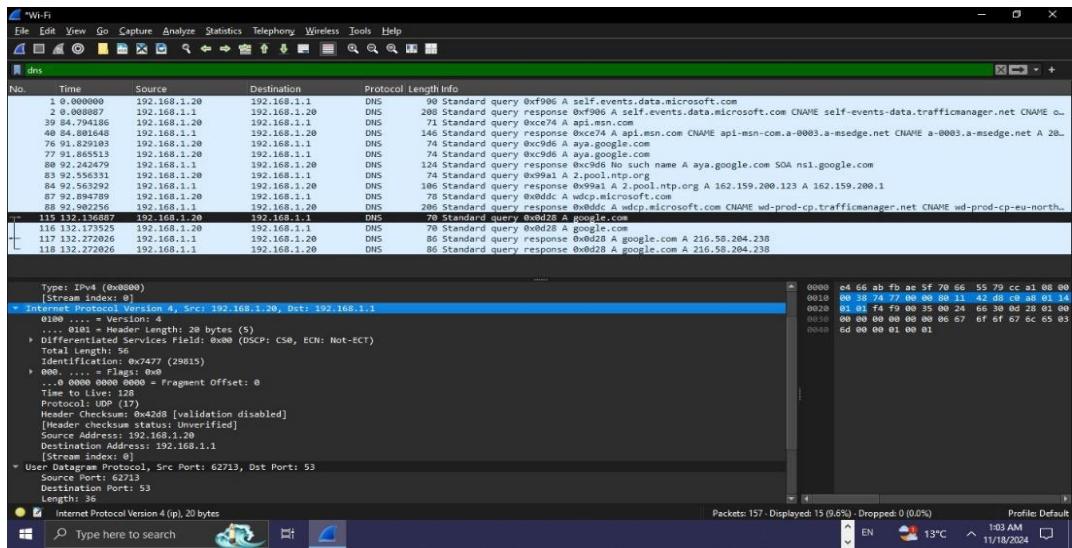


Figure 9: Wireshark

❖ Explanation:

The captured packet is a DNS query sent using the UDP protocol over IPv4. It originated from the device at 192.168.1.20 and was sent to the DNS server at 192.168.1.1 via source port 62713 and destination port 53. The query was for the hostname google.com, with a total packet length of 36 bytes. The Header Checksum has not been verified, and the ECN field is not enabled. This reflects a standard DNS request to retrieve the IP address of google.com.

❖ Part 2 – Web Server:

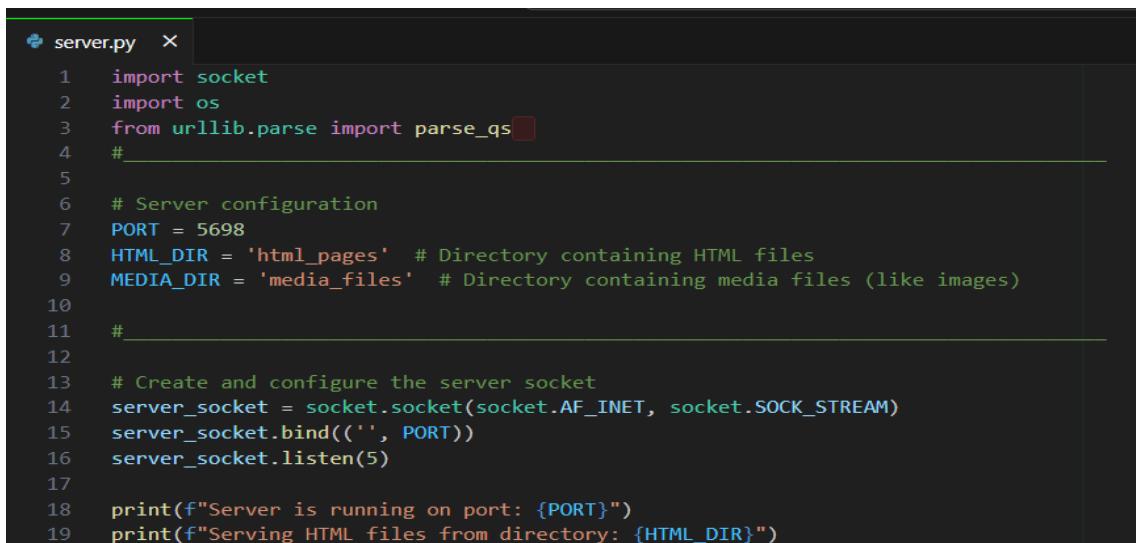
2.1 Introduction:

This task involves the development of a simple yet functional web server using socket programming, designed to listen on port 5698 and handle HTTP requests. The server supports multilingual web pages, including English and Arabic versions, styled with CSS for enhanced user experience. The main functionality includes serving static HTML files, handling form-based requests, and redirecting users to external resources for unavailable files based on their type. Error handling is also implemented to provide meaningful feedback for invalid requests. The project is divided into the following components:

- Main Web Pages:** English and Arabic versions containing structured content, including team details, a topic presentation, and resource links.
- Supporting Material Page:** A form to request multimedia files with conditional redirection for unavailable resources.
- Error Handling:** A custom 404 error page displaying a user-friendly message along with the client's details.

This task emphasizes the integration of HTTP request handling, content delivery, and server-side redirection mechanisms.

Server code and explanation in python:

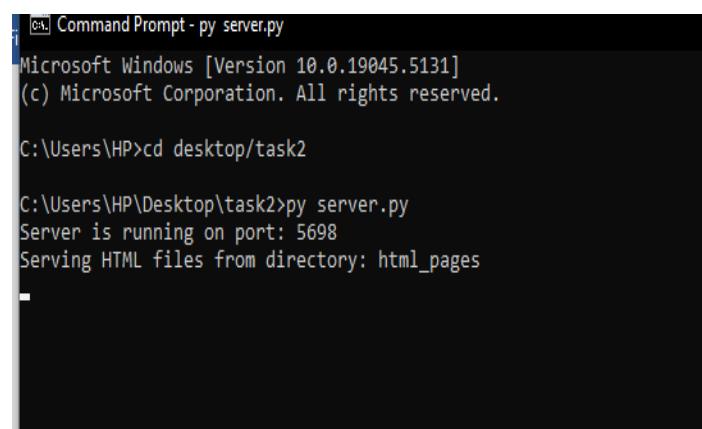


```

server.py  X
1 import socket
2 import os
3 from urllib.parse import parse_qs
4 #
5
6 # Server configuration
7 PORT = 5698
8 HTML_DIR = 'html_pages' # Directory containing HTML files
9 MEDIA_DIR = 'media_files' # Directory containing media files (like images)
10 #
11 #
12 # Create and configure the server socket
13 server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
14 server_socket.bind(('', PORT))
15 server_socket.listen(5)
16
17
18 print(f"Server is running on port: {PORT}")
19 print(f"Serving HTML files from directory: {HTML_DIR}")

```

Figure 10: Basic web server



```

C:\ Command Prompt - py server.py
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd desktop/task2

C:\Users\HP\Desktop\task2>py server.py
Server is running on port: 5698
Serving HTML files from directory: html_pages

```

Figure 11: Running server.py

This section of the code sets up a basic web server using Python's socket programming and incorporates essential libraries for its functionality:

1. Libraries Used:

- ✧ **socket**: Provides tools for network communication, enabling the creation and management of a TCP server socket to handle client requests.
- ✧ **os**: Facilitates interaction with the operating system, such as managing file paths and directories (e.g., HTML_DIR and MEDIA_DIR).
- ✧ **urllib.parse.parse_qs**: Parses query strings from HTTP requests, allowing the server to process user inputs effectively.

2. Server Configuration:

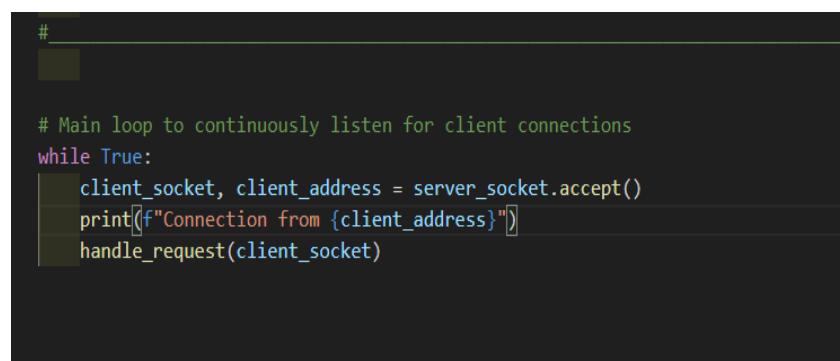
- ✧ **PORT**: The server listens for HTTP requests on port 5698.
- ✧ **HTML_DIR**: Specifies the directory containing the HTML files served to clients.
- ✧ **MEDIA_DIR**: Points to the folder with media files, like images or videos, for use in the web pages.

3. Socket Initialization and Setup:

- ✧ Creates a TCP socket using socket.AF_INET (IPv4) and socket.SOCK_STREAM (stream-based communication).
- ✧ Binds the server socket to the specified port (5698) to accept incoming connections.
- ✧ Configures the server to listen for up to 5 simultaneous connection requests using listen(5).

4. Initialization Feedback:

- ✧ Prints messages confirming that the server is running, the port in use, and the directories for HTML and media files, providing a clear overview of the server's setup.



```
# Main loop to continuously listen for client connections
while True:
    client_socket, client_address = server_socket.accept()
    print(f"Connection from {client_address}")
    handle_request(client_socket)
```

Figure 12: Manages the server's continuos

This segment of the code manages the server's continuous operation, allowing it to handle incoming client requests. Here's how it works:

1. Infinite Loop:

- ❖ The while True: statement creates a continuous loop, ensuring the server stays active and listens for client connections indefinitely.

2. Accepting Client Connections:

- ❖ server_socket.accept(): Waits for a client to initiate a connection. Once a connection is established, it returns:
- ❖ client_socket: A new socket object for communication with the specific client.
- ❖ client_address: The IP address and port number of the connected client.

3. Connection Notification:

- ❖ print(f"Connection from {client_address}"): Logs the client's address, making it easier for the server administrator to monitor activity and debug if necessary.

4. Handling Client Requests:

- ❖ handle_request(client_socket): A placeholder for a function designed to process the client's HTTP request. This typically involves:
 - ❖ Reading the request.
 - ❖ Determining the appropriate response (e.g., serving a file or returning an error).
 - ❖ Sending the response back to the client via client_socket.

➤ Go to handle_request function

//I will divide it in to three part

◆ part1 :

```
# Function to handle incoming requests
def handle_request(client_socket):
    """
    Reads the client request, determines the requested file,
    and sends the appropriate HTTP response.
    """
    request = client_socket.recv(1024).decode() # Read up to 1024 bytes
    print(f"Request received: {request}")
    if not request:
        response = "HTTP/1.1 400 Bad Request\nContent-Type: text/html\n\n"
        response += "<html><body><h1>Error 400</h1><p>Bad Request</p></body></html>"
        client_socket.sendall(response.encode())
        client_socket.close()
        return

    try:
        method, path, _ = request.split(' ', 2) # Extract the method, path, and protocol from the request
        print(f"Method: {method}, Path: {path}, Protocol: {_}") # Log request details
    except ValueError:
        response = "HTTP/1.1 400 Bad Request\nContent-Type: text/html\n\n"
        response += "<html><body><h1>Error 400</h1><p>Bad Request: Missing or malformed request</p></body></html>"
        client_socket.sendall(response.encode())
        client_socket.close()
        return
```

Figure 1: Part1 from handel_request function

This part of the function processes the incoming HTTP request from the client and determines how to respond. Here's how it works:

1. Reading the Client Request:

- ❖ `client_socket.recv(1024).decode()`: Reads up to 1024 bytes of data from the client and decodes it to a string.
- ❖ `print(f"Request received:\n{request}")`: Logs the received request to help monitor server activity or debug issues.

2. Empty Request Handling:

- ❖ If the request is empty (not `request`), the server sends an HTTP 400 Bad Request response to indicate that the client sent an invalid or incomplete request.
- ❖ `client_socket.sendall(response.encode())`: Sends the error response to the client.
- ❖ `client_socket.close()`: Closes the connection after responding.

3. Parsing the Request:

- ❖ `method, path, _ = request.split(' ', 2)`: Splits the request line into three components:
- ❖ `method`: The HTTP method used (e.g., GET).
- ❖ `path`: The requested resource path (e.g., /index.html).
- ❖ `_`: The HTTP protocol version (e.g., HTTP/1.1).
- ❖ `print(f"Method: {method}, Path: {path}, Protocol: {_}")`: Logs the parsed details for further processing or debugging.

4. Malformed Request Handling:

- ❖ If the request cannot be split properly (e.g., it's incomplete or invalid), the server:
 - Sends a 400 Bad Request response indicating the issue.
 - Closes the connection.

◆ Part 2 :

// if request is search

```

4 def handle_request(client_socket):
5     """
6         # Handle GET requests
7
8         # If the path starts with '/search', handle the search functionality
9         if path.startswith('/search'):
10             query = parse_qs(path.split('?', 1)[1]).get('query', [''])[0]
11
12             if not query:
13                 response = "HTTP/1.1 400 Bad Request\nContent-Type: text/html\n\n" \
14                         "<html><body><h1>Error 400</h1><p>No query provided.</p></body></html>"
15                 client_socket.sendall(response.encode())
16                 client_socket.close()
17                 return
18
19             file_path = os.path.join(MEDIA_DIR, query)
20             _, extension = os.path.splitext(file_path)
21             content_types = {
22                 '.png': 'image/png',
23                 '.jpg': 'image/jpeg',
24                 '.jpeg': 'image/jpeg',
25                 '.gif': 'image/gif',
26                 '.bmp': 'image/bmp',
27                 '.mp4': 'video/mp4',
28             }
29             content_type = content_types.get(extension, 'application/octet-stream')
30
31             if os.path.exists(file_path) and os.path.isfile(file_path):
32

```

Figure 13: Part2 from handle_request function (search)

```

if os.path.exists(file_path) and os.path.isfile(file_path):
    if content_type.startswith('image/'):
        with open(file_path, 'rb') as media_file:
            media_data = media_file.read()
        response_headers = f"HTTP/1.1 200 OK\nContent-Type: {content_type}\n\n"
        client_socket.sendall(response_headers.encode() + media_data)

    else:
        if content_type.startswith('video/'):
            youtube_url = f"https://www.youtube.com/results?search_query={query.replace(' ', '+')}"
            response = f"HTTP/1.1 307 Temporary Redirect\nLocation: {youtube_url}\n\n"
            client_socket.sendall(response.encode())
        else:
            google_url = f"https://www.google.com/search?tbm=isch&q={query.replace(' ', '+')}"
            response = f"HTTP/1.1 307 Temporary Redirect\nLocation: {google_url}\n\n"
            client_socket.sendall(response.encode())

    client_socket.close() # Close the connection
    return

```

This segment of code processes **search requests** sent to the server with a query parameter. It retrieves media files (e.g., images or videos) from a specified directory or redirects to external search engines if the requested file isn't available.

Key Steps:

1. Identifying Search Requests:

- ✧ if path.startswith('/search'):: Checks if the client requested the /search endpoint.
For example:

Ex:

GET /search?query=example.jpg HTTP/1.1

This means the user is searching for a file named example.jpg.

2. Extracting the Query Parameter:

- ✧ query = parse_qs(path.split('?', 1)[1]).get('query', [''])[0]:
 - Parses the query string (everything after ?) to extract the query parameter.
 - Example: For ?query=image.png, the extracted query is "image.png".
- ✧ If no query is provided, a 400 Bad Request response is sent to the client.

3. Determining File Path and Content Type:

- ✧ file_path = os.path.join(MEDIA_DIR, query): Constructs the full file path in the media directory.
- ✧ content_types: A dictionary maps file extensions (e.g., .png, .mp4) to appropriate MIME types (e.g., image/png, video/mp4).

4. File Handling:

- ✧ **If the file exists** (os.path.exists(file_path)):
 - Reads the file if it's an image:
 - with open(file_path, 'rb') as media_file:
 - media_data = media_file.read()
- Sends an HTTP 200 OK response with the file data.
- Example: If image.png is requested and exists, the client receives the image with:
 - HTTP/1.1 200 OK
 - Content-Type: image/png
 - [image data]

➤ **If the file doesn't exist:**

Redirects to external search engines:

Videos: Redirect to YouTube:

```
youtube_url= f"https://www.youtube.com/results?search_query={query.replace(' ', '+')}"
```

Example: Searching video.mp4 redirects to:

- ✧ HTTP/1.1 307 Temporary Redirect
- ✧ Location: https://www.youtube.com/results?search_query=video.mp4
- ✧ **Images or other files:** Redirect to Google Images:
- ✧ google_url = f"https://www.google.com/search?tbm=isch&q={query.replace(' ', '+')}"
- ✧ Example: Searching example.jpg redirects to:
- ✧ HTTP/1.1 307 Temporary Redirect
- ✧ Location: https://www.google.com/search?tbm=isch&q=example.jpg

5. Closing the Connection:

After handling the request, the server closes the connection with the client:client_socket.close()

Example Scenarios:

1. File Found:

Request: /search?query=image.png

File image.png exists in the MEDIA_DIR.

Response: Returns the image data with 200 OK.

2. File Not Found (Video):

Request: /search?query=video.mp4

File video.mp4 is missing.

Response: Redirects to YouTube with a 307 Temporary Redirect.

3. File Not Found (Image):

Request: /search?query=example.jpg

File example.jpg is missing.

Response: Redirects to Google Images with a 307 Temporary Redirect.

request wep page :

```
# Default behavior for other requests (HTML and CSS files)
file_mapping = {
    '/main_en.html': 'main_en.html',
    '/': 'main_en.html',
    '/index.html': 'main_en.html',
    '/en': 'main_en.html',
    '/main_ar.html': 'main_ar.html',
    '/ar': 'main_ar.html',
    '/supporting_material_en.html': 'supporting_material_en.html',
    '/supporting_material_ar.html': 'supporting_material_ar.html',
}

if path.endswith('.css'):
    file_path = path.lstrip('/') # Directly map CSS requests
    full_path = os.path.join(HTML_DIR, file_path)
else:
    file_path = file_mapping.get(path, '404.html')
    full_path = os.path.join(HTML_DIR, file_path)

if os.path.exists(full_path):
    _, extension = os.path.splitext(full_path)
    content_types = {
        '.html': 'text/html',
        '.css': 'text/css',
    }
```

Figure 14: Part2 from handle_request function

```
}
content_type = content_types.get(extension, 'text/plain')

with open(full_path, 'r', encoding='utf-8') as file:
    content = file.read()
    response = f"HTTP/1.1 200 OK\nContent-Type: {content_type}\n\n{content}"
else:
    response = "HTTP/1.1 404 Not Found\nContent-Type: text/html\n\n" \
               "<html><body><h1>Error 404</h1>" \
               "<p style='color: red;'>The file is not found</p></body></html>"

client_socket.sendall(response.encode())
client_socket.close()

#
```

Figure 12: Part2 from handle_request function

Code Explanation: Handling Default Requests for HTML and CSS Files

This part of the code handles incoming HTTP requests for **HTML** and **CSS** files and ensures the appropriate file is served or a 404 Not Found error is returned.

Key Steps:

1. **Path Mapping:**

- A **dictionary (file_mapping)** is used to map common request paths like /, /en, or /ar to their respective files, such as main_en.html or main_ar.html.
- If a requested path isn't explicitly mapped, the server defaults to serving a 404.html file.

