



BIRZEIT UNIVERSITY

**Faculty of Engineering & Technology – Computer Systems
Engineering Department**

First Semester 2024-2025

Computer Network ENCS3320

Project 1

Prepared by:

Basmala Abu Hakema - 1220184 - 4

Ibraheem Saadeh – 1221701 - 4

Fatima Al Zahra' Mustafa – 1220259 - 5

Instructor: Dr. Ibrahim Nimer , Dr. Muhammed Jubran

Date: 9th Nov 2024

Table of Contents :

Theory	5
Socket Programming and Networking Fundamentals	5
Web Server Implementation.....	5
UDP Client-Server Trivia Game	6
Task 1 – Network Commands and Wireshark:	7
a. Providing a brief explanation of each of the following commands:	7
b. performing the following actions:	8
c. Using the Wireshark packet analyzer to capture a DNS query and replying for any hostname of our choice: .	14
Results and Discussion.....	15
Task 2 – Web Server	15
The socket code.....	16
The main English Webpage (main_en.html).....	19
The main English code.....	19
The supporting material for English version	21
The main Arabic Webpage (main_ar.html).....	22
The main Arabic code	22
The supporting material for arabic version	24
From the first device	25
The main English file look	25
The main Arabic file look	27
From the second device.....	29
The main English File look	30
The main Arabic File look	31
Requesting The supporting material.....	32
The style.css File.....	33
When the server run :	35
The HTTP request in command line:	35
1-When the request / the browser displays:.....	35
2-When the request /en the browser displays:.....	36
Request Ritaj.....	37
Request Textbook website	37
Request supporting material	38
3-When the request /ar the browser displays:	44
Requeste Ritaj	44
Request Textbook website	45
Request supporting material	46
Task 3 UDP Client-Server Trivia Game Using Socket Programming.....	51
client code :	51
Server code :	53

Traviria Game Example	57
References.....	65
Team work.....	66

Table of figures:

Figure 1 client-server system	6
Figure 2: ipconfig/all.....	8
Figure 3: Ping a device in The Same Network from Laptop To Smart Phone	9
Figure 4: <i>Ping discover.engineering.utoronto.ca</i>	10
Figure 5 tracert discover.engineering.utoronto.ca	11
Figure 6: nslookup discover.engineering.utoronto.ca.....	12
Figure 7: telnet discover.engineering.utoronto.ca 80	12
Figure 8: telnet.....	13
Figure 9: <i>Result From Wireshark.</i>	14
Figure 10 The socket code	17
Figure 11 main_en.html	20
Figure 12 supporting_material_en.html	21
Figure 13 main_ar.html.....	23
Figure 14 supporting_material_ar.html	24
Figure 15 English web page	26
Figure 16 Arabic web page	28
Figure 17:other device.....	29
Figure 18 style.css.....	34
Figure 19 The http request	35
Figure 20 HTTP / request in command line	36
Figure 21 /en request.....	36
Figure 22HTTP /en request in command line	36
Figure 23 HTTP /supporting_matrial_en.html request in command line	39
Figure 24 Error massege	39
Figure 25 /ar request	44
Figure 26 The HTTP /ar request in command line	44
Figure 27 The HTTP/supporting_material_ar.html request in command line	47
Figure 28 HTTP /.css request in command line	50
Figure 29 client code.....	51
Figure 30 The question area	53
Figure 31 Server code	55

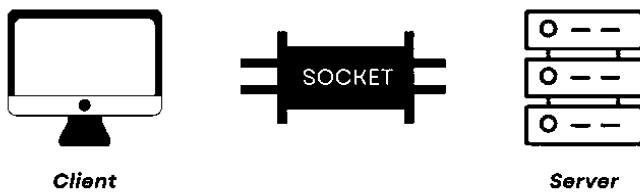
Theory

Socket Programming and Networking Fundamentals

Socket programming is a crucial concept in network communication, allowing devices to exchange data over a network. A socket serves as an endpoint for sending and receiving data, enabling communication between devices using specific network protocols. These protocols, such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol), define how data is transmitted and managed between devices.

TCP is a reliable, connection-oriented protocol that ensures the successful delivery of data through mechanisms like error checking, acknowledgment, and retransmission. It guarantees data integrity and delivery order, making it ideal for applications where accuracy is critical. On the other hand, UDP is a connectionless protocol that focuses on speed, sacrificing reliability for quicker communication. It is suitable for applications like real-time systems and gaming, where fast transmission is more important than ensuring every packet arrives correctly.

Understanding both TCP and UDP is fundamental for building efficient and robust network applications, as each protocol has its strengths and appropriate use cases.



Web Server Implementation

A web server is a system that handles incoming requests from clients (e.g., browsers) and provides requested resources such as HTML pages, images, and files. It listens on a specific port (e.g., 5698) and processes HTTP requests sent by clients. These requests typically specify a file or resource that the client wants to access, and the server responds with the appropriate content, accompanied by a header that defines the content type (e.g., text/html for web pages, image/png for images).

When a user requests a file that doesn't exist on the server, the server can either redirect the user to another resource or return an error, such as a 404 Not Found page. By utilizing socket programming, the server can manage multiple client requests simultaneously, handle HTTP protocols, and ensure smooth communication between the client and server. We used HTML to create the structure of the website, such as headings, sections, and buttons. CSS was used to style the website, making it look nice and easy to navigate.

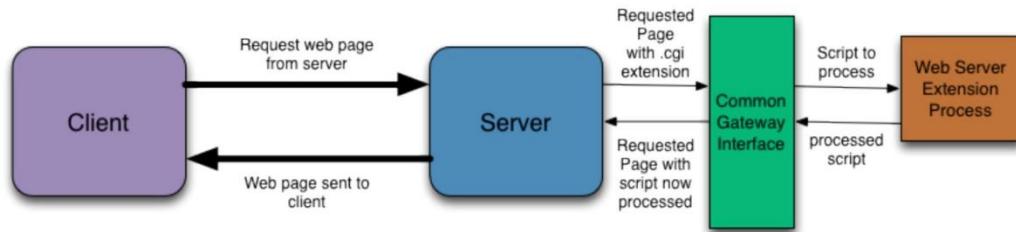


Figure 1client-server system

UDP Client-Server Trivia Game

A UDP-based client-server trivia game operates by leveraging the UDP protocol to create a fast, real-time gaming experience. The server manages the game by handling client connections, broadcasting questions, and collecting answers. It also tracks scores and announces winners after each round.

Clients connect to the server and receive questions to answer within a limited time. The server only accepts the first answer from each client and assigns points based on the correctness and timing of the responses. After each round, the server broadcasts the updated scores to all clients, keeping them informed of their standing. The use of UDP ensures low-latency communication, making the game interactive and responsive, as it prioritizes speed over reliability, which is ideal for fast-paced activities like trivia games. The server ensures that the game progresses only when a sufficient number of players are connected and handles game flow by organizing rounds and providing timely updates. This setup fosters an engaging and competitive environment for players while utilizing the efficiency of UDP for communication.

Python was used to build the backend of this game because it's simple and powerful for handling game logic. It processes the questions, checks answers, and keeps track of scores

Task 1 – Network Commands and Wireshark:

- a. Providing a brief explanation of each of the following commands:

1- Ipconfig:

Displays detailed information about network configurations, such as IP addresses, subnet masks, and gateways . It is useful for troubleshooting network connectivity and identifying configuration issues.

2- Ping:

Sends ICMP Echo Request packets to a target host and measures the response time.Used to test connectivity, determine network latency, and check for packet loss.

3- Tracert:

Traces the path data packets take to reach a destination, showing each hop and latency .Helps diagnose connectivity problems and pinpoint where delays or failures occur.

4- Telnet:

A protocol and tool to connect to remote devices over a network, providing a command-line interface .Commonly used for remote management and troubleshooting, though it lacks encryption.

5- Nslookup:

Queries DNS servers to resolve domain names to IP addresses or perform reverse lookups .It is used to troubleshoot DNS-related issues and verify proper name resolution.

b. performing the following actions:

1. ipconfig/all:

```
Ethernet adapter Ethernet 2:

  Connection-specific DNS Suffix . . . .
  Link-local IPv6 Address . . . . . : fe80::cabd:357e:8f91:9440%9
  IPv4 Address . . . . . : 192.168.56.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :

Wireless LAN adapter Wi-Fi:

  Connection-specific DNS Suffix . . . .
  Description . . . . . . . . . : Intel(R) Wi-Fi 6 AX201 160MHz
  Physical Address . . . . . . . . . : F0-20-FF-45-37-0B
  DHCP Enabled . . . . . . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes
  Link-local IPv6 Address . . . . . : fe80::e281:16e1:7b8e:2210%7(Preferred)
  IPv4 Address . . . . . . . . . : 192.168.0.152(Preferred)
  Subnet Mask . . . . . . . . . : 255.255.255.0
  Lease Obtained . . . . . . . . . : Monday, November 11, 2024 05:18:00 PM
  Lease Expires . . . . . . . . . : Monday, November 11, 2024 08:18:11 PM
  Default Gateway . . . . . . . . . : 192.168.0.1
  DHCP Server . . . . . . . . . : 192.168.0.1
  DHCPv6 IAID . . . . . . . . . : 133177599
  DHCPv6 Client DUID . . . . . . . . . : 00-01-00-01-2D-77-0E-71-E8-9C-25-C0-D9-3A
  DNS Servers . . . . . . . . . : 192.168.0.1
  NetBIOS over Tcpip. . . . . . . . . : Enabled
```

Figure 2: ipconfig/all

Ip address: 192.168.56.1

Subnet mask: 255.255.255.0

Default gateway: 192.168.0.1

Domain Name System (DNS): 192.168.0.1

2. ping:

IPv4 Address (laptop): 192.168.56.1

IPv4 Address (smartphone): 192.168.1.47

```
C:\Users\USER>ping 192.168.1.47
```

```
Pinging 192.168.1.47 with 32 bytes of data:
```

```
Reply from 192.168.1.47: bytes=32 time=37ms TTL=64
```

```
Reply from 192.168.1.47: bytes=32 time=47ms TTL=64
```

```
Reply from 192.168.1.47: bytes=32 time=71ms TTL=64
```

```
Reply from 192.168.1.47: bytes=32 time=61ms TTL=64
```

```
Ping statistics for 192.168.1.47:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
        Minimum = 37ms, Maximum = 71ms, Average = 54ms
```

Figure 3: Ping a device in The Same Network from Laptop To Smart Phone

- The ping command sends ICMP echo requests to the IP address 192.168.1.47, with each packet being 32 bytes.
- The reply line shows a response from 192.168.1.47, confirming the device responded to the request; the "time" parameter indicates the round-trip time, and "TTL" (Time To Live) shows the maximum hops before the packet is discarded.
- The final section provides ping statistics, showing 4 packets sent, all 4 received successfully (0% loss), with no packet loss.