2. CSS Handling:

- If the requested path ends with .css, it is treated as a direct CSS file. The full path is constructed by combining the base directory (HTML_DIR) and the stripped file name.

3. File Verification:

- The code checks whether the requested file exists using:
- if os.path.exists(full_path):
- If the file exists:

The **file type** is determined by its extension (e.g., .html, .css).

The file is read and sent to the client with an HTTP 200 OK response.

- If the file doesn't exist:

A 404 Not Found error page is returned, with a message styled in red.

4. Response Generation:

Valid files: Sent with appropriate content type (text/html for HTML, text/css for CSS).

Missing files: Returns a default HTML error page with a 404 status.

Example Scenarios:

1. Request for /en:

Mapped to main_en.html.

The server reads the file and sends it with a 200 OK response.

2. Request for /style.css:

Treated as a CSS file.

If style.css exists, it is served with a 200 OK response and Content-Type: text/css.

3. Request for /unknown:

Not mapped in file_mapping.

The server returns a 404 Not Found response with a red error message.

wep page :

1. Serrch for main page in English:

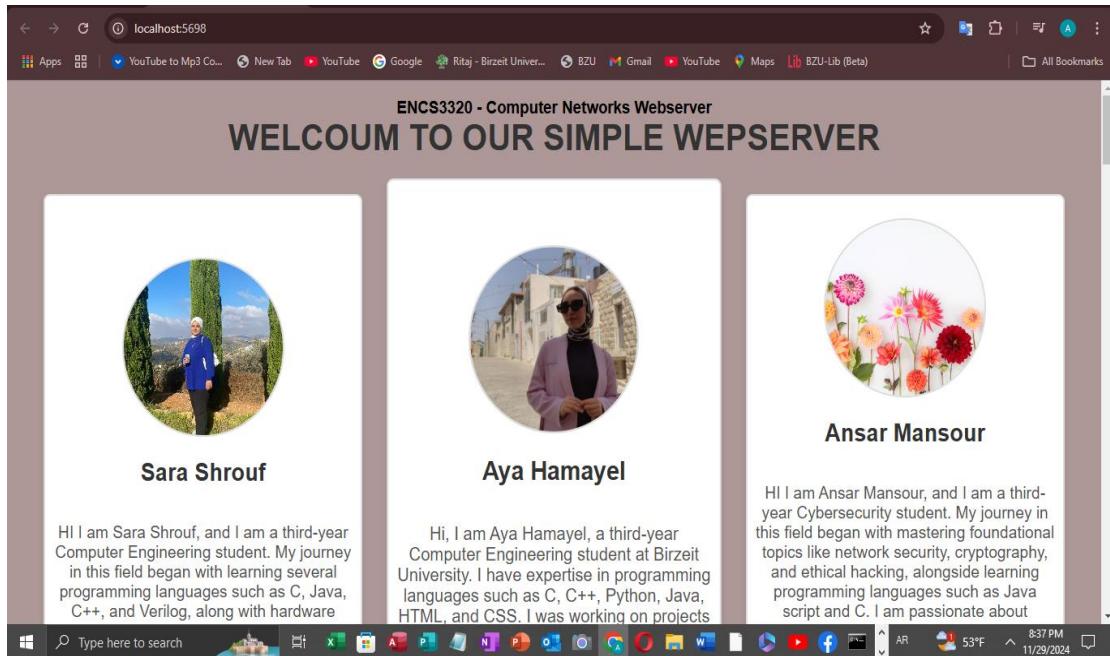


Figure 13: Search for main page in English

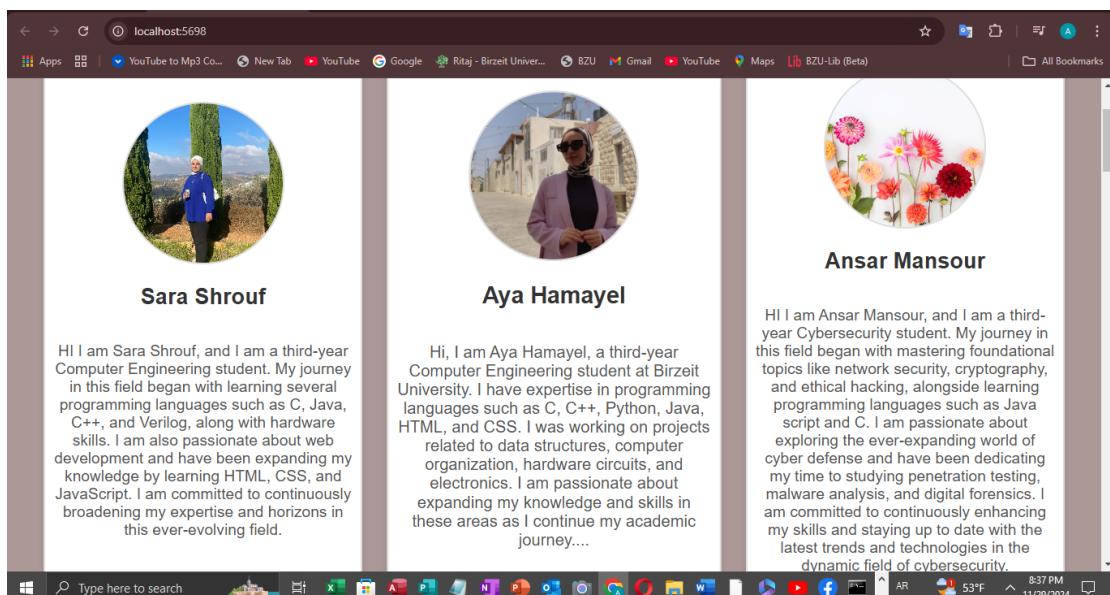


Figure 14: Search for main page in English



Figure 15: Search for main page in English

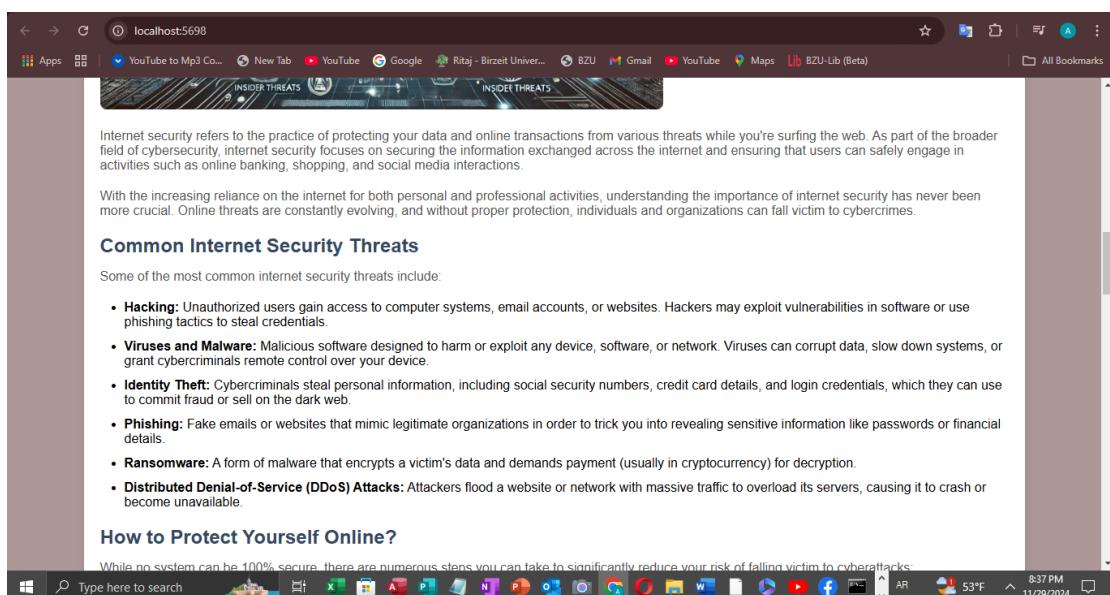


Figure 16: Search for main page in English

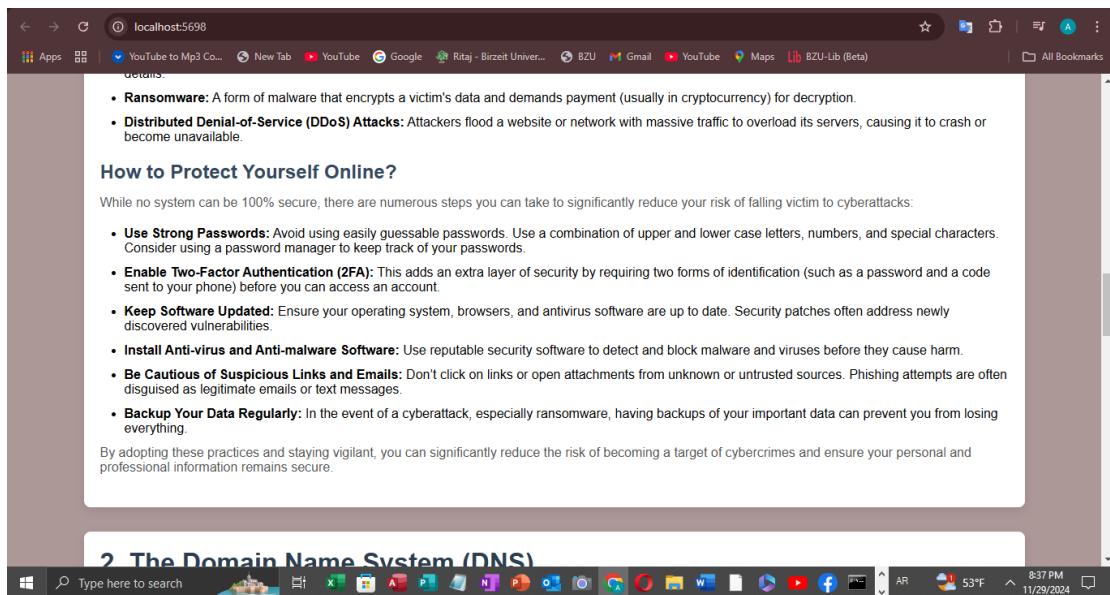


Figure 20: Search for main page in English

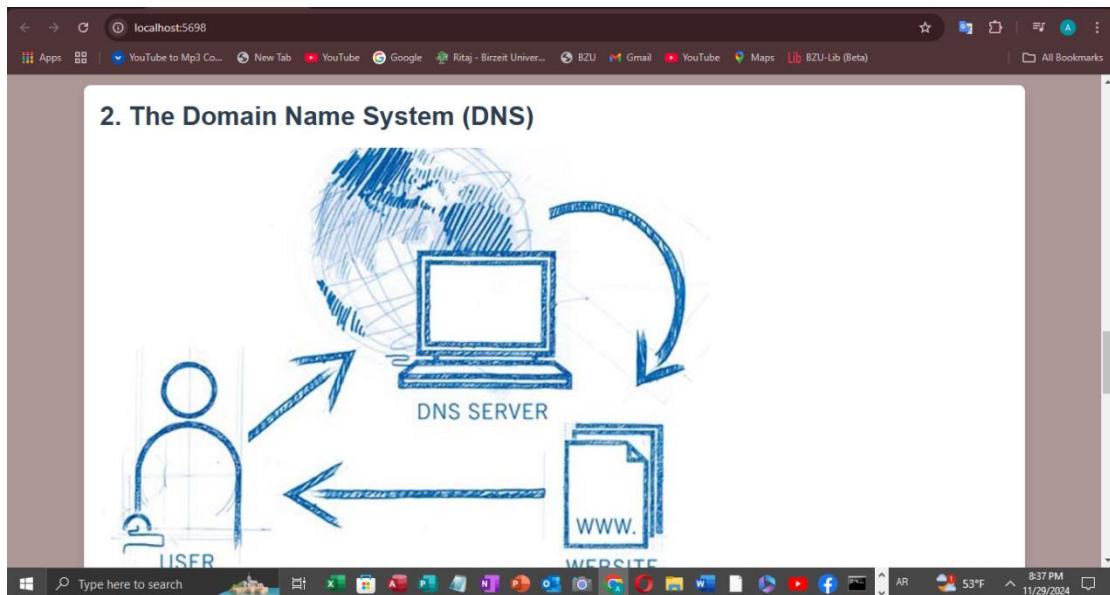


Figure21: Search for main page in English

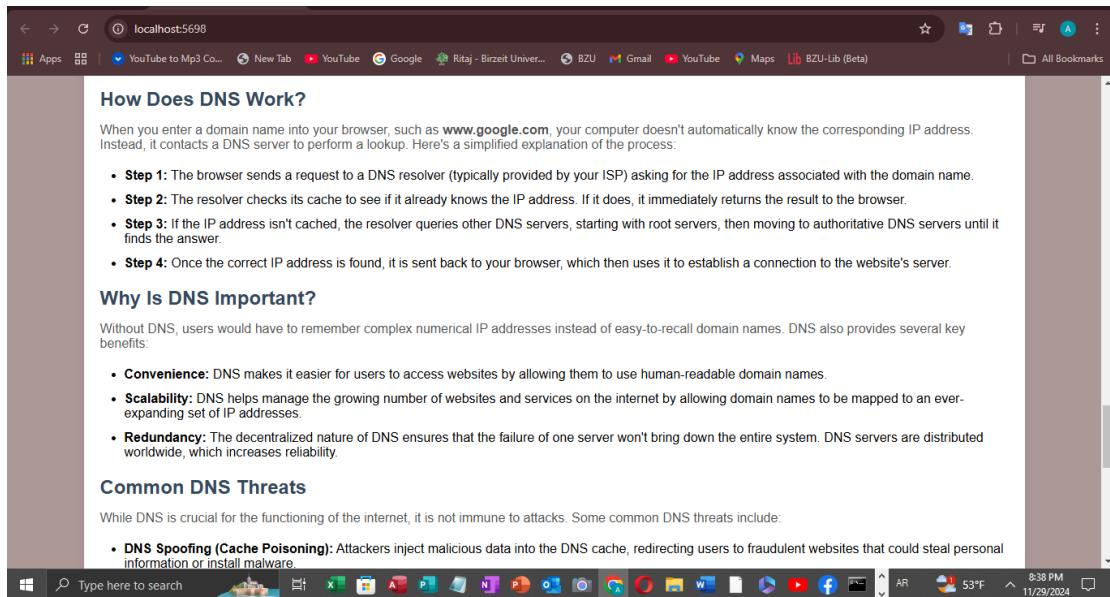


Figure 22: Search for main page in English

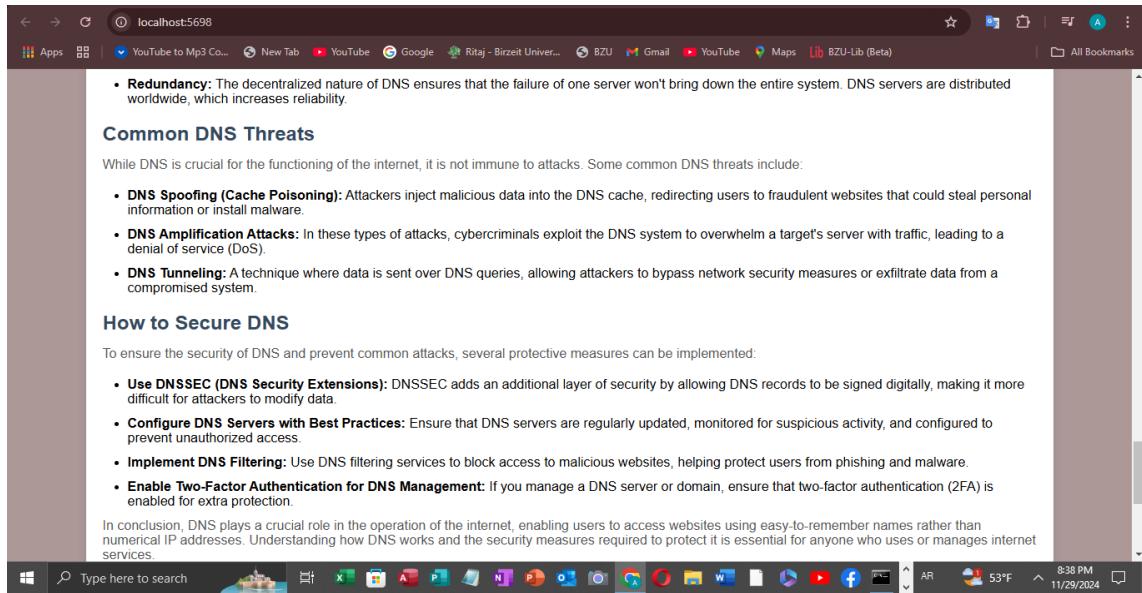


Figure 23: Search for main page in English

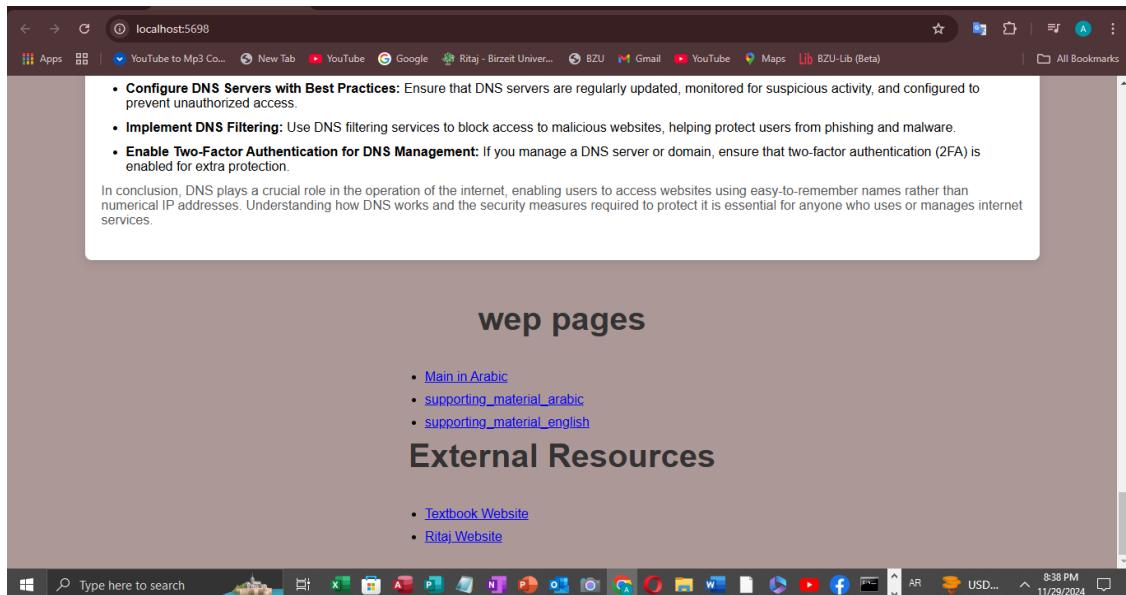


Figure 7: Search for main page in English

➤ After request this page :

This is an Access Log:

An Access Log is a file that records detailed information about the requests sent to a web server over the HTTP protocol. It includes data such as the client's IP address, the type of request (e.g., GET, POST), the requested path, as well as additional information like the user agent (browser type), and headers sent with the request. This log is vital for monitoring server performance, diagnosing errors, and analyzing user behavior. It helps administrators in troubleshooting issues, optimizing performance, and improving the user experience.

```
PS C:\Command Prompt - py server.py
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd desktop\task2

C:\Users\HP\Desktop\task2>py server.py
Server is running on port: 5698
Serving static files from directory: html_pages
Connection from ('127.0.0.1', 51784)
Request received:
GET / HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?
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/131.0.0.0 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: cross-site
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?
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/131.0.0.0 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: cross-site
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6
```

Figure 8: Server log

```

[cmd] Select Command Prompt - py server.py
GET /styles.css HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /styles.css, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 51790)
Request received:
GET /search?query=sara.jpg HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /search?query=sara.jpg, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 51792)
Request received:
GET /search?query=aya.jpg HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /search?query=aya.jpg, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 51792)
Request received:
GET /search?query=ansar.jpg HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

```

```

[cmd] Select Command Prompt - py server.py
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /search?query=aya.jpg, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 51792)
Request received:
GET /search?query=aya.jpg HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /search?query=aya.jpg, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 51792)
Request received:
GET /search?query=ansar.jpg HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

```

Here is a summary of the server log in a more organized manner:

Server Log Summary:

- **Server Status:**
 - The server is running on port **5698**.
 - Serving HTML files from the **html_pages** directory.

Requests:

1. **Connection** from IP: 127.0.0.1 (localhost):
 - o **Request 1:** GET / HTTP/1.1
 - Requested Path: / (root page)
 - **Host:** localhost:5698
2. **Connection** from IP: 127.0.0.1 (localhost):
 - o **Request 2:** GET /styles.css HTTP/1.1
 - Requested Path: /styles.css (CSS file)
 - **Host:** localhost:5698
3. **Connection** from IP: 127.0.0.1 (localhost):
 - o **Request 3:** GET /search?query=sara.jpg HTTP/1.1
 - Requested Path: /search?query=sara.jpg (image search)
 - **Host:** localhost:5698
4. **Connection** from IP: 127.0.0.1 (localhost):
 - o **Request 4:** GET /search?query=aya.jpg HTTP/1.1
 - Requested Path: /search?query=aya.jpg (image search)
 - **Host:** localhost:5698
5. **Connection** from IP: 127.0.0.1 (localhost):
 - o **Request 5:** GET /search?query=ansar.jpg HTTP/1.1
 - Requested Path: /search?query=ansar.jpg (image search)
 - **Host:** localhost:5698
6. **Connection** from IP: 127.0.0.1 (localhost):
 - o **Request 6:** GET /search?query=subject1.png HTTP/1.1
 - Requested Path: /search?query=subject1.png (image search)
 - **Host:** localhost:5698
7. **Connection** from IP: 127.0.0.1 (localhost):
 - o **Request 7:** GET /search?query=dns.png HTTP/1.1
 - Requested Path: /search?query=dns.png (image search)
 - **Host:** localhost:5698

Key Observations:

- **Requests** show that the client (browser) is interacting with the server to fetch the main page (/), styles (styles.css), and images (like sara.jpg, aya.jpg, ansar.jpg, etc.).
- All requests come from 127.0.0.1, which refers to the local machine (localhost).
- The **Host** header indicates the server's address (localhost:5698), which is being accessed via port 5698.

Summary:

The server is receiving requests for the main page and various resources (CSS, images), all from the same machine (localhost). The server responds to these requests with the appropriate files.

page in Arabic:

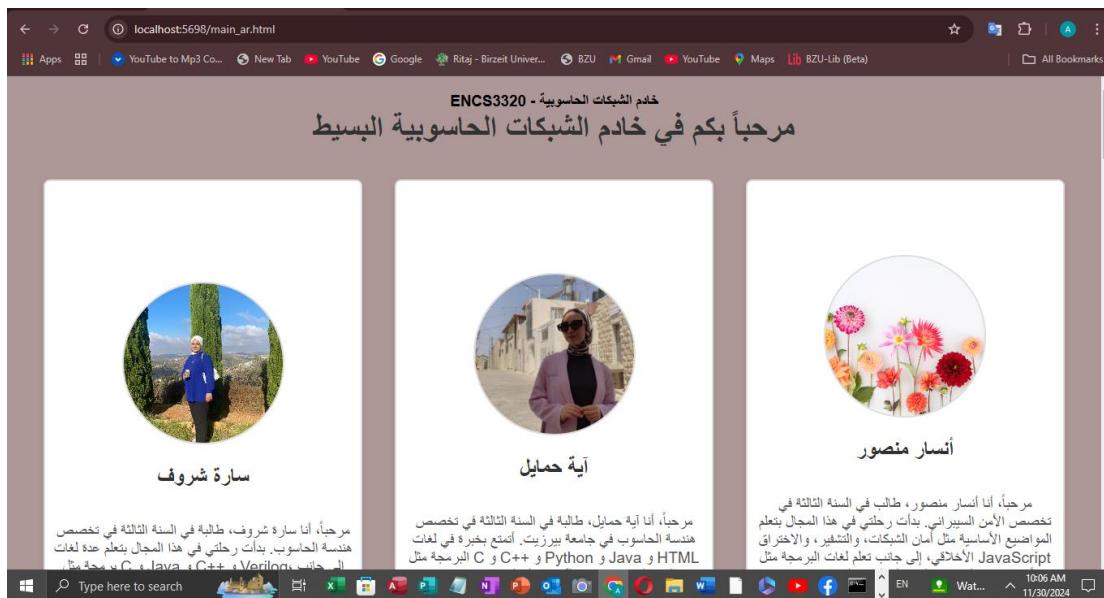
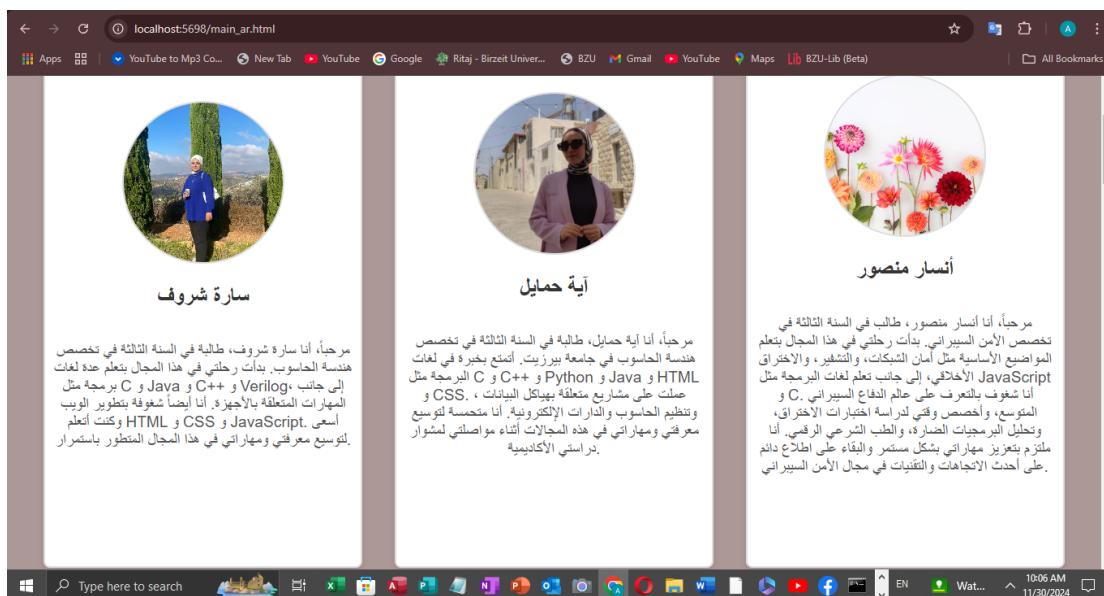


Figure 9: Page in Arabic



العنوان: سناش موضوعي في الشبكات

1. أمان الانترنت: ما هو وكيف يمكن حماية نفسك عبر الانترنت؟



يمكن المستخدمون عبر المتصفحات المختلفة حماية أنفسهم من المخاطر عبر الإنترنت، مثل البريد الإلكتروني أو حسابات المواقع الإلكترونية.

في هذه المقدمة، سنوضح لك بعض النصائح والطرق التي يمكنك اتخاذها لحماية نفسك عبر الإنترنت.

بعض النصائح الأساسية:

- الغوصة: ينصح بتجنب فتح الروابط المجهولة أو غير المألوفة.
- الفيروسات والبرمجيات الخبيثة: ينصح بتنزيل برامج مكافحة الفيروسات.
- سرقة الهوية: ينصح بحفظ كلمات المرور في مكان آمن.
- التصدي: ينصح بتنزيل برامج مكافحة التجسس.
- البرمجيات الخبيثة: ينصح بتنزيل برامج مكافحة التجسس.
- الهجمات التي تهدى إلى إدخال المحتوى الضار: ينصح بتنزيل برامج مكافحة التجسس.

للحاجة إلى مزيد من المعلومات، يرجى زيارة المواقع المنشورة في المقدمة.

التهديدات الشائعة لأنان الانترنت

بعض التهديدات الشائعة لأنان الانترنت:

- الغوصة: ينصح بتجنب فتح الروابط المجهولة أو غير المألوفة.
- الفيروسات والبرمجيات الخبيثة: ينصح بتنزيل برامج مكافحة الفيروسات.
- سرقة الهوية: ينصح بحفظ كلمات المرور في مكان آمن.
- التصدي: ينصح بتنزيل برامج مكافحة التجسس.
- البرمجيات الخبيثة: ينصح بتنزيل برامج مكافحة التجسس.
- الهجمات التي تهدى إلى إدخال المحتوى الضار: ينصح بتنزيل برامج مكافحة التجسس.

للحاجة إلى مزيد من المعلومات، يرجى زيارة المواقع المنشورة في المقدمة.

كيفية حماية نفسك عبر الانترنت؟

رغم أنه لا يوجد نظام يمكن أن يكون آمناً بنسبة 100%، إلا أنه هناك العديد من الخطوات التي يمكنك اتخاذها لتقليل المخاطر بشكل كبير.

استخدم كلمات مرور قوية: يجب استخدام كلمات مرور مبتكرة ومتعددة.

يُنصح بتنزيل برامج مكافحة الفيروسات.

تحديث البرمجيات بانتظام: تأكد من تحديث نظام التشغيل والبرمجيات.

ثبات برامج مكافحة الفيروسات والبرمجيات الخبيثة: استخدم برامج مكافحة الفيروسات.

احتفظ بنسخ احتياطية لبياناتك: في حال تعرضاً لهجوم سيريري، يمكنك تقليل المخاطر.

من خلال اعتماد هذه الممارسات والبقاء بقى، يمكنك تقليل المخاطر بشكل كبير وضمان أن معلوماتك الشخصية والمادية.

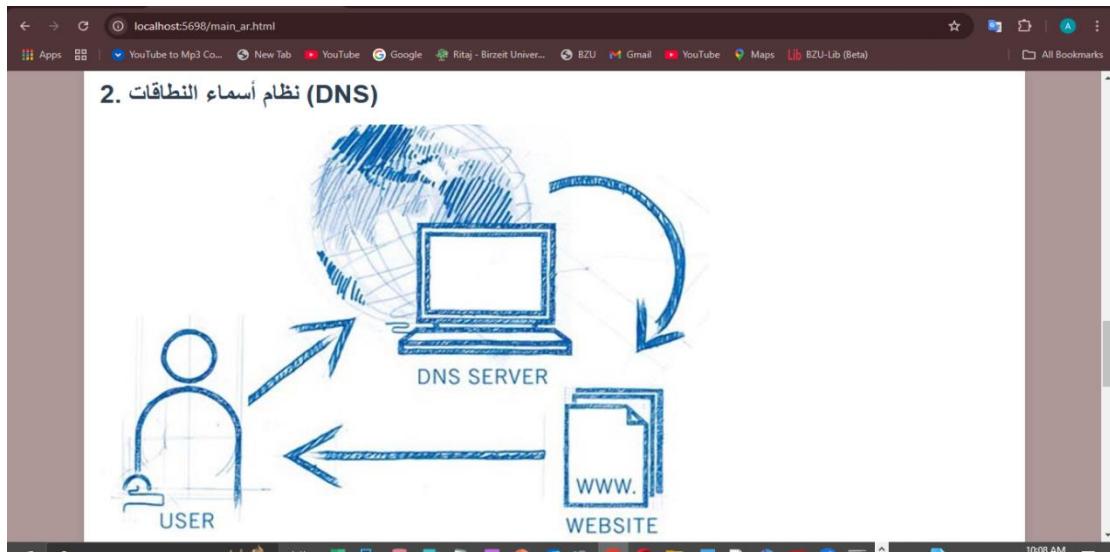
2. نظام أسماء النطاقات (DNS)



يعتبر نظام DNS أحد أكثر الأنظمة إثارة للجدل في الآونة الأخيرة، حيث يتيح تغيير اسم الموقعي إلى عنوان IP.

يمكن استخدام DNS لتنزيل الملفات من مواقع الآخرين.

يمكن تنزيل الملفات من مواقع الآخرين.



مثال) إلى عارفين (www.example.com) هو أحد المكونات الأساسية في بنية الإنترنت. يمثل كـ "لبل الهاتف" لمتصفحك، حيث يترجم أسماء النطاقات التي يمكن قراءتها بسهولة (DNS) إلى عارفين (IP) إلى عارفين (IP) مطلوب عارف (عادةً ما توفره مزود خدمة الإنترنت).
المرتبط باسم النطاق IP لمطلب عارف (عادةً ما توفره مزود خدمة الإنترنت) DNS الخطوة 1: يرسل المتصفح مطلباً إلى موجه • إذا كان يعرفه، يقوم برجوع النتيجة على الفور إلى المتصفح IP. الخطوة 2: يتحقق الموجه من ذاكرةه للتأكد من صحة عارف • الأخرى الخطوة 3: إذا لم يكن الموجه في الذاكرة، يقوم الموجه بالاستفسار من موجه • الصحيح، يتم إرساله إلى المتصفح الذي يستخدمه لإنشاء الاتصال بالحاسوب IP الخطوة 4: بمجرد التحول على عارف •

كيف يعمل DNS؟

لإجراء عملية البحث، عليك شرح مبسط للمعلمية DNS المقابل. بدلاً من ذلك، يتصل بخاتم IP لا يعرف جهاز الكمبيوتر الذي تطالعه عارف www.google.com. عند إدخال اسم العارف في متصفحك مثل

- المرتبط باسم النطاق IP لمطلب عارف (عادةً ما توفره مزود خدمة الإنترنت) DNS الخطوة 1: يرسل المتصفح مطلباً إلى موجه
- إذا كان يعرفه، يقوم برجوع النتيجة على الفور إلى المتصفح IP. الخطوة 2: يتحقق الموجه من ذاكرةه للتأكد من صحة عارف
- الأخرى الخطوة 3: إذا لم يكن الموجه في الذاكرة، يقوم الموجه بالاستفسار من موجه
- الصحيح، يتم إرساله إلى المتصفح الذي يستخدمه لإنشاء الاتصال بالحاسوب IP الخطوة 4: بمجرد التحول على عارف

لماذا يعتبر DNS مهمًا؟

الحدث من الفوائد DNS الجديدة بدلاً من أسماء النطاقات البسيطة، كما يوفر IP كـ "كتاب عنوان" DNS بين المستخدمين الذين يستخدمون متصفحهم إلى تذكر عارفين DNS.

- الوصول إلى الموقع أسهل للمستخدمين من خلال استخدام أسماء النطاقات القليلة للقراءة البشريّة الراحة: يجلب
- في إدارة العدد المتزايد من المواقع والخدمات على الإنترنت DNS المقليّة للتيسير: يساعد
- الموزع عالمياً أن يقبل أحد الخوادم لا يزور إلى تحفيظ النظام بالكامل DNS المرونة: يضمن نظام

التهديدات الشائعة لـ DNS

يتمثل DNS في أنه ليس محملاً ضد الهجمات. بعض التهديدات الشائعة لـ DNS على الرغم من أهميتها

- لإعادة توجيه المستخدمين إلى مواقع احتيالية قد تسرق المعلومات الشخصية أو ثبّت البرمجيات الضارة DNS يقوم المهاجمون بإدخال بيانات شائكة في ذاكرة: DNS التصدير غير

التهديدات الشائعة لـ DNS

تتمثل DNS في أنه أداة محسّنة ضد الهجمات، يحصل التهديدات الشائعة لـ DNS على الرغم من أهميتها.

- لإعادة توجيه المستخدمين إلى موقع احتيالي قد تسرى المعلومات الشخصية أو ثبت البرمجيات المخالفة DNS يقوم المهاجمون بخداع بروتوكول حفارة في ذاكه: **DNS**: التضليل غير.
- يُستغل الفرصة هذه الهجمات لتحميل الخادم المستهدف بحربة مرور منتهية: **DNS**: هجمات تعزير.
- لحجز كبير الأمان DNS يسمح هذا الأسلوب بتغيير البيانات غير انتقادات: **DNS**: ناق.

كيفية تأمين DNS

ومنع الهجمات الشائعة، يمكن تنفيذ التدابير الوقائية التالية لحماية DNS.

- طريقة آمنة إضافية من خلال التوقيع الرقمي للسجلات، مما يجعل من الصعب على المهاجمين تغيير البيانات: **DNSSEC** يوفر استخدام.
- باتباعها لافتتاحية الشبكة DNS بأفضل الظروف: **DNS**: إعداد خوادم.
- لتحقيق الوصول إلى الموقع العادي DNS اشتق خدمة متصفح: **DNS**: تصفيقة.
- ذلك من تمكن المساعدة الثالثية توفير أمان إضافي: **DNS**: إذا كنت تدير خادم DNS: تأمين المصادقة الثنائية إدارة.

والإجراءات الأمنية اللازمة لحماية أمراً DNS الرقمية. يدفهم كيفية عمل IP ذروياً في عمل الإنترنت، مما يتيح للمستخدمين الوصول إلى الموقع باستخدام أسماء سهلة التذكر بدلاً من عارفين DNS في الخادم، يطلب إضافياً لكل من يستخدم أو يدير خدمات الإنترنت.

صفحات الويب

صفحات الويب

طريقة آمنة إضافية من خلال التوقيع الرقمي للسجلات، مما يجعل من الصعب على المهاجمين تغيير البيانات: **DNSSEC** يوفر استخدام.

- باتباعها لافتتاحية الشبكة DNS بأفضل الظروف: **DNS**: إعداد خوادم.
- لتحقيق الوصول إلى الموقع العادي DNS اشتق خدمة متصفح: **DNS**: تصفيقة.
- ذلك من تمكن المساعدة الثالثية توفير أمان إضافي: **DNS**: إذا كنت تدير خادم DNS: تأمين المصادقة الثنائية إدارة.

والإجراءات الأمنية اللازمة لحماية أمراً DNS الرقمية. يدفهم كيفية عمل IP ذروياً في عمل الإنترنت، مما يتيح للمستخدمين الوصول إلى الموقع باستخدام أسماء سهلة التذكر بدلاً من عارفين DNS في الخادم، يطلب إضافياً لكل من يستخدم أو يدير خدمات الإنترنت.