3. Ping *discover.engineering.utoronto.ca*:

```
PS C:\Users\Asus> 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=77ms TTL=53
Reply from 23.185.0.2: bytes=32 time=60ms TTL=53
Reply from 23.185.0.2: bytes=32 time=60ms TTL=53
Reply from 23.185.0.2: bytes=32 time=60ms TTL=53

Ping statistics for 23.185.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 60ms, Maximum = 77ms, Average = 64ms
```

Figure 4: *Ping discover.engineering.utoronto.ca*

The ping results for *discover.engineering.utoronto.ca* show a relatively low response time (59–60 ms) from Palestine, suggesting the server is closer than Canada. The IP address (23.185.0.2) likely belongs to a content delivery network (CDN), which stores content in multiple global locations to improve access speed. Thus, while the main server might be in Canada, you are likely connecting to a nearby CDN server, resulting in reduced latency.

4. tracert discover.engineering.utoronto.ca:

```
PS C:\Users\Asus> 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  192.168.0.1
 2   2 ms    1 ms    1 ms www.webgui.Nokiawifi.com [192.168.1.254]
 3   3 ms    3 ms    2 ms ADSL-185.17.235.202.mada.ps [185.17.235.202]
 4   3 ms    3 ms    2 ms 172.16.250.77
 5   *        3 ms    4 ms 10.160.160.253
 6  55 ms   66 ms   55 ms et-2-3-3-100.cr7-fra2.ip4.gtt.net [195.72.86.13]
 7  60 ms   58 ms   54 ms ae22.cr11-fra2.ip4.gtt.net [89.149.180.226]
 8   *        *        Request timed out.
 9  56 ms   56 ms   55 ms be3763.ccr41.fra05.atlas.cogentco.com [154.54.76.209]
10  56 ms   56 ms   61 ms be5484.rcr22.fra06.atlas.cogentco.com [130.117.1.2]
11  58 ms   57 ms   58 ms fastly.demarc.cogentco.com [149.11.20.186]
12  59 ms   59 ms   59 ms 23.185.0.2
Trace complete.
```

Figure 5 tracert discover.engineering.utoronto.ca

In some cases, the traceroute may encounter hops that do not respond to the traceroute request, resulting in "Request timed out" messages. This could be due to network congestion, firewall configurations, or other reasons.

In the last hop, you reach the server at 23.185.0.2, which corresponds to discover.engineering.utoronto.ca. The response time is 59 ms, which is typical for reaching a CDN or server in a distant location, like Canada, from Palestine.

5. nslookup discover.engineering.utoronto.ca:

```
C:\Users\USER>nslookup discover.engineering.utoronto.ca
Server: 192.168.1.1
Address: 192.168.1.1

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

Figure 6: nslookup discover.engineering.utoronto.ca

The nslookup command for discover.engineering.utoronto.ca provides information about the domain name and its associated IP addresses.

Server: 192.168.1.1: This line indicates the DNS server used for the nslookup. And the DNS server at IP address 192.168.1.1 is providing the response.

Non-authoritative answer: This message indicates that the response comes from a DNS server other than the authoritative DNS server for the domain.

discover.engineering.utoronto.ca resolves to three IP addresses, these are the destinations that the computer would communicate with when accessing discover.engineering.utoronto.ca over the internet.

6. telnet discover.engineering.utoronto.ca:

```
C:\>telnet discover.engineering.utoronto.ca 80
GET / HTTP/1.1
Host: discover.engineering.utoronto.ca
```

Figure 7: telnet discover.engineering.utoronto.ca 80