صفحات الويب

- صفحة الرئيسية باللغة الإنجليزية
- صفحة المواد الداعمة بالعربي
- صفحة المواد الداعمة بالإنجليزي

الموارد الخارجية

- موقع الكتاب الدراسي
- موقع منتدى

Supporting Material page in English

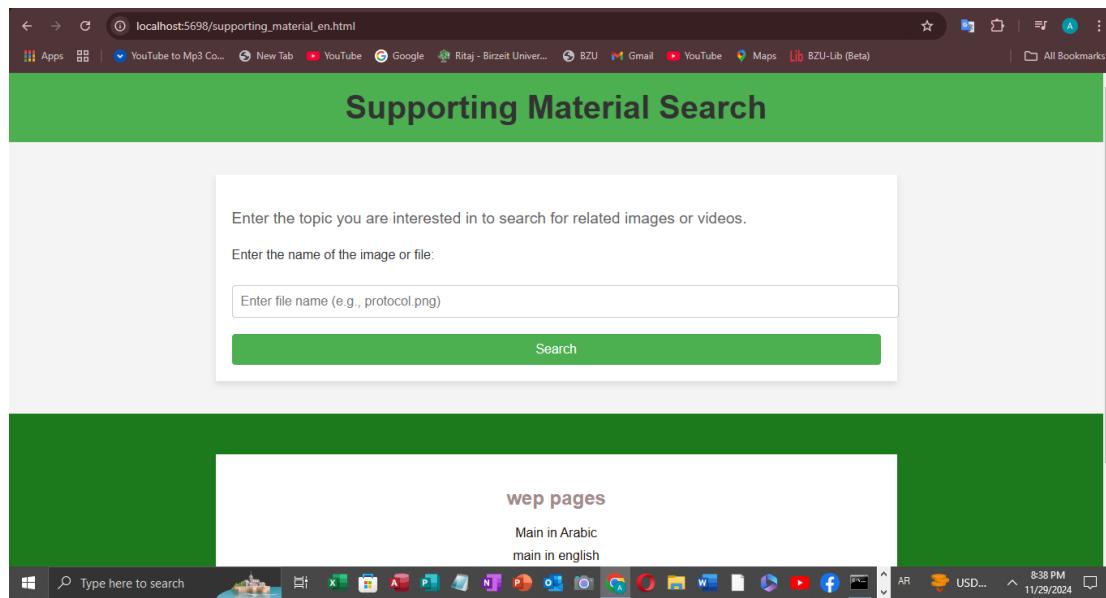
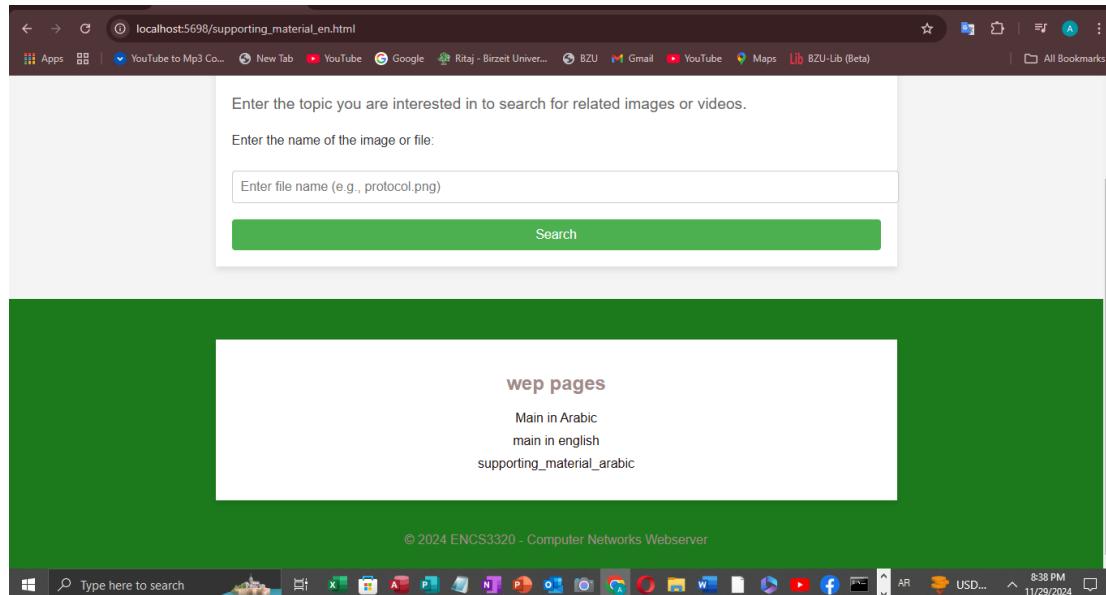


Figure 10: Supporting Material page in English



Access Log:

```
cmd Select Command Prompt - py server.py
Request received:
GET /supporting_material_en.html HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "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/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: document
Sec-Fetch-Policy: strict-unsafe
Referer: http://localhost:5698/main_ar.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /supporting_material_en.html, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "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/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:5698/main_ar.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 52026)
Request received:
GET /supp_styles.css HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /supp_styles.css, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6
```

Figure 11: Access Log

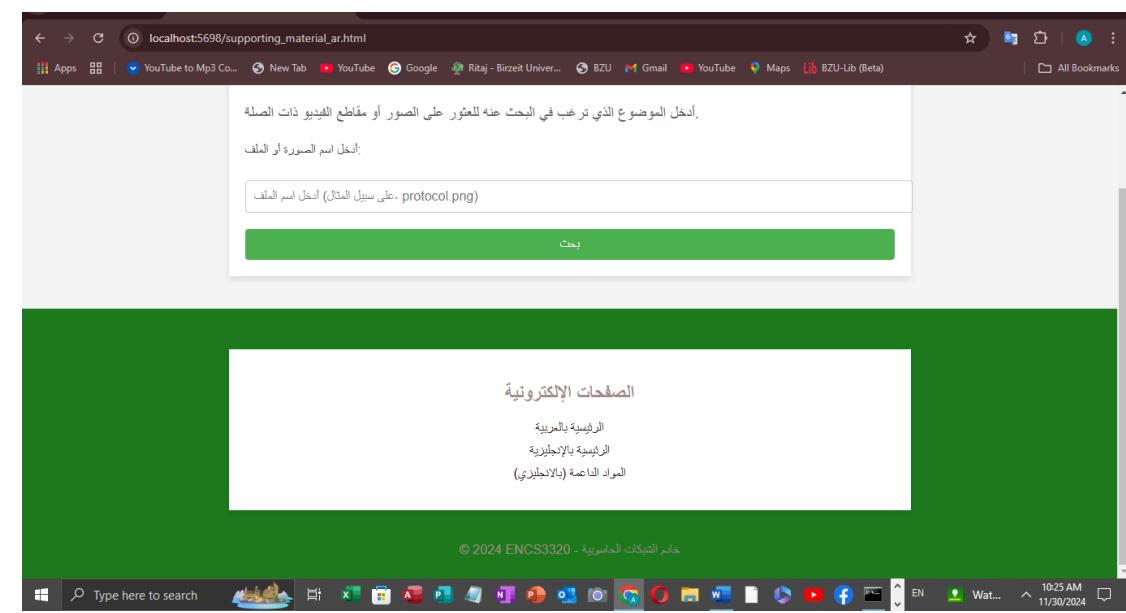
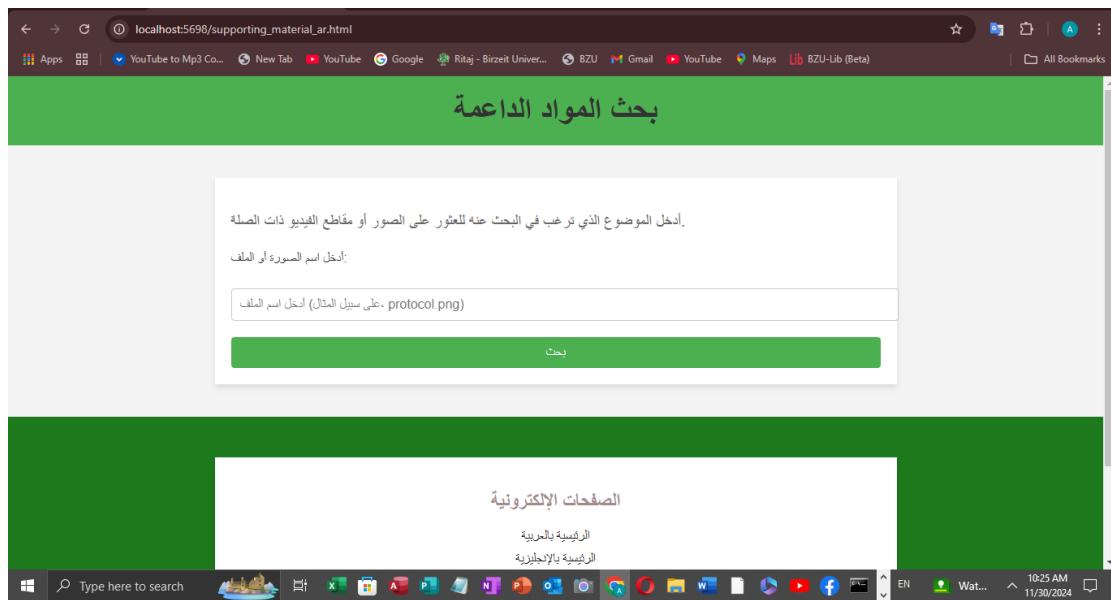
```
cmd Select Command Prompt - py server.py
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:5698/main_ar.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 52026)
Request received:
GET /supp_styles.css HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /supp_styles.css, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua-platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?0
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6
```

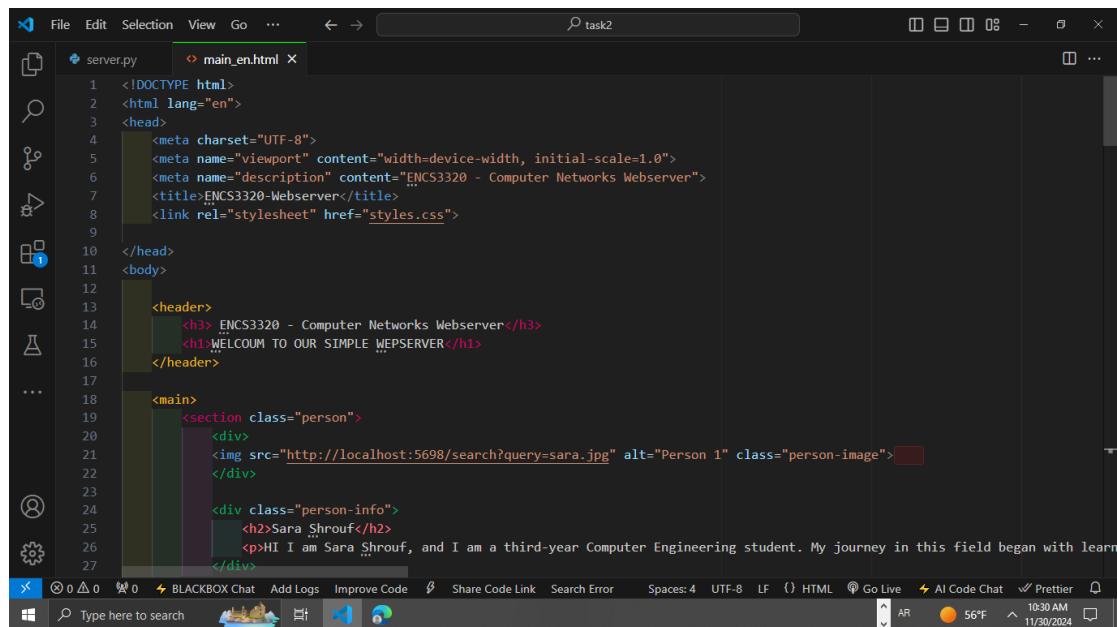
Supporting Material page in Arabic:

Figure 12: Supporting Material page in Arabic



HTML and SCC CODE :

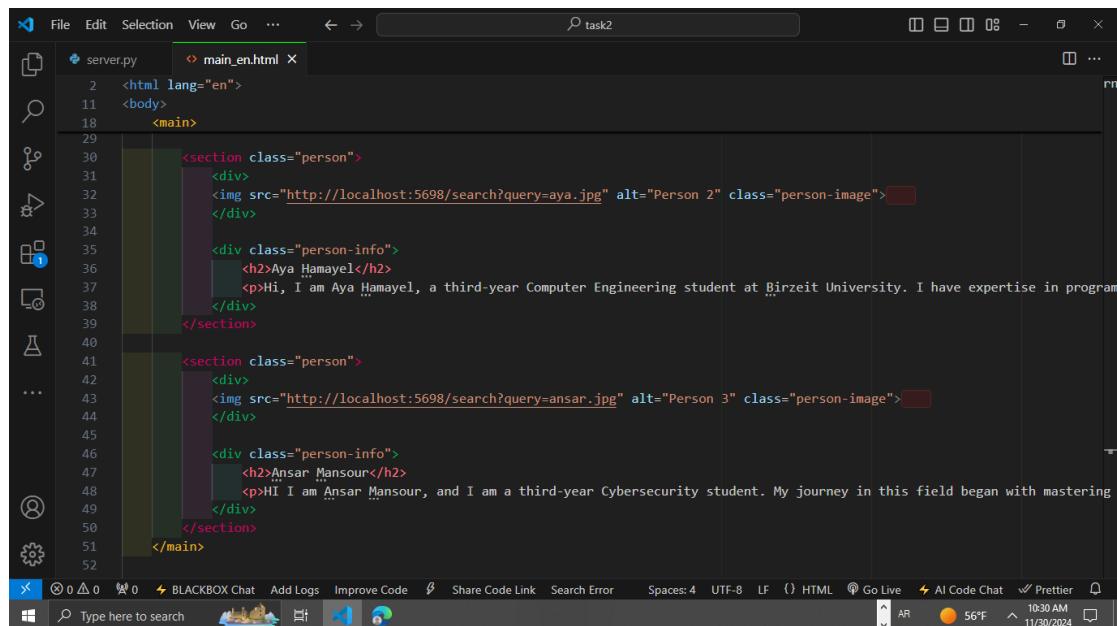
Main in Arabic:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="ENCS3320 - Computer Networks Webserver">
    <title>ENCS3320-Webserver</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h3> ENCS3320 - Computer Networks Webserver</h3>
        <h1>WELCOME TO OUR SIMPLE WEB SERVER</h1>
    </header>
    <main>
        <section class="person">
            <div>
                
            </div>
            <div class="person-info">
                <h2>Sara Shrouf</h2>
                <p>Hi I am Sara Shrouf, and I am a third-year Computer Engineering student. My journey in this field began with learning HTML and CSS, and I have since expanded my knowledge to include Java and Python. I am currently working on a project to build a web-based application for managing student records.</p>
            </div>
        </section>
        <section class="person">
            <div>
                
            </div>
            <div class="person-info">
                <h2>Aya Hamayel</h2>
                <p>Hi, I am Aya Hamayel, a third-year Computer Engineering student at Birzeit University. I have expertise in programming languages such as C, C++, and Java. I am currently working on a project to build a mobile application for managing inventory in a retail store.</p>
            </div>
        </section>
        <section class="person">
            <div>
                
            </div>
            <div class="person-info">
                <h2>Ansar Mansour</h2>
                <p>Hi I am Ansar Mansour, and I am a third-year Cybersecurity student. My journey in this field began with mastering network protocols and security concepts. I am currently working on a project to build a web-based application for managing user authentication and authorization.</p>
            </div>
        </section>
    </main>
</body>

```

Figure 29: Main in Arabic



```
<html lang="en">
<body>
    <main>
        <section class="person">
            <div>
                
            </div>
            <div class="person-info">
                <h2>Sara Shrouf</h2>
                <p>Hi I am Sara Shrouf, and I am a third-year Computer Engineering student. My journey in this field began with learning HTML and CSS, and I have since expanded my knowledge to include Java and Python. I am currently working on a project to build a web-based application for managing student records.</p>
            </div>
        </section>
        <section class="person">
            <div>
                
            </div>
            <div class="person-info">
                <h2>Aya Hamayel</h2>
                <p>Hi, I am Aya Hamayel, a third-year Computer Engineering student at Birzeit University. I have expertise in programming languages such as C, C++, and Java. I am currently working on a project to build a mobile application for managing inventory in a retail store.</p>
            </div>
        </section>
        <section class="person">
            <div>
                
            </div>
            <div class="person-info">
                <h2>Ansar Mansour</h2>
                <p>Hi I am Ansar Mansour, and I am a third-year Cybersecurity student. My journey in this field began with mastering network protocols and security concepts. I am currently working on a project to build a web-based application for managing user authentication and authorization.</p>
            </div>
        </section>
    </main>
</body>

```

```

File Edit Selection View Go ... ⏪ ⏩ task2
server.py main_en.html
2 <html lang="en">
11 <body>
56
57     <div class="subject">
58         <h2>We will discuss two subjects of networking</h2>
59     <div class="s1">
60
61         <h3>1. Internet Security: What is it, and How Can You Protect Yourself Online?</h3>
62         
63         <p>Internet security refers to the practice of protecting your data and online transactions from various threats while you're sur...
64         <p>With the increasing reliance on the internet for both personal and professional activities, understanding the importance of in...
65
66         <h4>Common Internet Security Threats</h4>
67         <p>Some of the most common internet security threats include:</p>
68         <ul>
69             <li><strong>Hacking:</strong> Unauthorized users gain access to computer systems, email accounts, or websites. Hackers may ex...
70             <li><strong>Viruses and Malware:</strong> Malicious software designed to harm or exploit any device, software, or network. Vi...
71             <li><strong>Identity Theft:</strong> Cybercriminals steal personal information, including social security numbers, credit car...
72             <li><strong>Phishing:</strong> Fake emails or websites that mimic legitimate organizations in order to trick you into revealin...
73             <li><strong>Ransomware:</strong> A form of malware that encrypts a victim's data and demands payment (usually in cryptocurren...
74             <li><strong>Distributed Denial-of-Service (DDoS) Attacks:</strong> Attackers flood a website or network with massive traffic...
75
76         <h4>How to Protect Yourself Online?</h4>
77         <p>While no system can be 100% secure, there are numerous steps you can take to significantly reduce your risk of falling victim...
78         <ul>
79             <li><strong>Use Strong Passwords:</strong> Avoid using easily guessable passwords. Use a combination of upper and lower case...
80             <li><strong>Enable Two-Factor Authentication (2FA):</strong> This adds an extra layer of security by requiring two forms of i...
81             <li><strong>Keep Software Updated:</strong> Ensure your operating system, browsers, and antivirus software are up to date. Se...
82             <li><strong>Install Anti-virus and Anti-malware Software:</strong> Use reputable security software to detect and block malw...
83             <li><strong>Be Cautious of Suspicious Links and Emails:</strong> Don't click on links or open attachments from unknown or untr...
84             <li><strong>Backup Your Data Regularly:</strong> In the event of a cyberattack, especially ransomware, having backups of your...
85
86         <p>By adopting these practices and staying vigilant, you can significantly reduce the risk of becoming a target of cybercrimes and...
87
88     </div>
89
90
91     <div class="s2">
92         <h3>2. The Domain Name System (DNS)</h3>
93         
94         <p>The Domain Name System (DNS) is one of the most critical components of the internet's infrastructure. It functions as the "phon...
95
96         <h4>How Does DNS Work?</h4>

```

```

File Edit Selection View Go ... ⏪ ⏩ task2
server.py main_en.html
2 <html lang="en">
11 <body>
57   <div class="subject">
90
91     <div class="s2">
92       <h3>2. The Domain Name System (DNS)</h3>
93       
94       <p>The Domain Name System (DNS) is one of the most critical components of the internet's infrastructure. It functions as the "phone book" of the internet, translating human-readable domain names into machine-readable IP addresses. This allows us to access websites like www.google.com instead of having to remember complex numerical IP addresses. DNS also provides a distributed system for managing domain names across many servers.
95
96       <h4>How Does DNS Work?</h4>
97       <p>When you enter a domain name into your browser, such as <strong>www.google.com</strong>, your computer doesn't automatically know the IP address. Instead, it sends a request to a DNS resolver (typically provided by your ISP). The resolver checks its cache to see if it already knows the IP address. If it does, it immediately returns it. If not, it queries other DNS servers, starting with root servers. Once the correct IP address is found, it is sent back to your browser, which then uses it to establish a connection.
98
99
100      <h4>Why Is DNS Important?</h4>
101      <p>Without DNS, users would have to remember complex numerical IP addresses instead of easy-to-recall domain names. DNS also provides a distributed system for managing domain names across many servers.
102
103      <h4>Common DNS Threats</h4>
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134

```

File Edit Selection View Go ... ⏪ ⏩ task2

server.py main_en.html

2 <html lang="en">

11 <body>

57 <div class="subject">

90

91 <div class="s2">

92 <h3>2. The Domain Name System (DNS)</h3>

93

94 <p>The Domain Name System (DNS) is one of the most critical components of the internet's infrastructure. It functions as the "phone book" of the internet, translating human-readable domain names into machine-readable IP addresses. This allows us to access websites like www.google.com instead of having to remember complex numerical IP addresses. DNS also provides a distributed system for managing domain names across many servers.

95

96 <h4>How Does DNS Work?</h4>

97 <p>When you enter a domain name into your browser, such as www.google.com, your computer doesn't automatically know the IP address. Instead, it sends a request to a DNS resolver (typically provided by your ISP). The resolver checks its cache to see if it already knows the IP address. If it does, it immediately returns it. If not, it queries other DNS servers, starting with root servers. Once the correct IP address is found, it is sent back to your browser, which then uses it to establish a connection.

98

99

100 <h4>Why Is DNS Important?</h4>

101 <p>Without DNS, users would have to remember complex numerical IP addresses instead of easy-to-recall domain names. DNS also provides a distributed system for managing domain names across many servers.

102

103 <h4>Common DNS Threats</h4>

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

```
File Edit Selection View Go ... < → task2
```

```
server.py main_en.html
```

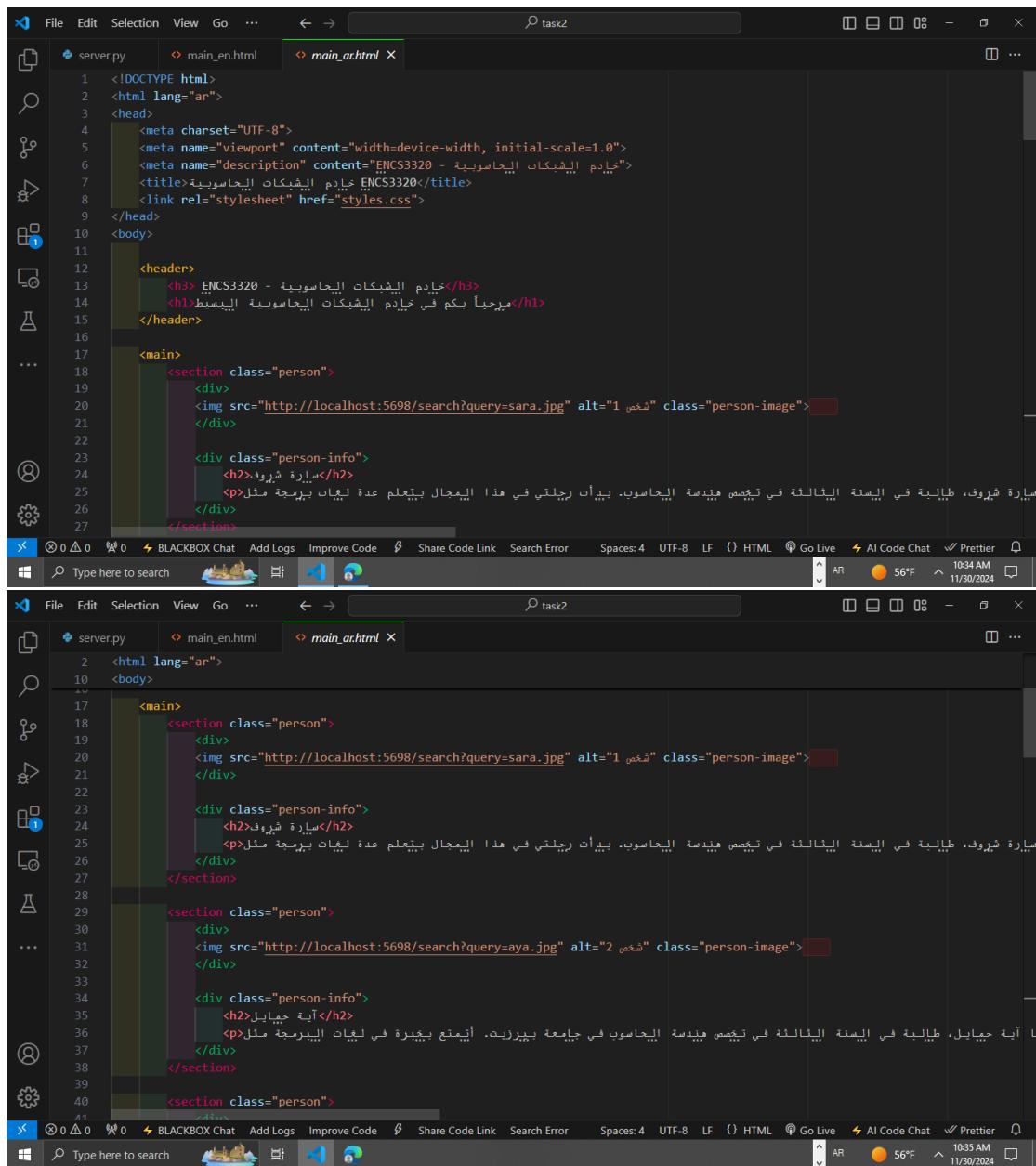
```
2 <html lang="en">
11 <body>
57   <div class="subject">
133
134   </div>
135
136
137
138
139 </body>
140 <footer>
141   <section>
142     <h2>wep pages</h2>
143     <ul>
144       <li><a href="main_ar.html">Main in Arabic</a></li>
145       <li><a href="supporting_material_ar.html">supporting_material_arabic</a></li>
146       <li><a href="supporting_material_en.html">supporting_material_english</a></li>
147     </ul>
148   </section>
149   <section>
150     <h2>External Resources</h2>
151     <ul>
152       <li>
153         <a href="https://gaia.cs.umass.edu/kurose_ross/index.php" target="_blank">
154           Textbook Website
155         </a>
156       </li>
157     </ul>
158   </section>
159 </Footer>
160 </html>
```

```
File Edit Selection View Go ... < → task2
```

```
server.py main_en.html
```

```
2 <html lang="en">
140 <Footer>
149   <section>
151     <ul>
152       <li>
153         <a href="#">
154           </a>
155       </li>
156     <li>
157       <a href="https://ritaj.birzeit.edu/" target="_blank">
158         Ritaj Website
159       </a>
160     </li>
161   </ul>
162 </section>
163 </Footer>
164 </html>
165
166
167
```

//main In Arabic



```
File Edit Selection View Go ... ← → task2
server.py main_en.html main_ar.html
1 <!DOCTYPE html>
2 <html lang="ar">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta name="description" content="ENCS3320 - خادم الشبكات الحاسوبية" data-bbox="147 140 849 180">
7   <title>خادم الشبكات الحاسوبية ENCS3320</title>
8   <link rel="stylesheet" href="styles.css">
9 </head>
10 <body>
11
12 <header>
13   <h3>خادم الشبكات الحاسوبية - ENCS3320</h3>
14   <h1>مرحباً بكم في خادم الشبكات الحاسوبية البنسيط</h1>
15 </header>
16
17 <main>
18   <section class="person">
19     <div>
20       
21     </div>
22
23     <div class="person-info">
24       <h2>سارة شلوف</h2>
25       <p>أيام الالتحاق: ٢٠٢٤-٢٠٢٥</p>
26     </div>
27   </section>
28
29   <section class="person">
30     <div>
31       
32     </div>
33
34     <div class="person-info">
35       <h2>آية حمائل</h2>
36       <p>أيام الالتحاق: ٢٠٢٤-٢٠٢٥</p>
37     </div>
38   </section>
39
40   <section class="person">
41     <div>
```

سارة شلوف، طالبة في السنة الثالثة في تخصص هندسة الحاسوب. بدأت رحلتي في هذا المجال بتعلم عدة لغات برمجة مثل Python.

آية حمائل، طالبة في السنة الثالثة في تخصص هندسة الحاسوب في جامعة بنزرت. أتمتع بقدرة في لغات البرمجة مثل Python.

اللقطة الأولى توضح الكود المكتوب في ملف main_arhtml، حيث يحتوي على جزء من HTML و CSS.

```

<html lang="ar">
  <body>
    <main>
      <section class="person">
        <div>
          
        </div>
        <div class="person-info">
          <h2>أليمار ميلمور</h2>
          <p>المجال يتعلم المواضيع الأساسية مثل أمان الشبكات، والتشفير، والاختراق الأخلاقي، إلى جانب تعلم لغات البرمجة مثل المستخدمين يمكنهم الاختراق في أنظمة مثل البنوك عبر الإنترنت، والتسوق، والتفاعل غير وسائل التواصل الاجتماعي بأمان</p>
        </div>
      </section>
    </main>
    <div class="subject">
      <h2>متناقض موضوع عن في الشبكات</h2>
      <div class="s1">
        <h3>أمان الإنترنت: ما هو وكيف يمكن حماية نفسك غير الإنترنت؟</h3>
        
        <p>المستخدمين يمكنهم الاختراق في أنظمة مثل البنوك عبر الإنترنت، والتسوق، والتفاعل غير وسائل التواصل الاجتماعي بأمان</p>
        <h4>التهديدات الشائعة لأمان الإنترنت</h4>
        <p>بعض التهديدات الشائعة لأمان الإنترنت تشمل</p>
        <ul>
          <li><strong>البرمجيات الفدية</strong></li>
          <li><strong>القرصنة</strong></li>
          <li><strong>سرقة الهوية</strong></li>
          <li><strong>بيانات الدخول</strong></li>
          <li><strong>التجسس</strong></li>
          <li><strong>البرمجة المورعة</strong></li>
        </ul>
      </div>
    </div>

```

اللقطة الثانية توضح الكود المكتوب في ملف main_en.html، حيث يحتوي على جزء من HTML و CSS.

```

<html lang="ar">
  <body>
    <div class="subject">
      <h2>متناقض موضوع عن في الشبكات</h2>
      <div class="s1">
        <h3>أمان الإنترنت: ما هو وكيف يمكن حماية نفسك غير الإنترنت؟</h3>
        
        <p>المستخدمين يمكنهم الاختراق في أنظمة مثل البنوك عبر الإنترنت، والتسوق، والتفاعل غير وسائل التواصل الاجتماعي بأمان</p>
        <h4>التهديدات الشائعة لأمان الإنترنت</h4>
        <p>بعض التهديدات الشائعة لأمان الإنترنت تشمل</p>
        <ul>
          <li><strong>البرمجيات الفدية</strong></li>
          <li><strong>القرصنة</strong></li>
          <li><strong>سرقة الهوية</strong></li>
          <li><strong>بيانات الدخول</strong></li>
          <li><strong>التجسس</strong></li>
          <li><strong>البرمجة المورعة</strong></li>
        </ul>
      </div>
    </div>

```

اللقطة الثالثة توضح الكود المكتوب في ملف main_en.html، حيث يحتوي على جزء من HTML و CSS.

```

<html lang="ar">
  <body>
    <div class="subject">
      <h2>متناقض موضوع عن في الشبكات</h2>
      <div class="s1">
        <h3>أمان الإنترنت: ما هو وكيف يمكن حماية نفسك غير الإنترنت؟</h3>
        
        <p>المستخدمين يمكنهم الاختراق في أنظمة مثل البنوك عبر الإنترنت، والتسوق، والتفاعل غير وسائل التواصل الاجتماعي بأمان</p>
        <h4>التهديدات الشائعة لأمان الإنترنت</h4>
        <p>بعض التهديدات الشائعة لأمان الإنترنت تشمل</p>
        <ul>
          <li><strong>البرمجيات الفدية</strong></li>
          <li><strong>القرصنة</strong></li>
          <li><strong>سرقة الهوية</strong></li>
          <li><strong>بيانات الدخول</strong></li>
          <li><strong>التجسس</strong></li>
          <li><strong>البرمجة المورعة</strong></li>
        </ul>
      </div>
    </div>

```

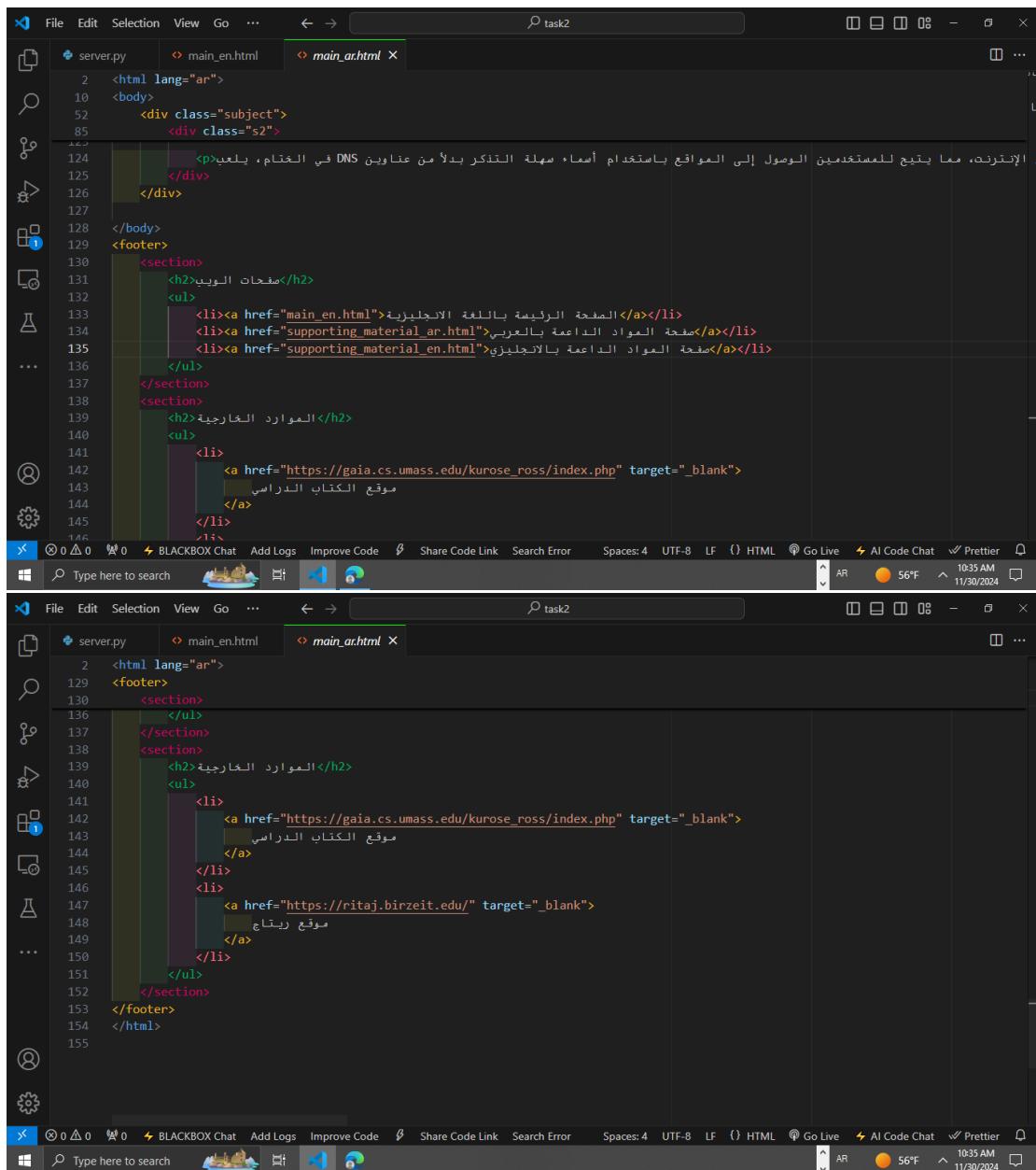
The screenshot shows a Microsoft Edge browser window with a Python code editor and a terminal at the bottom.

Code Editor:

```
File Edit Selection View Go ... ← → task2
server.py main_en.html main_ar.html
2 <html lang="ar">
3   <body>
4     <div class="subject">
5       <div class="s2">
6
7         <h4>مهماً DNS لماداً يعتبر</h4>
8           <p>أسهل للمستخدمين من خلال استخدام أسماء النطاقات المهمة. كما يوفر IP كان المستخدمون سيظلون إلى تذكر عنوانين بدلاً من الأسماء.</p>
9           <ul>
10             <li><strong>الراحة</strong></li>
11             <li><strong>النفع</strong></li>
12             <li><strong>المرونة</strong></li>
13           </ul>
14
15         <h4>DNS التهديدات الشائعة لـ DNS</h4>
16           <p>تشمل DNS إلا أنه ليس مخصوصاً ضد الهجمات. بعض التهديدات الشائعة لـ DNS، على الرغم من أهميتها، هي:</p>
17           <ul>
18             <li><strong>التصيد غير DNS</strong></li>
19             <li><strong>مجمات تعزز DNS</strong></li>
20             <li><strong>تفقق DNS</strong></li>
21           </ul>
22
23         <h4>كيفية تامين DNS</h4>
24           <p>ومنع الهجمات الشائعة، يمكن تنفيذ التدابير الوقائية التالية لحماية DNS:</p>
25           <ul>
26             <li><strong>DNSSEC</strong></li>
27             <li><strong>إضافة خوادم DNS ثانوية</strong></li>
28             <li><strong>التحقق من خصوصية خوادم DNS</strong></li>
29           </ul>
30
31         <h4>الخطوات الموصى بها</h4>
32           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
33           <ol>
34             <li>تحديث البيانات المهمة</li>
35             <li>استخدام خوادم DNS ثانوية</li>
36             <li>تحقيق التكامل بين خدمات DNS</li>
37           </ol>
38
39         <h4>الخطوات الموصى بها</h4>
40           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
41           <ol>
42             <li>تحديث البيانات المهمة</li>
43             <li>استخدام خوادم DNS ثانوية</li>
44             <li>تحقيق التكامل بين خدمات DNS</li>
45           </ol>
46
47         <h4>الخطوات الموصى بها</h4>
48           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
49           <ol>
50             <li>تحديث البيانات المهمة</li>
51             <li>استخدام خوادم DNS ثانوية</li>
52             <li>تحقيق التكامل بين خدمات DNS</li>
53           </ol>
54
55         <h4>الخطوات الموصى بها</h4>
56           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
57           <ol>
58             <li>تحديث البيانات المهمة</li>
59             <li>استخدام خوادم DNS ثانوية</li>
60             <li>تحقيق التكامل بين خدمات DNS</li>
61           </ol>
62
63         <h4>الخطوات الموصى بها</h4>
64           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
65           <ol>
66             <li>تحديث البيانات المهمة</li>
67             <li>استخدام خوادم DNS ثانوية</li>
68             <li>تحقيق التكامل بين خدمات DNS</li>
69           </ol>
70
71         <h4>الخطوات الموصى بها</h4>
72           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
73           <ol>
74             <li>تحديث البيانات المهمة</li>
75             <li>استخدام خوادم DNS ثانوية</li>
76             <li>تحقيق التكامل بين خدمات DNS</li>
77           </ol>
78
79         <h4>الخطوات الموصى بها</h4>
80           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
81           <ol>
82             <li>تحديث البيانات المهمة</li>
83             <li>استخدام خوادم DNS ثانوية</li>
84             <li>تحقيق التكامل بين خدمات DNS</li>
85           </ol>
86
87         <h4>الخطوات الموصى بها</h4>
88           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
89           <ol>
90             <li>تحديث البيانات المهمة</li>
91             <li>استخدام خوادم DNS ثانوية</li>
92             <li>تحقيق التكامل بين خدمات DNS</li>
93           </ol>
94
95         <h4>الخطوات الموصى بها</h4>
96           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
97           <ol>
98             <li>تحديث البيانات المهمة</li>
99             <li>استخدام خوادم DNS ثانوية</li>
100            <li>تحقيق التكامل بين خدمات DNS</li>
101          </ol>
102
103        <h4>الخطوات الموصى بها</h4>
104           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
105           <ol>
106             <li>تحديث البيانات المهمة</li>
107             <li>استخدام خوادم DNS ثانوية</li>
108             <li>تحقيق التكامل بين خدمات DNS</li>
109           </ol>
110
111         <h4>الخطوات الموصى بها</h4>
112           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
113           <ol>
114             <li>تحديث البيانات المهمة</li>
115             <li>استخدام خوادم DNS ثانوية</li>
116             <li>تحقيق التكامل بين خدمات DNS</li>
117           </ol>
118
119         <h4>الخطوات الموصى بها</h4>
120           <p>للحفاظ على الموثوقية، اتبع الخطوات التالية:</p>
```

Terminal:

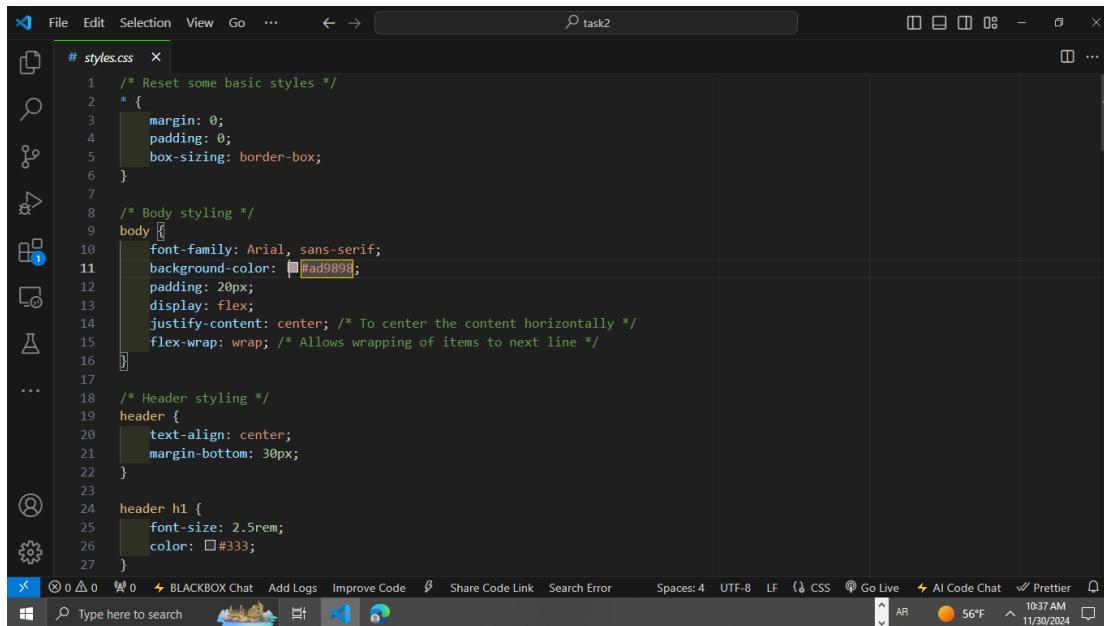
```
X 0 △ 0 ⚡ BLACKBOX Chat Add Logs Improve Code ⚡ Share Code Link Search Error Spaces: 4 UTF-8 LF () HTML ⚡ Go Live ⚡ AI Code Chat ⚡ Prettier
Windows Type here to search AR 56°F 10:35 AM 11/30/2024
```



الإنترنت، مما يتيح للمستخدمين الوصول إلى المواقع باستخدام أسماء سهلة التذكر بدلاً من عنوان DNS في العنوان، بلغة

```
File Edit Selection View Go ... ← → task2
server.py main_en.html main_ar.html
2 <html lang="ar">
10 <body>
124 <div class="subject">
125 <div class="s2">
126 </div>
127 </body>
129 <footer>
130 <section>
131 <h2>صفحات الويب</h2>
132 <ul>
133 <li><a href="main_en.html">الصفحة الرئيسية باللغة الإنجليزية</a></li>
134 <li><a href="supporting_material_ar.html">صفحة المواد الداعمة بالعربية</a></li>
135 <li><a href="supporting_material_en.html">صفحة المواد الداعمة بالإنجليزية</a></li>
136 </ul>
137 </section>
138 <section>
139 <h2>المصادر الخارجية</h2>
140 <ul>
141 <li><a href="https://gaia.cs.umass.edu/kurose_ross/index.php" target="_blank">موقع الكتاب الدراسي</a></li>
142 </ul>
143 </section>
144 <ul>
145 <li><a href="https://ritaj.birzeit.edu/" target="_blank">موقع ريتاج</a></li>
146 </ul>
147 </section>
148 </footer>
149 </html>
150
151
152
153
154
155
```

Css _style code :

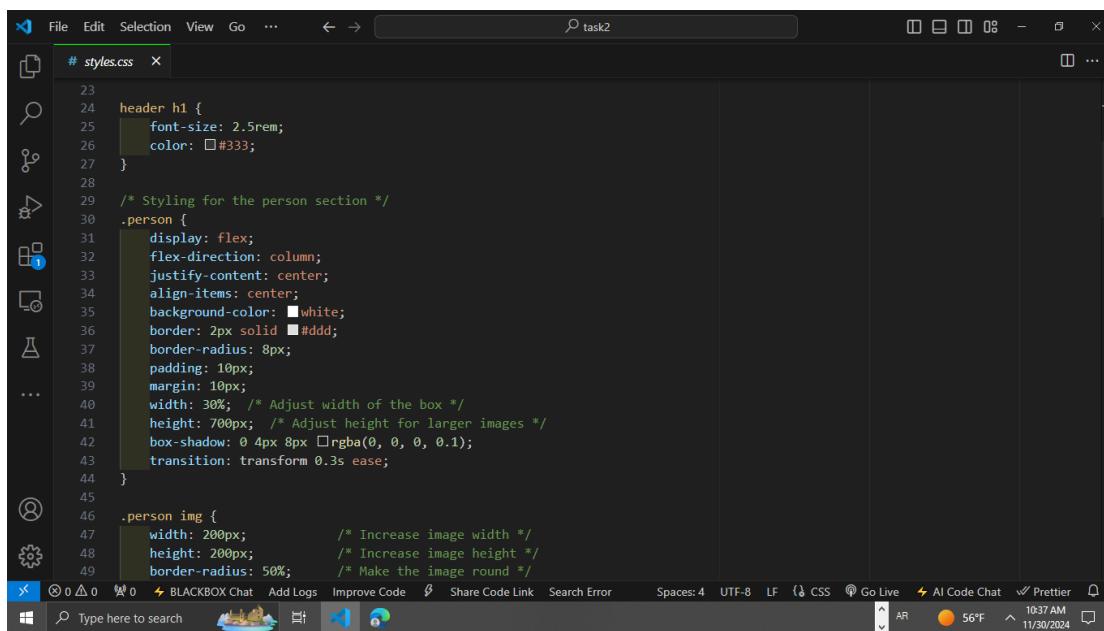


A screenshot of a code editor window titled "task2". The file being edited is "styles.css". The code is as follows:

```
# styles.css x
1  /* Reset some basic styles */
2  * {
3      margin: 0;
4      padding: 0;
5      box-sizing: border-box;
6  }
7
8  /* Body styling */
9 body {
10    font-family: Arial, sans-serif;
11    background-color: #ad9988;
12    padding: 20px;
13    display: flex;
14    justify-content: center; /* To center the content horizontally */
15    flex-wrap: wrap; /* Allows wrapping of items to next line */
16 }
17
18 /* Header styling */
19 header {
20    text-align: center;
21    margin-bottom: 30px;
22 }
23
24 header h1 {
25    font-size: 2.5rem;
26    color: #333;
27 }
```

The code editor has a dark theme with syntax highlighting. The "body" selector is currently selected. The status bar at the bottom shows "Type here to search" and various system icons.

Figure 13:Css_style cod



A screenshot of a code editor window titled "task2". The file being edited is "styles.css". The code is as follows:

```
23
24 header h1 {
25     font-size: 2.5rem;
26     color: #333;
27 }
28
29 /* Styling for the person section */
30 .person {
31     display: flex;
32     flex-direction: column;
33     justify-content: center;
34     align-items: center;
35     background-color: white;
36     border: 2px solid #ddd;
37     border-radius: 8px;
38     padding: 10px;
39     margin: 10px;
40     width: 30%; /* Adjust width of the box */
41     height: 700px; /* Adjust height for larger images */
42     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
43     transition: transform 0.3s ease;
44 }
45
46 .person img {
47     width: 200px; /* Increase image width */
48     height: 200px; /* Increase image height */
49     border-radius: 50%; /* Make the image round */
```

The code editor has a dark theme with syntax highlighting. The ".person" selector is currently selected. The status bar at the bottom shows "Type here to search" and various system icons.

The image shows three separate Microsoft Edge browser windows, each displaying a different part of a CSS file named '# styles.css'. The code is written in a dark-themed code editor.

Top Window: Person Section Styling

```

44 }
45
46 .person img {
47   width: 200px; /* Increase image width */
48   height: 200px; /* Increase image height */
49   border-radius: 50%; /* Make the image round */
50   border: 2px solid #ddd; /* Border around image */
51   object-fit: cover; /* Ensure image doesn't get distorted */
52   margin-bottom: 20px; /* Space between image and text */
53 }
54
55 .person-info {
56   text-align: center; /* Center align the text */
57 }
58
59 .person-info h2 {
60   font-size: 1.7rem; /* Increase font size for heading */
61 }
62
63 .person-info p {
64   font-size: 1.2rem; /* Increase font size for paragraph */
65 }
66
67 /* Hover effect for the person box */
68 .person:hover {
69   transform: scale(1.05); /* Slight zoom effect on hover */
70 }

```

Middle Window: Main Container Styling

```

59 .person-info h2 {
60   font-size: 1.7rem; /* Increase font size for heading */
61 }
62
63 .person-info p {
64   font-size: 1.2rem; /* Increase font size for paragraph */
65 }
66
67 /* Hover effect for the person box */
68 .person:hover {
69   transform: scale(1.05); /* Slight zoom effect on hover */
70 }
71
72 /* Container to hold all the person sections */
73 main {
74   display: flex;
75   flex-wrap: wrap; /* Allow wrapping of person boxes */
76   justify-content: center; /* Center the boxes horizontally */
77   gap: 20px; /* Space between person boxes */
78 }
79
80 /* Styling for the subject section */
81 .subject {
82   max-width: 1200px;
83   margin: 0 auto;
84   padding: 20px;
85 }

```

Bottom Window: Subject Section Styling

```

86
87 /* Heading for subjects */
88 h2 {
89   text-align: center;
90   font-size: 2.5rem;
91   margin-bottom: 40px;
92   color: #333;
93 }
94
95 /* Styling for each individual section */
96 .s1, .s2 {
97   background-color: #fff;
98   border-radius: 8px;
99   padding: 20px;
100  margin-bottom: 30px;
101  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
102 }
103
104 .s1 h3, .s2 h3 {
105   font-size: 2rem;
106   color: #2c3e50;
107   margin-bottom: 20px;
108 }
109
110 .s1 img, .s2 img {
111   max-width: 100%;
112   height: auto;
113 }

```

The image shows three vertically stacked Microsoft Edge browser windows, each displaying a different CSS file:

- Top Window:** Displays the content of `# styles.css`. The code includes styling for images, paragraphs, lists, and subheadings. Key snippets include:

```
108     }
109     .s1 img, .s2 img {
110         max-width: 100%;
111         height: auto;
112         border-radius: 8px;
113         margin-bottom: 20px;
114     }
115
116     /* Paragraph and list styling */
117     p {
118         font-size: 1rem;
119         color: #555;
120         margin-bottom: 20px;
121     }
122
123     ul {
124         list-style-type: disc;
125         margin-left: 20px;
126     }
127
128     ul li {
129         font-size: 1rem;
130         margin-bottom: 10px;
131     }
132
133     /* Subheading Styling */
134     h4 {
135         font-size: 1.5rem;
136         color: #34495e;
137         margin-top: 20px;
138         margin-bottom: 15px;
139     }
140
141     /* Section-specific spacing */
142     .s1 ul, .s2 ul {
143         margin-left: 30px;
144     }
145
146     .s1 ul li, .s2 ul li {
147         margin-bottom: 12px;
148     }
149
150
151
```

- Middle Window:** Displays the content of `# styles.css`, showing the same code as the top window.
- Bottom Window:** Displays the content of `# supp_styles.css`. The code defines general styles for the body and header, and specific styles for the h1 element. Key snippets include:

```
1     /* General Styles */
2     body {
3         font-family: Arial, sans-serif;
4         margin: 0;
5         padding: 0;
6         background-color: #f4f4f4;
7         color: #333;
8     }
9
10    h1 {
11        font-size: 2.5rem;
12        text-align: center;
13        color: #333;
14        margin-top: 40px;
15    }
16
17    /* Header Styles */
18    header {
19        background-color: #4CAF50;
20        padding: 20px;
21        color: white;
22        text-align: center;
23    }
24
25    header h1 {
26        margin: 0;
27        font-size: 2.5rem;
```

The image shows two side-by-side screenshots of the Microsoft Edge browser's developer tools, specifically the CSS panel. Both windows have the title '# supp_styles.css'.

Top Window (Lines 23-49):

```
23 }
24
25 header h1 {
26     margin: 0;
27     font-size: 2.5rem;
28 }
29
30 /* Section Styles */
31 section {
32     padding: 20px;
33     max-width: 800px;
34     margin: 40px auto;
35     background-color: #fff;
36     box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
37 }
38
39 section p {
40     font-size: 1.2rem;
41     line-height: 1.6;
42     color: #666;
43 }
44
45 form {
46     display: flex;
47     flex-direction: column;
48     margin-top: 20px;
49 }
```

Bottom Window (Lines 75-101):

```
75
76 button[type="submit"]:hover {
77     background-color: #45a049;
78 }
79
80 /* Footer Styles */
81 footer {
82     background-color: #1c7a1c;
83     color: #rgb(163, 139, 139);
84     text-align: center;
85     padding: 10px;
86     font-size: 1rem;
87 }
88
89 footer ul {
90     list-style: none;
91     padding: 0;
92 }
93
94 footer ul li {
95     margin: 10px 0;
96 }
97
98 footer ul li a {
99     color: #1d1212;
100    text-decoration: none;
101 }
```

supp_styles.css

```
51 label {  
52   font-size: 1rem;  
53   margin-bottom: 10px;  
54 }  
55  
56 input[type="text"] {  
57   padding: 10px;  
58   font-size: 1rem;  
59   margin-bottom: 20px;  
60   border: 1px solid #ccc;  
61   border-radius: 4px;  
62   width: 100%;  
63 }  
64  
65 button[type="submit"] {  
66   padding: 10px 20px;  
67   background-color: #4CAF50;  
68   color: white;  
69   font-size: 1rem;  
70   border: none;  
71   border-radius: 4px;  
72   cursor: pointer;  
73   width: 100%;  
74 }  
75  
76 button[type="submit"]:hover {  
77   background-color: #45a049;
```

supp_styles.css

```
87   
88  
89 footer ul {  
90   list-style: none;  
91   padding: 0;  
92 }  
93  
94 footer ul li {  
95   margin: 10px 0;  
96 }  
97  
98 footer ul li a {  
99   color: #1d1212;  
100  text-decoration: none;  
101 }  
102  
103 footer ul li a:hover {  
104  text-decoration: underline;  
105 }  
106  
107 /* Remove fixed position for footer */  
108 footer {  
109  position: relative;  
110 }  
111
```

Test our wep server in another device in same local network:

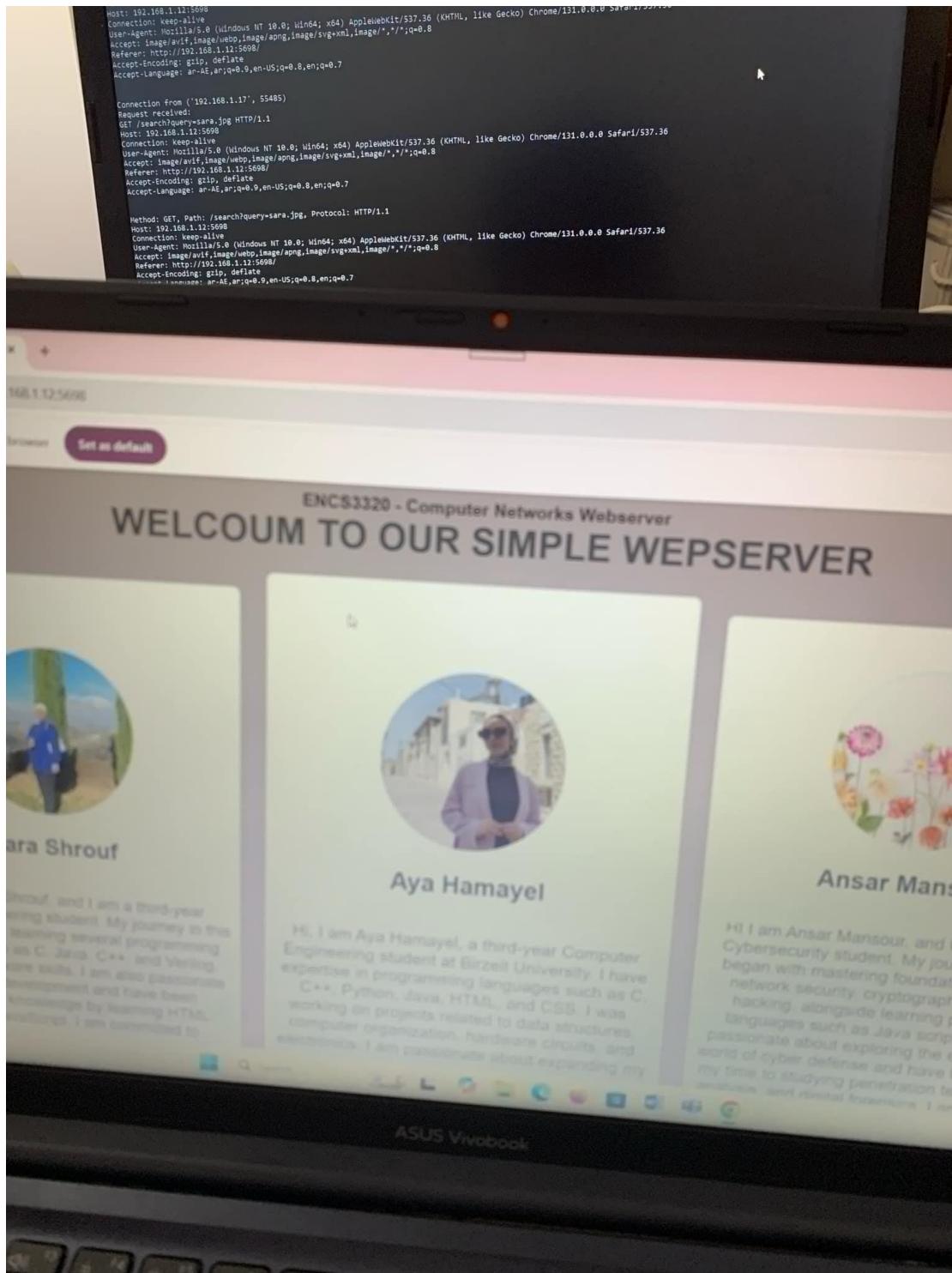


Figure 30: Test our wep server in another device