```
HTTP/1.1 301 Moved Permanently
Connection: keep-alive
Content-Length: 162
Content-Type: text/html
Location: https://discover.engineering.utoronto.ca/
Server: nginx
X-Pantheon-Styx-Hostname: styx-fe1fe2-h-68b569ddf9-t58fs
X-Styx-Req-Id: 04e3a793-ae95-11ef-9b66-5e616499056e
Cache-Control: public, max-age=86400
Date: Sat, 30 Nov 2024 20:42:28 GMT
X-Served-By: cache-yyz4572-YYZ, cache-mrs10569-MRS
X-Cache: MISS, HIT
X-Cache-Hits: 0, 0
X-Timer: S1732999349.572143,VS0,VE5
Vary: Cookie, Cookie
Age: 85296
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>
<hr><center>nginx</center>
</body>
</html>
```

Figure 8: telnet

The HTTP response 301 Moved Permanently indicates that the requested resource has been permanently relocated to a new URL, provided in the Location header (<https://discover.engineering.utoronto.ca/>). This status code informs clients (e.g., browsers or tools) to update their records and access the resource using the new URL in the future. The headers, such as Cache-Control and Age, show caching details, allowing intermediate systems to cache the redirection for up to 24 hours. Other headers, like Server (nginx), identify the web server. The response body contains a simple HTML document explaining the redirection, which is typically displayed if a client doesn't automatically follow the redirect. To proceed, i should manually access the new URL over HTTPS.

- c. Using the Wireshark packet analyzer to capture a DNS query and replying for any hostname of our choice:

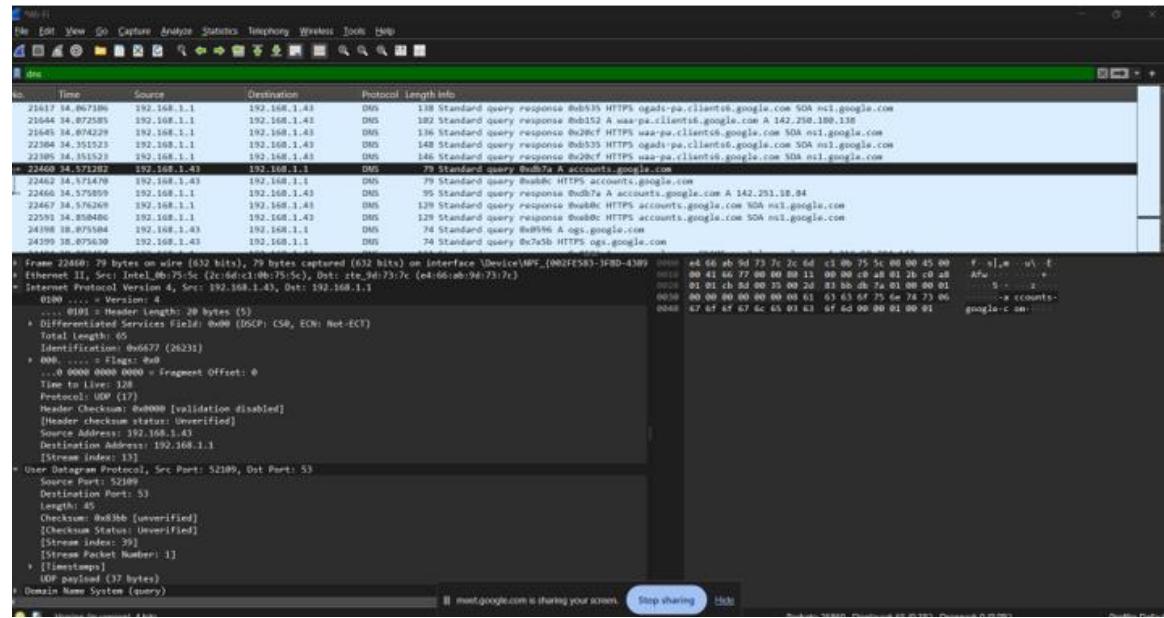


Figure 9: Result From Wireshark.

1. Version: the internet protocol addressing version is IPV4.
2. Header Length: the total length of the header is 20 bytes.
3. TTL: The packet can travel through 128 routers.
4. Source Address: the local IP address of my PC = 192.168.1.43.
5. Destination Address: the IP address of the destination = 192.168.1.1.
6. Source Port: the source port of the packets = 52109
7. Destination Port: the destination port of the packets = 53.

Results and Discussion

Task 2 – Web Server

The website contains information about us, including our photos, projects, and hobbies. It also provides explanations and figures related to network delays. From the website, you can access Ritaj, a textbook, or supporting materials, which you can search for specific photo or video .the website is available in both languages English and Arabic .

The website built using HTML for the structure, styled with CSS, and Python used for the driver file to handle functionality and interaction.

The socket code

```
driver.py  X  main_ar.html  main_en.html  supporting_material_ar.html  supporting_material_en.html
network_project > driver.py > ...
1 import socket
2 import os
3
4 # Set the server address and port
5 serverAddress = ('127.0.0.1', 5698)
6 # Create a TCP socket
7 serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8 # Bind the server socket to the address and port
9 serverSocket.bind(serverAddress)
10 # Listen for incoming connections
11 serverSocket.listen(10)
12
13 # Set the base directory where all the files are located
14 file_path_base = "C:/Users/HP/Desktop/my/network_project/" → Note: if you want to turn the server in your computer You
15                                         should make sure that you put the correct path...
16 while True:
17     print("The server is ready to receive")
18     # Accept a new connection
19     connection, clientAddress = serverSocket.accept()
20
21     try:
22         # Receive the request from the client
23         data = connection.recv(1024).decode()
24         # Parse the request line
25         request_line = data.splitlines()[0]
26         # Get the requested URI
27         requestUri = request_line.split()[1] if len(request_line.split()) > 1 else "/"
28
29         # Handle specific request for processing material
30         if requestUri.startswith('/process_request'):
31             # Extract query parameters
32             query_string = requestUri.split('?')[1] if '?' in requestUri else ""
33             params = dict([param.split('=') for param in query_string.split('&')])
34             file_name = params.get('fileName', '')
35
36             # Check if the file exists locally
37             file_path = os.path.join(file_path_base, file_name)
38             if os.path.exists(file_path):
39                 # Serve the file if it exists locally
40                 content_type = "image/png" if file_name.endswith(".png") else "image/jpeg"
41                 with open(file_path, "rb") as f:
42                     connection.sendall(f"HTTP/1.1 200 OK\r\nContent-Type: {content_type}\r\n\r\n".encode() + f.read())
43             else:
44                 # Check the file extension for redirection
45                 file_extension = file_name.split('.')[1].lower()
46                 if file_extension in ['jpg', 'jpeg', 'png']:
47                     # Redirect to Google Image search
48                     redirect_url = f"https://www.google.com/search?q={file_name.replace(' ', '+')}&tbo=isch"
49                     connection.sendall(f"HTTP/1.1 307 Temporary Redirect\r\nLocation: {redirect_url}\r\n\r\n".encode())
50                 elif file_extension in ['mp4', 'avi', 'mkv', 'mov']:
51                     # Redirect to YouTube search
52                     redirect_url = f"https://www.youtube.com/results?search_query={file_name.replace(' ', '+')}"
53                     connection.sendall(f"HTTP/1.1 307 Temporary Redirect\r\nLocation: {redirect_url}\r\n\r\n".encode())
54                 else:
55                     # If file type is not recognized
56                     error_message = (
57                         "<html><head><meta charset='UTF-8'><title>Unsupported file type</title></head>"
58                         "<body><h1>Unsupported file type. Please enter a valid file name.</h1></body></html>"
59                     )
60                     connection.sendall(
61                         b"HTTP/1.1 400 Bad Request\r\nContent-Type: text/html\r\n\r\n" + error_message.encode())
62
63         continue
64     except Exception as e:
65         print(f"An error occurred: {e}")
66
67 finally:
68     serverSocket.close()
```

```

63
64     # Determine the file path and content type based on the request URI
65     file_path = ""
66     content_type = "text/html" # Default to HTML content types
67
68     if requestUri in ['/ar', '/main_ar.html']:
69         file_path = file_path_base + "main_ar.html"
70     elif requestUri in ['/', '/en', '/index.html', '/main_en.html']:
71         file_path = file_path_base + "main_en.html"
72     elif requestUri == '/supporting_material_en.html':
73         file_path = file_path_base + "supporting_material_en.html"
74     elif requestUri == '/supporting_material_ar.html':
75         file_path = file_path_base + "supporting_material_ar.html"
76     elif requestUri.endswith(".css"):
77         file_path = file_path_base + "style.css"
78         content_type = "text/css"
79     elif requestUri.endswith(".png") or requestUri.endswith(".jpg"):
80         # Serve image files
81         file_path = file_path_base + requestUri.strip("/")
82         content_type = "image/png" if requestUri.endswith(".png") else "image/jpeg"
83
84     # Check if the file exists and send it if found
85     if file_path and os.path.exists(file_path):
86         # Open and read the file in binary mode
87         with open(file_path, "rb") as f:
88             connection.sendall(f"HTTP/1.1 200 OK\r\nContent-Type: {content_type}\r\n\r\n".encode() + f.read())
89     else:
90         # If the file is not found, send a 404 error page
91         error_message = (
92             f"<html><head><meta charset='UTF-8'><title>Error 404</title></head>"
93             f"<body style='color: red;'><h1>The file is not found</h1>"
94             f"<p>Client IP: {clientAddress[0]}</p>"
95             f"<p>Client Port: {clientAddress[1]}</p>"
96             f"</body></html>"
97         )
98         connection.sendall(b"HTTP/1.1 404 Not Found\r\nContent-Type: text/html\r\n\r\n" + error_message.encode())
99
100 except Exception as e:
101     # Print any errors for debugging
102     print(f"An error occurred: {e}")
103 finally:
104     # Close the connection
105     connection.close()
106     print("Connection closed")

```

Figure 10 The socket code

This Python script creates a simple HTTP server using the socket library to handle client requests over a TCP connection. The server is set up to listen on all available network interfaces with the IP address 0.0.0.0 and port 5698. Once the server is initialized, it waits for incoming client connections using the `serverSocket.accept()` method. Upon accepting a connection, the server reads the HTTP request from the client using `connection.recv(1024)` and decodes the data. The request is then parsed to extract the request line, which contains the URI. The server checks the requested URI to determine which resource the client is asking for. If the URI points to a valid file, the server

constructs the full file path by joining the base directory (file_path_base) with the requested resource and verifies if the file exists using os.path.exists().

If the file exists, the server sends an HTTP response with a 200 OK status and includes the appropriate Content-Type header, such as text/html for HTML files or image/png for images. The server reads the file in binary mode and sends it to the client using connection.sendall(). The website consists of several HTML files, such as main_ar.html, main_en.html, and others, which are designed to serve as the content for different language versions of the website. These HTML files are stored in the server's base directory and are linked with the server through the URI requests. The server checks which HTML page is being requested and serves the corresponding file based on the request, ensuring the appropriate content is delivered based on the user's query.

If the requested file does not exist locally, the server checks the file extension and, based on the type of file requested, may redirect the client to an external site. For example, if the requested file is an image, the server sends a 307 Temporary Redirect response with a Location header pointing to Google Images. Similarly, for video files, the server redirects to a YouTube search. If the file type is unsupported or unrecognized, the server sends a 400 Bad Request response with a custom error message explaining the issue. The server also handles requests that include the /process_request endpoint, where query parameters such as fileName are extracted, and appropriate actions are taken based on the file extension. In these cases, the server either serves the file directly or redirects the user to an external resource.

Additionally, the server is designed to return 404 Not Found errors when the requested file is not available and includes details about the client's IP address and port in the error message for debugging. After each request is processed, the server ensures that the connection is closed with connection.close(). This script demonstrates key Python functionalities, including socket communication (socket.socket()), file handling (os.path.exists()), and HTTP response management (connection.sendall()), while the associated HTML files are integral to presenting the website's content to users. The HTML files, designed to render the necessary website pages, are connected to the server through the file path and URI handling mechanism in the code, making it a basic yet effective example of a file-serving web server.

The main English Webpage (main_en.html)

The main English code

```
network_project > main_en.html > html > body > section.team
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>ENCS3320-Webserver</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <header>
11         <h1>Welcome to ENCS3320 - Computer Networks Webserver</h1>
12     </header>
13
14     <section class="team">
15         <h2>Meet Our Team</h2>
16         <table>
17             <tr>
18                 <td class="team-member">
19                     
20                     <p><strong>Name:</strong> Basmalah Abu Hakma</p>
21                     <p><strong>ID:</strong> 1220184</p>
22                     <p><strong>Skills:</strong> Mathematics, Critical Thinking, Communication Skills</p>
23                     <p><strong>Projects:</strong></p>
24                     <ul>
25                         <li>Handling populations of distincts using a linked list</li>
26                         <li>Developing undo-redo operation with a stack structure</li>
27                         <li>Hashing project</li>
28                     </ul>
29                 </td>
30                 <td class="team-member">
31                     
32                     <p><strong>Name:</strong> Ibrahim Saadah</p>
33                     <p><strong>ID:</strong> 1221701</p>
34                     <p><strong>Skills:</strong> Music, Leadership, Programming</p>
35                     <p><strong>Projects:</strong></p>
36                     <ul>
37                         <li>Developing a control unit in ARM assembly</li>
38                         <li>Handling populations of distincts using a linked list</li>
39                     </ul>
40                 </td>
41                 <td class="team-member">
42                     
43                     <p><strong>Name:</strong> Fatima AL Zahraa Mustafa</p>
44                     <p><strong>ID:</strong> 1220259</p>
45                     <p><strong>Skills:</strong> Problem Solving, Mathematics, Adaptability</p>
46                     <p><strong>Projects:</strong></p>
47                     <ul>
48                         <li>Handling ADT polynomial operations using linked lists and stacks</li>
49                         <li>Performing various polynomial operations</li>
50                         <li>ARM assembly project</li>
51                     </ul>
52                 </td>
53             </tr>
54         </table>
55     </section>
56 
```