```
Command Prompt - py server.py
Connection from ('192.168.1.17', 55528)
Request received:
Connection from ('192.168.1.17', 55529)
Request received:
GET / HTTP/1.1
Host: 192.168.1.12:5698
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: ar-AE,ar;q=0.9,en-US;q=0.8,en;q=0.7

Method: GET, Path: /, Protocol: HTTP/1.1
Host: 192.168.1.12:5698
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: ar-AE,ar;q=0.9,en-US;q=0.8,en;q=0.7

Connection from ('192.168.1.17', 55546)
Request received:
GET / HTTP/1.1
Host: 192.168.1.12:5698
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: ar-AE,ar;q=0.9,en-US;q=0.8,en;q=0.7

Method: GET, Path: /, Protocol: HTTP/1.1
Host: 192.168.1.12:5698
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: ar-AE,ar;q=0.9,en-US;q=0.8,en;q=0.7
```

```
Command Prompt - py server.py
Request received:
GET /search?query=ansar.jpg HTTP/1.1
Host: 192.168.1.12:5698
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*/*;q=0.8
Referer: http://192.168.1.12:5698
Accept-Encoding: gzip, deflate
Accept-Language: ar-AE,ar;q=0.9,en-US;q=0.8,en;q=0.7

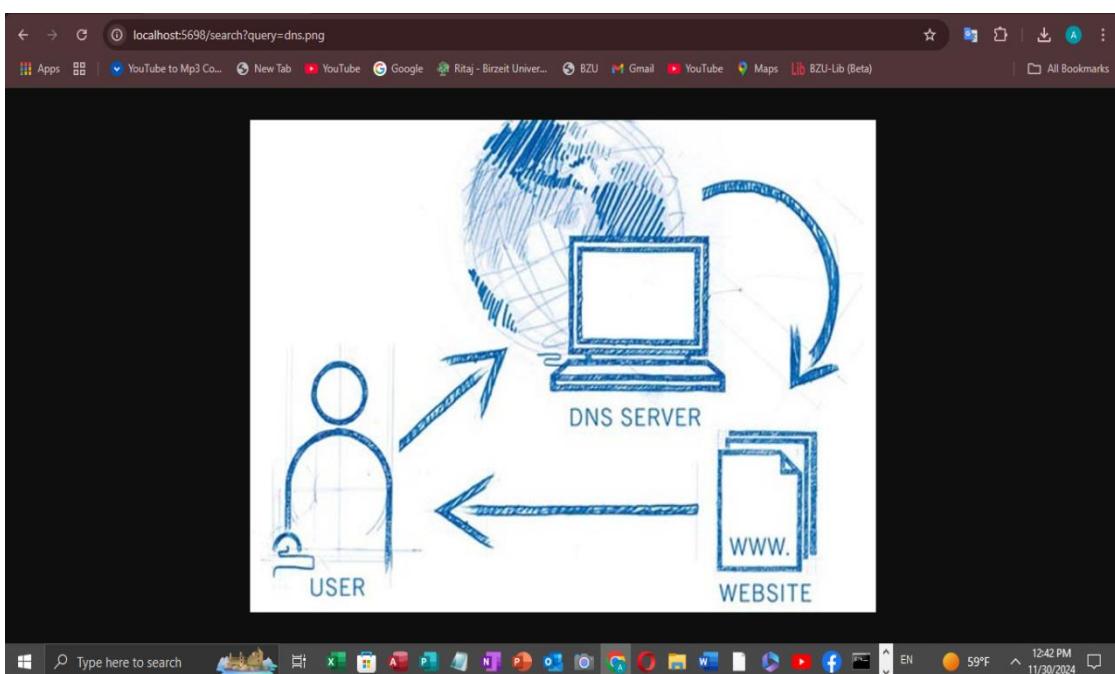
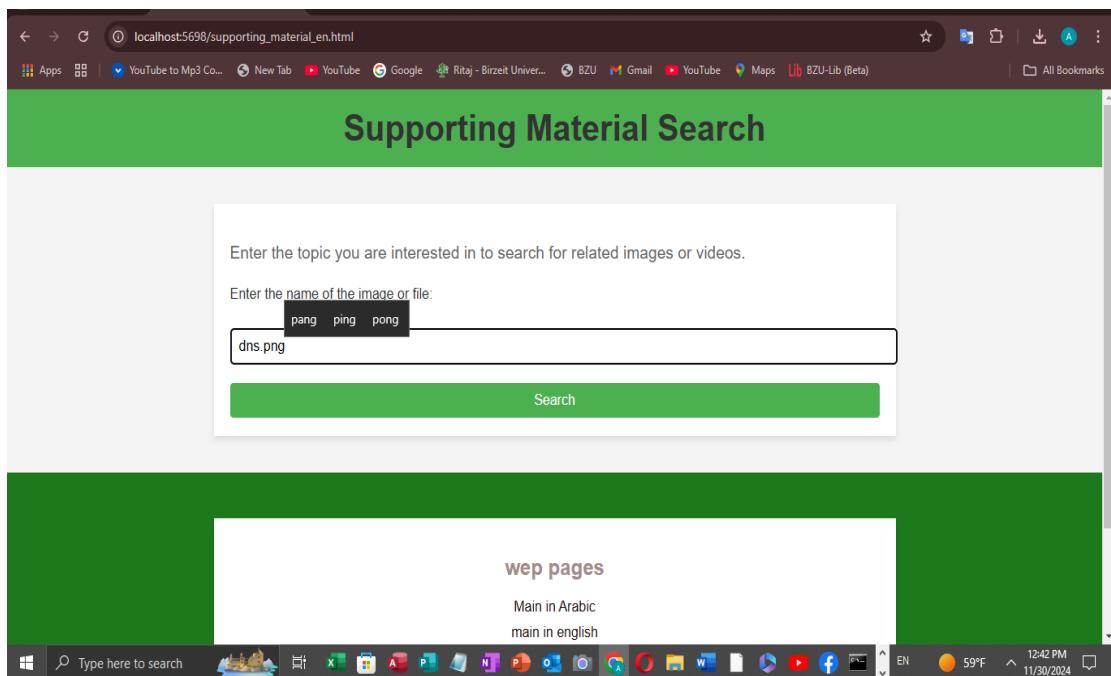
Method: GET, Path: /search?query=ansar.jpg, Protocol: HTTP/1.1
Host: 192.168.1.12:5698
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*/*;q=0.8
Referer: http://192.168.1.12:5698
Accept-Encoding: gzip, deflate
Accept-Language: ar-AE,ar;q=0.9,en-US;q=0.8,en;q=0.7

Connection from ('192.168.1.17', 55553)
Request received:
GET /favicon.ico HTTP/1.1
Host: 192.168.1.12:5698
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*/*;q=0.8
Referer: http://192.168.1.12:5698
Accept-Encoding: gzip, deflate
Accept-Language: ar-AE,ar;q=0.9,en-US;q=0.8,en;q=0.7

Method: GET, Path: /favicon.ico, Protocol: HTTP/1.1
Host: 192.168.1.12:5698
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*/*;q=0.8
Referer: http://192.168.1.12:5698
Accept-Encoding: gzip, deflate
Accept-Language: ar-AE,ar;q=0.9,en-US;q=0.8,en;q=0.7
```

Test supporting material photo that exist in server:

Figure 14: Test supporting mat photo that exist in server



```

[1] Command Prompt - py server.py
Request received:
GET /search?query=dns.png HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "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/131.0.0.0 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: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /search?query=dns.png, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "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/131.0.0.0 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: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 63472)
Request received:
Connection from ('127.0.0.1', 63473)
Request received:

```

Test fore material not exist:

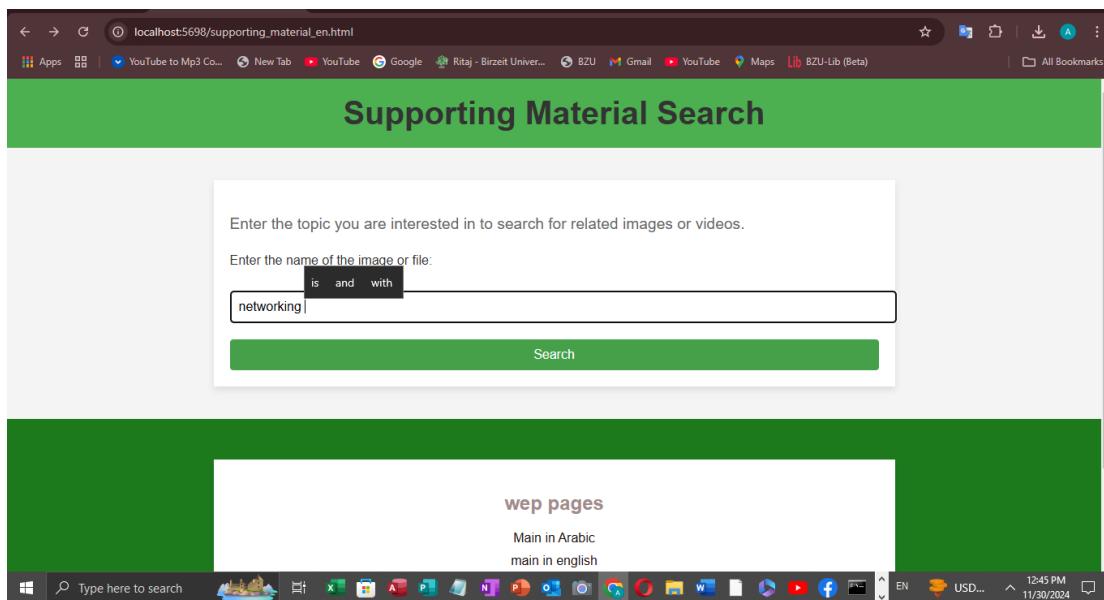
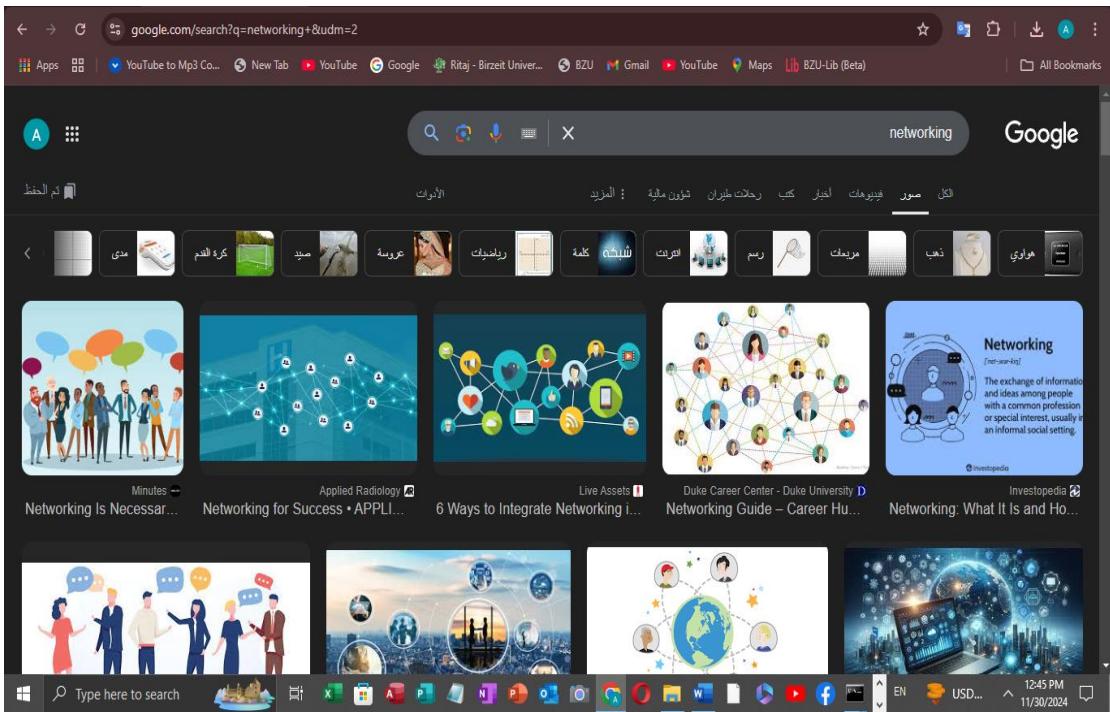


Figure 31: Test for material not exist



```

Command Prompt - py server.py
Request received:
GET /search?query=networking+ HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "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/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /search?query=networking+, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "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/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 63503)
Request received:
Connection from ('127.0.0.1', 63505)
Request received:

```

Test for video:

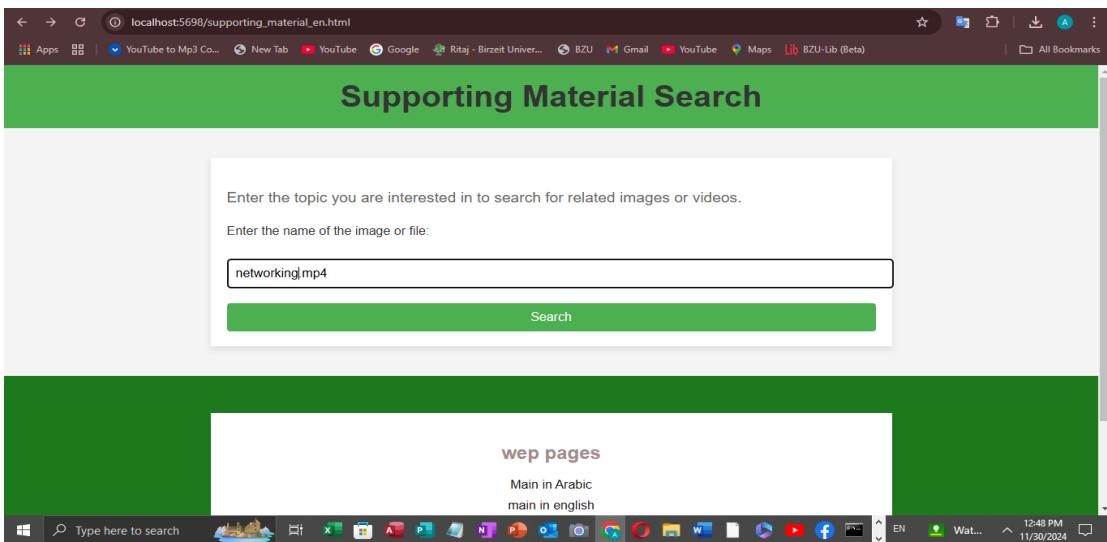
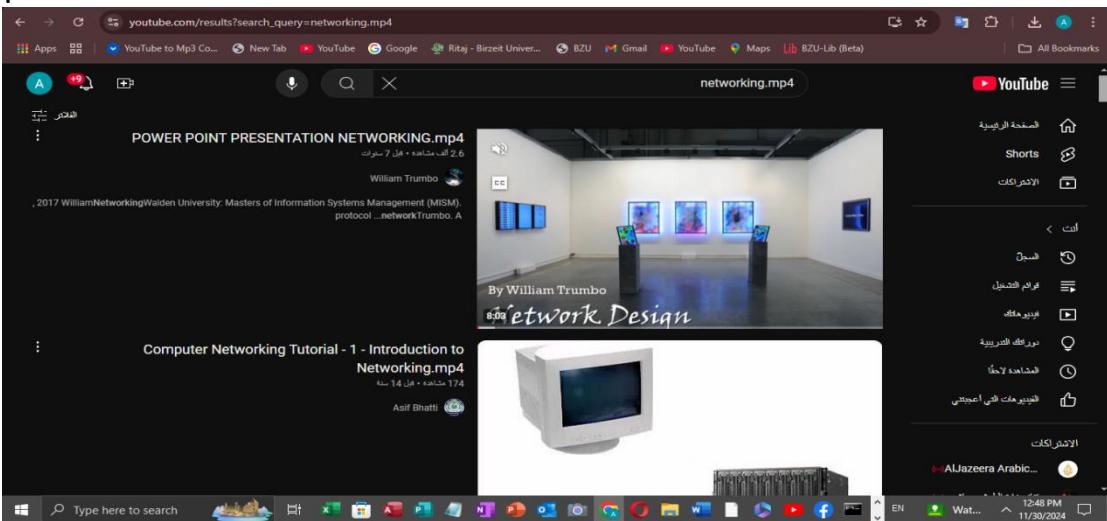


Figure 15: Test for video



```
 Command Prompt - py server.py
Request received:
GET /search?query=networking.mp4 HTTP/1.1
Host: localhost:5698
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Method: GET, Path: /search?query=networking.mp4, Protocol: HTTP/1.1
Host: localhost:5698
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "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/131.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-AU,en-GB;q=0.9,en-US;q=0.8,en;q=0.7,ar;q=0.6

Connection from ('127.0.0.1', 63541)
Request received:

Connection from ('127.0.0.1', 63544)
Request received:
```

❖ Task 3 – UDP Client-Server Trivia Game Using Socket Programming :

- In this task ,we implements a multiplayer trivia game using UDP socket programming, fostering an engaging competition where players answer questions to earn points. The server manages connections, broadcasts questions, and tracks scores, while clients interact by answering questions and receiving updates. The game proceeds in rounds, requiring at least two players, with the server pausing between rounds for preparation .

1. Server code :

```
C: > Users > u > Desktop > my python > server.py > ...
1  import socket
2  import threading
3  import time
4  import random
5  import sys
6
7  # Initialize server
8  server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
9  SERVER_IP = '172.19.51.160' #my server's IP
10 SERVER_PORT = 5689
11 server_socket.bind((SERVER_IP, SERVER_PORT))
12
13 print(f"Server is running on IP: {SERVER_IP}, Port: {SERVER_PORT}")
14
```

Figure 32: UDP server

```
C:\Users\u\Desktop\my python>py server.py
Server is running on IP: 172.19.51.160, Port: 5689
```

The UDP server is built in Fig ().we used The socket library to Creates a UDP server socket for communication using (SOCK_DGRAM) , server port identified to 5689 and server's IP to my device .and Binds the socket to the server's IP and port number to start listening for data. , a message identifying that the server is ready is receiv

```
15  # Data structures
16  Activeclients = {}
17  point_Struct = {}
18  question_Struct = [
19      ("What is the capital of France?", "Paris"),
20      ("What is 2 + 2?", "4"),
21      ("5 + 6?", "11"),
22      ("2 * 5?", "10"),
23      ("What is the chemical symbol for water?", "H2O"),
24      ("How many continents are there on Earth?", "7"),
25      ("What is the smallest prime number?", "2"),
26      ("In which year did World War II end?", "1945"),
27      ("What is the square root of 81?", "9"),
28      ("What is the longest river in the world?", "Amazon"),
29  ]
30  Selectedquestions_Struct = []
31  C_Struct = []
```

Figure 33 : Datastructures to implement

5 important Datastructures are required to easily implement the task.

A- For Initially Active Clients (**Activeclient**)

B- For Scores to each client addresses (**point_Struct**)

E- Database for Questions (**question_Struct**)

F- Database for N randomly Selected question with len (N) (**selectedqstion_Struct**)

G- For Clients Who Answered No matter For correctness (**c_struct**)

```
32 def main_clients():
33     while True:
34         data, addr = server_socket.recvfrom(4096)
35         client_message = data.decode().strip()
36
37         if client_message.lower() == "exit":
38             if addr in Activeclients:
39                 print(f"Client {Activeclients[addr]} ({addr}) has left the game.")
40                 del Activeclients[addr]
41                 del point_Struct[addr]
42                 server_socket.sendto("You have left the game.".encode(), addr)
43         elif addr not in Activeclients:
44             Activeclients[addr] = client_message
45             point_Struct[addr] = 0 # Initialize score
46             print(f"{client_message} ({addr}) joined the game.")
47             server_socket.sendto(f"Welcome {client_message}! Waiting for the game to start...".encode(), addr)
48
49     threading.Thread(target=main_clients, daemon=True).start()
50
51 print("Waiting for clients to join...")
```

Figure 34 : Handling Clients

In this method the server handling with incoming client connections and messages. And Receives data and client address from the socket.then Decodes the received message into a string . If the message sent from the client is an exit , they are removed from the active clients and their score data. if a new client joins, it adds them to activeclient and initializes their score in point _Struct

```
53 try:
54     # Main game loop
55     while True:
56         while len(Activeclients) < 2:
57             print("Waiting for at least 2 clients to start the game...")
58             time.sleep(5)
59             continue
60
61         # Select N random questions
62         N = 5
63         Selectedquestions_Struct = random.sample(question_Struct, N)
64         print("Round starting...")
65         for addr in Activeclients:
66             server_socket.sendto("Round starting in 1 minute! GET READY!".encode(), addr)
67             time.sleep(60)
68
69         # Ask each question
70         for i, (question, correct_answer) in enumerate(Selectedquestions_Struct, start=1):
71             # Broadcast the question
72             for addr in Activeclients:
73                 server_socket.sendto(f"Question {i}: {question}".encode(), addr)
74             print(f"Broadcasted Question {i}: {question}")
75
76             C_Struct = [] # Reset answered clients
77             start_time = time.time()
78
79             while time.time() - start_time < 90: # Wait for 90 seconds
```

Figure 35: gameloop

Each round begins only if there are at least 2 players active Clients and it each round include 5 question then start after 60 s, and Sends each question to all clients .