```

57 <section class="topic">
58   <h2>Understanding Network Delay</h2>
59   <p>Network delay is the time it takes for data to travel from the source to the destination across a network. It is a critical factor
60
61   <p>The four main sources of delay are:</p>
62   <ol>
63     <li>
64       <strong>Propagation Delay:</strong> The time it takes for a signal to travel through the physical medium (like cables or wireless)
65       <br>
66     </li>
67     <li>
68       <strong>Transmission Delay:</strong> The time required to send the entire data packet onto the network, determined by the packet's size
69       <br>
70     </li>
71     <li>
72       <strong>Queuing Delay:</strong> The time a packet spends waiting in a router's queue due to network congestion or traffic load
73       <br>
74     </li>
75     <li>
76       <strong>Processing Delay:</strong> The time taken by network devices (like routers) to process the packet, which includes task switching overhead
77       <br>
78     </li>
79   </ol>
80
81   <p>The total network delay is calculated by summing the individual delays:</p>
82   <p><strong>d<sub>total</sub> = d<sub>proc</sub> + d<sub>queue</sub> + d<sub>trans</sub> + d<sub>prop</sub></strong></p>
83
84   <p>Real-world internet delays can be measured using tools like <strong>Traceroute</strong>, which tracks the round-trip time for packets between two hosts
85   <br>
86 </section>
87
88 <footer>
89   <nav>
90     <a href="supporting_material_en.html">Supporting Material</a> |
91     <a href="https://gaia.cs.umass.edu/kurose_ross/index.php" target="_blank">Textbook Website</a> |
92     <a href="https://ritaj.birzeit.edu/" target="_blank">Ritaj</a>
93   </nav>
94 </footer>
95 </body>
96 </html>
97

```

Figure 11 main_en.html

The supporting material for English version

```
network_project > ◊ supporting_material_en.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Supporting Material</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <header>
11         <h1>Request Supporting Material</h1>
12     </header>
13
14     <section class="form-section">
15         <h2>Request an Image or Video</h2>
16         <p>Enter the name of the image or video you want. If unavailable, you will be redirected to a search page.</p>
17         <form action="/process_request" method="get">
18             <label for="fileName">File name (include extension):</label>
19             <input type="text" id="fileName" name="fileName" placeholder="Enter file name with extension" required>
20
21             <button type="submit">Submit</button>
22         </form>
23     </section>
24
25     <footer>
26         <nav>
27             <a href="main_en.html">Back to Main Page</a>
28         </nav>
29     </footer>
30 </body>
31 </html>
```

Figure 12 supporting_material_en.html

The main Arabic Webpage (main_ar.html)

The main Arabic code

```
driver.py          main_ar.html X  main_en.html ●  supporting_material_ar.html  supporting_material_en.html
network_project > 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      <title> خادم ويب الشبكات الحاسوبية - ENCS3320 </title>
7      <link rel="stylesheet" href="../static/css/style.css">
8  </head>
9  <body dir="rtl">
10     <header>
11         <h1> خادم الشبكات الحاسوبية - ENCS3320 مرحبا بكم في </h1>
12     </header>
13
14     <section class="team">
15         <h2> تعرف على فريقنا </h2>
16         <table>
17             <tr>
18                 <td class="team-member">
19                     
20                     <p><strong> بسمة أبو حاكمة </strong></p>
21                     <p><strong> الاسم: الرقم الجامعي </strong> 1220184</p>
22                     <p><strong> المهراء: الرياضيات، التفكير الابداعي، مهارات التواصل </strong></p>
23                     <p><strong> المعاشر: البرمجيات، القيادة، البرمجة </strong></p>
24                     <ul>
25                         <li> Handling populations of distincts using a linked list </li>
26                         <li> Developing undo-redo operation with a stack structure </li>
27                         <li> Hashing project </li>
28                     </ul>
29                 </td>
30                 <td class="team-member">
31                     
32                     <p><strong> إبراهيم ملاحات </strong></p>
33                     <p><strong> الاسم: الرقم الجامعي </strong> 1221701</p>
34                     <p><strong> المهراء: الموسيقى، القيادة، البرمجة </strong></p>
35                     <p><strong> المعاشر: البرمجة، حل المشكلات، الرياضيات، التكيف </strong></p>
36                     <ul>
37                         <li> Developing a control unit in ARM assembly </li>
38                         <li> Handling populations of distincts using a linked list </li>
39                     </ul>
40                 </td>
41                 <td class="team-member">
42                     
43                     <p><strong> فاطمة الزهراء مصطفى </strong></p>
44                     <p><strong> الاسم: الرقم الجامعي </strong> 1220259</p>
45                     <p><strong> المهراء: حل المشكلات، الرياضيات، التكيف </strong></p>
46                     <p><strong> المعاشر: البرمجة، حل المشكلات، الرياضيات، التكيف </strong></p>
47                     <ul>
48                         <li> Handling ADT polynomial operations using linked lists and stacks </li>
49                         <li> Performing various polynomial operations </li>
50                         <li> ARM assembly project </li>
51                     </ul>
52                 </td>
53             </tr>
54         </table>
55     </section>
56
```