```

79     while time.time() - start_time < 90:
80         try:
81             server_socket.settimeout(1)
82             data, addr = server_socket.recvfrom(1024)
83             answer = data.decode().strip()
84
85             if addr in Activeclients and addr not in C_Struct: # Only process the first answer
86                 C_Struct.append(addr) # Mark client as having answered
87                 is_correct = answer.lower() == correct_answer.lower()
88
89                 if is_correct:
90                     time_taken = time.time() - start_time
91                     points = round(10 / time_taken, 2)
92                     point_Struct[addr] += points
93                     print(f"player {Activeclients[addr]} ({addr}) answered: {answer} - Correct! +{points:.2f} points")
94                 else:
95                     print(f"player {Activeclients[addr]} ({addr}) answered: {answer} - Incorrect")
96             else:
97                 print(f"Ignored subsequent answer from {Activeclients.get(addr, 'Unknown')} ({addr})")
98         except socket.timeout:
99             continue
100
101     # Broadcast the correct answer
102     for addr in Activeclients:
103         server_socket.sendto(f"Correct answer: {correct_answer}".encode(), addr)

```

Figure 36 :game code

Waits 90 seconds for client responses.and the server receive the first answer from each client only, If correct, calculates score based on response time .

```

105     leaderboard = sorted(point_Struct.items(), key=lambda x: x[1], reverse=True)
106     leaderboard_message = "totalscore:\n" + "\n".join([f"{Activeclients[addr]}: {score:.2f}" for addr, score in leaderboard])
107     for addr in Activeclients:
108         server_socket.sendto(leaderboard_message.encode(), addr)
109
110     leading_player = leaderboard[0]
111     for addr in Activeclients:
112         server_socket.sendto(f"Winning player: {Activeclients[leading_player[0]]}".encode(), addr)
113
114     print("Round ended!")
115     print(leaderboard_message)
116
117     print("Next round will start in 30 seconds...")
118     time.sleep(30)
119
120 except KeyboardInterrupt:
121     print("\nServer is shutting down...")
122     for addr in Activeclients:
123         server_socket.sendto("Server is shutting down. Goodbye!".encode(), addr)
124     sys.exit(0)
125

```

Figure 37 : round end code

Sorts and broadcasts the leaderboard after all questions.

2. Client code :

```
C: > Users > u > Desktop > my python > client.py > ...
1 import socket
2 import threading
3
4 DEFAULT_SERVER_IP = "172.19.51.160"
5 DEFAULT_SERVER_PORT = 5689
6 BUFFER_SIZE = 1024
7
8 |
9 def receive_messages(client_socket):
10     """Listen for messages from the server."""
11     while True:
12         try:
13             message = client_socket.recv(BUFFER_SIZE).decode()
14             print(message)
15         except Exception as e:
16             print(f"Disconnected from server: {e}")
17             break
18
19
20 def main():
21     print("Welcome to the Game!")
22     server_ip = input("Enter server IP : ").strip() or DEFAULT_SERVER_IP
23     server_port = input("Enter server port : ").strip()
24     server_port = int(server_port) if server_port.isdigit() else DEFAULT_SERVER_PORT
25
26     username = input("Enter your username: ").strip()
27     if not username:
28         print("Username cannot be empty. Exiting.")
29         return
30
31     client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
32
33     try:
34         client_socket.sendto(username.encode(), (server_ip, server_port))
35         print(f"Connected to server at {server_ip}:{server_port}\n")
36     except Exception as e:
37         print(f"Failed to connect: {e}")
38         return
39
40     threading.Thread(target=receive_messages, args=(client_socket,), daemon=True).start()
41
42     try:
43         while True:
44             answer = input("Your answer (or 'exit' to quit): ").strip()
45             if answer.lower() == 'exit':
46                 print("Exiting game.")
47                 break
48             client_socket.sendto(answer.encode(), (server_ip, server_port))
49     except KeyboardInterrupt:
50         print("\nExiting game.")
51     finally:
52         client_socket.close()
53
54
55 if __name__ == "__main__":
56     main()
```

Figure 38:client code

The UDB server is built in Fig (6). The socket library is used, the client socket is defined UDB using (SOCK_DGRAM), The client enters the IP address and port number of the server to communicate , and the client answerd the question then send it to the server If the client enter exit by send massage to server

3)Here is an example when running the game :

1. the server connected the game and waiting player :

The image shows four separate Command Prompt windows arranged in a 2x2 grid. The top-left window shows the command `python server.py` being run, with the output indicating the server is listening on IP 172.19.51.160, Port 5689, and waiting for clients to join. The top-right window shows the command `python client.py` being run, with the output showing the client connecting to the server at 172.19.51.160:5689. The bottom-left window is empty, and the bottom-right window is also empty.

Figure 39: test game1

2) the first client "sara " has joined by enter the ip and port number for server and the still wait at least another client to joind

The image shows four Command Prompt windows. The top-left window shows the command `python client.py` being run, with the output showing the client connecting to the server at 172.19.51.160:5689. The top-right window shows the command `python server.py` being run, with the output indicating the server is listening on IP 172.19.51.160, Port 5689, and waiting for at least 2 clients to start the game. The bottom-left window is empty, and the bottom-right window is also empty.

Figure 40: test game2

3) the 2 other client have joined 'aya ' &'ansar'
Then the game will be start after 60sec

```

Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\wcd\Desktop\my python
Welcome to the Game!
Enter server IP : 172.19.51.160
Enter server port : 5689
Enter your username: sara
Connected to server at 172.19.51.160:5689

Welcome sara! Waiting for the game to start...
Your answer (or 'exit' to quit): Round starting in 1 minute! GET READY!

Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\wcd\Desktop\my python
Welcome to the Game!
Enter server IP : 172.19.51.160
Enter server port : 5689
Enter your username: aya
Connected to server at 172.19.51.160:5689

Welcome aya! Waiting for the game to start...
Your answer (or 'exit' to quit): Round starting in 1 minute! GET READY!

```

Figure 41: game test3

- 4) the server broadcast the question for all client and calculate the score For those whose answer is correct, points will be increased for the quickest answer

```

C:\Users\wcd\Desktop\my python>py client.py
Welcome to the Game!
Enter server IP : 172.19.51.160
Enter server port : 5689
Enter your username: aya
Connected to server at 172.19.51.160:5689

Welcome aya! Waiting for the game to start...
Your answer (or 'exit' to quit): Round starting in 1 minute! GET READY!
Question 1: 2 * 5?
10
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit):

C:\Users\wcd\Desktop\my python>py client.py
Welcome to the Game!
Enter server IP : 172.19.51.160
Enter server port : 5689
Enter your username: sara
Connected to server at 172.19.51.160:5689

Welcome sara! Waiting for the game to start...
Your answer (or 'exit' to quit): Round starting in 1 minute! GET READY!
Question 1: 2 * 5?
10
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit):

data, addr = server_socket.recvfrom(4096)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
TimeoutError: timed out
playeraya (('172.19.51.160', 52918)) answered: 10 - Correct! +0.92 point
s
playeransar (('172.19.51.160', 57811)) answered: 10 - Correct! +0.40 points
player sara (('172.19.51.160', 54559)) answered: 11 - Incorrect
Broadcasted Question 2: What is the smallest prime number?
playersara (('172.19.51.160', 54559)) answered: 2 - Correct! +0.77 points
s
playeraya (('172.19.51.160', 52918)) answered: 2 - Correct! +0.46 points
playeransar (('172.19.51.160', 57811)) answered: 2 - Correct! +0.40 points
ts

```

Figure 42:game test4

- 5) if client send more on answer It is ignored for example in this pic

```

[aya] Enter your username: aya
Connected to server at 172.19.51.160:5689
Welcome aya! Waiting for the game to start...
Your answer (or 'exit' to quit): Round starting in 1 minute! GET READY!
Question 1: 2 * 5?
10
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
amman
Your answer (or 'exit' to quit): paris
Your answer (or 'exit' to quit): |
```



```

[sara] Enter your username: sara
Connected to server at 172.19.51.160:5689
Welcome sara! Waiting for the game to start...
Your answer (or 'exit' to quit): Round starting in 1 minute! GET READY!
Question 1: 2 * 5?
10
Your answer (or 'exit' to quit): 11
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
paris
Your answer (or 'exit' to quit): |
```



```

[ansar] Enter server IP : 172.19.51.160
Enter server port : 5689
Enter your username: ansar
Connected to server at 172.19.51.160:5689
Welcome ansar! Waiting for the game to start...
Your answer (or 'exit' to quit): Question 1: 2 * 5?
10
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
paris
Your answer (or 'exit' to quit): |
```



```

[serve] player sara (('172.19.51.160', 54559)) answered: 11 - Incorrect
Broadcasted Question 2: What is the smallest prime number?
playersara (('172.19.51.160', 54559)) answered: 2 - Correct! +0.77 point
playeraya (('172.19.51.160', 52918)) answered: 2 - Correct! +0.46 points
playeransar (('172.19.51.160', 57811)) answered: 2 - Correct! +0.40 points
Broadcasted Question 3: What is the capital of France?
playersara (('172.19.51.160', 54559)) answered: paris - Correct! +0.66 points
player aya (('172.19.51.160', 52918)) answered: amman - Incorrect
playeransar (('172.19.51.160', 57811)) answered: paris - Correct! +0.31 points
Ignored subsequent answer from aya (('172.19.51.160', 52918))
```

Figure 43:game test5

6) if one of client send incorrect answer does not give any score

```

[aya] Welcome aya! Waiting for the game to start...
Your answer (or 'exit' to quit): Round starting in 1 minute! GET READY!
Question 1: 2 * 5?
10
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
amman
Your answer (or 'exit' to quit): paris
Your answer (or 'exit' to quit): Correct answer: Paris
Question 4: What is the square root of 81?
9
Your answer (or 'exit' to quit): |
```



```

[sara] Welcome sara! Waiting for the game to start...
Your answer (or 'exit' to quit): Round starting in 1 minute! GET READY!
Question 1: 2 * 5?
10
Your answer (or 'exit' to quit): 11
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
paris
Your answer (or 'exit' to quit): |
```



```

[ansar] Connected to server at 172.19.51.160:5689
Welcome ansar! Waiting for the game to start...
Your answer (or 'exit' to quit): Question 1: 2 * 5?
10
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
paris
Your answer (or 'exit' to quit): Correct answer: Paris
Question 4: What is the square root of 81?
9
Your answer (or 'exit' to quit): |
```



```

[serve] playeransar (('172.19.51.160', 57811)) answered: 2 - Correct! +0.40 points
Broadcasted Question 3: What is the capital of France?
playersara (('172.19.51.160', 54559)) answered: paris - Correct! +0.66 points
player aya (('172.19.51.160', 52918)) answered: amman - Incorrect
playeransar (('172.19.51.160', 57811)) answered: paris - Correct! +0.31 points
Ignored subsequent answer from aya (('172.19.51.160', 52918))
Broadcasted Question 4: What is the square root of 81?
playeransar (('172.19.51.160', 57811)) answered: 9 - Correct! +0.94 points
playeraya (('172.19.51.160', 52918)) answered: 9 - Correct! +0.64 points
player sara (('172.19.51.160', 54559)) answered: 5 - Incorrect
```

Figure 44 :gamenetst6

```

10
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
amman
Your answer (or 'exit' to quit): paris
Your answer (or 'exit' to quit): Correct answer: Paris
Question 4: What is the square root of 81?
9
Your answer (or 'exit' to quit): Correct answer: 9
Question 5: In which year did World War II end?
1945
Your answer (or 'exit' to quit): |

10
Your answer (or 'exit' to quit): 11
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
paris
Your answer (or 'exit' to quit): Correct answer: Paris
Question 4: What is the square root of 81?
5
Your answer (or 'exit' to quit): Correct answer: 9
Question 5: In which year did World War II end?
1945
Your answer (or 'exit' to quit): |

10
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
paris
Your answer (or 'exit' to quit): Correct answer: Paris
Question 4: What is the square root of 81?
9
Your answer (or 'exit' to quit): Correct answer: 9
Question 5: In which year did World War II end?
1945
Your answer (or 'exit' to quit): |

10
Your answer (or 'exit' to quit): 11
Your answer (or 'exit' to quit): Correct answer: 10
Question 2: What is the smallest prime number?
2
Your answer (or 'exit' to quit): Correct answer: 2
Question 3: What is the capital of France?
paris
Your answer (or 'exit' to quit): Correct answer: Paris
Question 4: What is the square root of 81?
5
Your answer (or 'exit' to quit): Correct answer: 9
Question 5: In which year did World War II end?
1945
Your answer (or 'exit' to quit): |

points
Ignored subsequent answer from aya (('172.19.51.160', 52918))
Broadcasted Question 4: What is the square root of 81?
playeransar (('172.19.51.160', 57811)) answered: 9 - Correct! +0.94 points
playeraya (('172.19.51.160', 52918)) answered: 9 - Correct! +0.64 points
player sara (('172.19.51.160', 54559)) answered: 5 - Incorrect
Broadcasted Question 5: In which year did World War II end?
playersara ((('172.19.51.160', 54559))) answered: 1945 - Correct! +0.76 points
playeraya (('172.19.51.160', 52918)) answered: 1945 - Correct! +0.34 points
player ansar ((('172.19.51.160', 57811))) answered: 1999 - Incorrect
Ignored subsequent answer from ansar ((('172.19.51.160', 57811)))

```

Figure 45: game test7

7) after end of round score are calculated and who is the winner and the next round will be start after 30 sec

```

amman
Your answer (or 'exit' to quit): paris
Your answer (or 'exit' to quit): Correct answer: Paris
Question 4: What is the square root of 81?
9
Your answer (or 'exit' to quit): Correct answer: 9
Question 5: In which year did World War II end?
1945
Your answer (or 'exit' to quit): Correct answer: 1945
totalscore:
aya: 2.36
sara: 2.19
ansar: 2.05
Winning player: aya

Question 3: What is the capital of France?
paris
Your answer (or 'exit' to quit): Correct answer: Paris
Question 4: What is the square root of 81?
9
Your answer (or 'exit' to quit): Correct answer: 9
Question 5: In which year did World War II end?
1999
Your answer (or 'exit' to quit): 1945
Your answer (or 'exit' to quit): Correct answer: 1945
totalscore:
aya: 2.36
sara: 2.19
ansar: 2.05
Winning player: aya

player sara ((('172.19.51.160', 54559))) answered: 5 - Incorrect
Broadcasted Question 5: In which year did World War II end?
playersara ((('172.19.51.160', 54559))) answered: 1945 - Correct! +0.76 points
playeraya ((('172.19.51.160', 52918))) answered: 1945 - Correct! +0.34 points
player ansar ((('172.19.51.160', 57811))) answered: 1999 - Incorrect
Ignored subsequent answer from ansar ((('172.19.51.160', 57811)))
Round ended!
totalscore:
aya: 2.36
sara: 2.19
ansar: 2.05
Next round will start in 30 seconds...

```

Figure 46: test game8

Teamwork :

The project was done by three students: sara shrouf ,aya hamayl ,and ansar Mansour
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:

- Task 1 – Network Commands and Wiresharka : Ansar Mansour
- . Task 2 – Web Server : Aya Hamayl
- Task 3 – UDP Client-Server Trivia Game Using Socket Programming : Sara Shrouf
- Report: Each of us added her own task and shared the rest of the requirements



Figure 47 :teamwork

Conclusion:

This project allowed us to explore the fundamentals of socket programming and computer networking. We successfully implemented various tasks, such as working with network commands, capturing DNS queries, and developing both a web server and a trivia game server using UDP. Through this hands-on experience, we gained a deeper understanding of networking protocols like TCP and UDP and learned how to design and configure web servers. The project also enhanced our teamwork skills and gave us valuable experience in solving real-world networking problems, preparing us for future challenges in the field

References :

- [1] : https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html [Accessed on 30 Nov 2024].
- [2] : <https://www.geeksforgeeks.org/socket-programming-cc/> [Accessed on 30 Nov 2024]
- [3] : <https://www.fortinet.com/resources/cyberglossary/tcp-ip> [Accessed on 30 Nov 2024].
- [4] : <https://www.fortinet.com/resources/cyberglossary/user-datagram-protocol-udp> [Accessed on 30 Nov 2024].
- [5] : <https://iximiuz.com/en/posts/writing-web-server-in-python-sockets/> [Accessed on 30 Nov 2024].
- [6] : <https://developer.mozilla.org/en-US/docs/Web/HTTP> [Accessed on 30 Nov 2024].
- [7] : <https://www.geeksforgeeks.org/socket-programming-python/> [Accessed on 30 Nov 2024].