```

56
57 <section class="topic">
58   <h2>فهم تأخير الشبكة</h2>
59   الشبكة هو الوقت الذي يستغرقه البيانات للسفر من المصدر إلى الوجهة عبر الشبكة. إنه عامل حاسم يؤثر على أداء التطبيقات الشبكية<p>
60
61   </p> المصادر الأربع الرئيسية للتأخير هي<br>
62   <ol>
63     <li>
64       <strong>تأخير الاتصال</strong> يمثل الأسلام أو اللاسلكي من المرسل إلى المستقبل<br>
65     <br> يستغرقه الإشارة للانتقال عبر الوسط المادي (مثل الأسلام أو اللاسلكي) من المرسل إلى المستقبل<br>
66     لمطلوب لإرسال الحزمة بأكملها عبر الشبكة، ويعتمد على حجم الحزمة وعرض النطاق الترددي للشبكة<br>
67     <br> حزمة في انتظار دورها في قائمة الانتظار لجهاز التوجيه بسبب ازدحام الشبكة أو العمل الزائد<br>
68     <br> جهاز الشبكة (مثل أجهزة التوجيه) لمعالجة الحزمة، ويشمل ذلك مهام مثل فحص الأخطاء والتوجيه<br>
69     <br> </li>
70     <li>
71       <strong>تأخير الإجمالي للشبكة</strong> يتم حساب التأخير الإجمالي للشبكة عن طريق جمع التأخيرات الفردية<br>
72     <br> </li>
73   </ol>
74
75   يمكن قياس التأخيرات في الإنترنط الحقيقية باستخدام أدوات مثل<br>
76   <br> ر والنقل والانتشار عند كل جهاز توجيه، <strong>Traceroute</strong>، موقع الكتاب<a href="https://gaia.cs.umass.edu/kurose_ross/index.php" target="_blank">https://gaia.cs.umass.edu/kurose_ross/index.php</a> | ريتا ج<a href="https://ritaj.birzeit.edu/" target="_blank">https://ritaj.birzeit.edu/</a>
77
78 </section>
79
80 <footer>
81   <nav>
82     <a href="supporting_material_ar.html">المواد الداعمة</a> | <a href="https://gaia.cs.umass.edu/kurose_ross/index.php" target="_blank">موقع الكتاب</a> | <a href="https://ritaj.birzeit.edu/" target="_blank">ريتا ج</a>
83   </nav>
84 </footer>
85 </body>
86 </html>
87
88
89
90
91
92
93
94
95
96
97

```

Figure 13 main_ar.html

The supporting material for arabic version

```
network_project > ◊ supporting_material_ar.html > ◊ html > ◊ body > ◊ section.content
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      <title> المواد المساعدة - ENCS3320</title>
7      <link rel="stylesheet" href="../static/css/style.css">
8  </head>
9  <body>
10     <header>
11         <h1>المواد المساعدة لدورة شبكات الكمبيوتر</h1>
12     </header>
13
14     <section class="content">
15         <h2>طلب صورة أو فيديو</h2>
16         <p>يرجى إدخال اسم الملف الذي تبحث عنه (بما في ذلك الامتداد)، إذا كان الملف غير متوفّر، سيتم إعادة توجيهك إلى صفحة بحث مور جوجل أو يوتوب</p>
17
18         <!-- نموذج لتنكين المستخدم من تقديم طلب البحث --!
19         <form id="searchForm" action="/process_request" method="GET">
20             <label for="fileName"> نوع الامتداد، مثلاً (protocol.png):</label>
21             <input type="text" id="fileName" name="fileName" required placeholder="أدخل اسم الملف مع الامتداد">
22             <br><br>
23
24             <!-- زر لإرسال ليد، عملية البحث --!
25             <button type="submit">بحث</button>
26         </form>
27
28         <!-- رسالة الخطأ التي ستظهر إذا لم يدخل المستخدم اسم الملف --!
29         <p id="errorMessage" style="color: red; display: none;"></p>
30     </section>
31
32     <footer>
33         <nav>
34             <!-- رابط للانتقال للعودة إلى الصفحة الرئيسية --!
35             <a href="ar">العودة إلى الصفحة الرئيسية</a>
36         </nav>
37     </footer>
38 </body>
39 </html>
```

Figure 14 supporting_material_ar.html

From the first device

The main English file look

Welcome to ENCS3320 - Computer Networks Webserver

Meet Our Team

Member	Name	ID	Skills	Projects
1	Basmalah Abu Hakma	1220184	Mathematics, Critical Thinking, Communication Skills	Handling populations of distincts using a linked list Developing undo-redo operation with a stack structure Hashing project
2	Ibrahim Saadah	1221701	Music, Leadership, Programming	Developing a control unit in ARM assembly Handling populations of distincts using a linked list
3	Fatima AL Zahraa Mustafa	1220259	Problem Solving, Mathematics, Adaptability	Handling ADT polynomial operations using linked lists and stacks Performing various polynomial operations ARM assembly project

Understanding Network Delay

Network delay is the time it takes for data to travel from the source to the destination across a network. It is a critical factor that affects the performance of networked applications.

The four main sources of delay are:

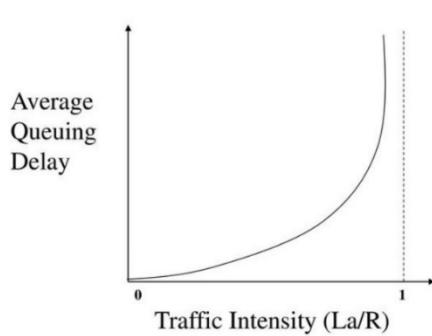
- Propagation Delay:** The time it takes for a signal to travel through the physical medium (like cables or wireless) from the sender to the receiver.

Packet Rate (per second)	Propagation Delay (seconds)
0.5	10
1	10
2	10
10	10

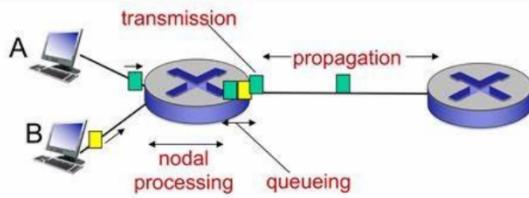
- Transmission Delay:** The time required to send the entire data packet onto the network, determined by the packet size and network bandwidth.

Number of Stations (SSs)	TBFB W=32 (ms)	TBE:B W=64 (ms)	RBCA (ms)
0	0	0	0
20	1.5	1.0	0.5
40	3.0	2.0	1.0
60	4.5	3.0	1.5
80	6.0	4.0	2.0
100	7.5	5.0	2.5
120	9.0	6.0	3.0
140	10.5	7.0	3.5

3. **Queuing Delay:** The time a packet spends waiting in a router's queue due to network congestion or traffic load.



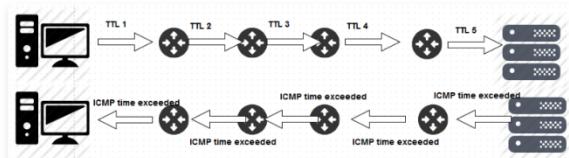
4. **Processing Delay:** The time taken by network devices (like routers) to process the packet, which includes tasks like error checking and routing.



The total network delay is calculated by summing the individual delays:

$$d_{\text{total}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

Real-world internet delays can be measured using tools like **Traceroute**, which tracks the round-trip time for packets to pass through routers. It measures the cumulative effect of processing, queuing, transmission, and propagation delays at each router.



[Supporting Material](#) | [Textbook Website](#) | [RitaJ](#)

Figure 15 English web page

The main Arabic file look

مرحباً بكم في ENCS3320 - خادم الشبكات الحاسوبية

تعرف على فريقنا

الاسم: فاطمة الزهراء مصطفى الرقم الجامعي: 1220259 المهارات: حل المسئكلات، الرياضيات، التكيف، المشاريع. المثلث: Handling ADT polynomial operations using linked lists and stacks Performing various polynomial operations ARM assembly project	الاسم: إبراهيم صلاح الدين الرقم الجامعي: 1221701 المهارات: الموسيقى، الملاحة، البرمجة المشاريع: Developing a control unit in ARM assembly Handling populations of distincts using a linked list	الاسم: سمية أبو حكمة الرقم الجامعي: 1220184 المهارات: الرياضيات، التكيف الاداري، مهارات التواصل المشاريع: Handling populations of distincts using a stack structure Developing undo-redo operation with a stack structure Hashing project

فهم تأخير الشبكة

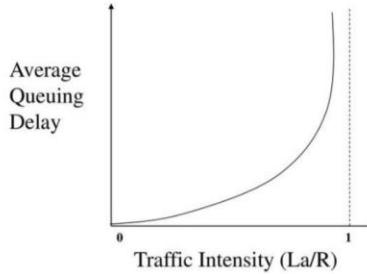
تأخير الشبكة هو الوقت الذي يستغرقه البيانات للسفر من المصدر إلى الوجهة عبر الشبكة. إنه عمل حاسم يؤثر على أداء التطبيقات الشبكية.

المصادر الأربع الرئيسية للتأخير هي:

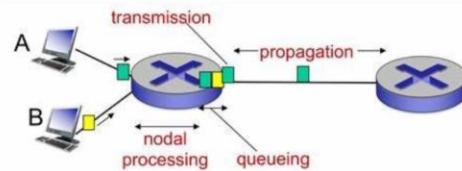
- تأخير الاتصال:** الوقت الذي يستغرقه الإشارة لانتقال غير الوسط المادي (مثل الألياف أو التسلك) من المرسل إلى المستقبل.

- تأخير الفضل:** الوقت المطلوب لإرسال الحزمة ينبعها عبر الشبكة، ويعتمد على حجم الحزمة وعرض النطاق الترددي للشبكة.

3. **نافر الإنتظار:** الوقت الذي تقضيه الحزمة في الانتظار تورها في قائمة الانتظار لجهاز التوجيه بسبب ازدحام الشبكة أو الحمل الزائد.



4. **نافر المعدجة:** الوقت الذي يستغرقه جهاز الشبكة (مثل أجهزة التوجيه) لمعالجة الحزمة، ويشمل ذلك مهام مثل فحص الأخطاء و التوجيه.



يتم حساب التأخير الإجمالي للشبكة عن طريق جمع التأخيرات الفردية:

$$d_{\text{total}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

يمكن قياس التأخيرات في الإنترنيت الحقيقة باستخدام أنواع مثل **Traceroute**، والتي تتغيب وقت الرحلة ذاتها وبينما للجزء غير أجهزة التوجيه، تقيس التأخير التراكمي لتأثيرات المعالجة والانتظار والتل والانتشار عند كل جهاز توجيه.

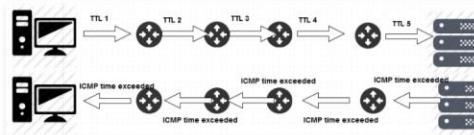


Figure 16 Arabic web page

From the second device

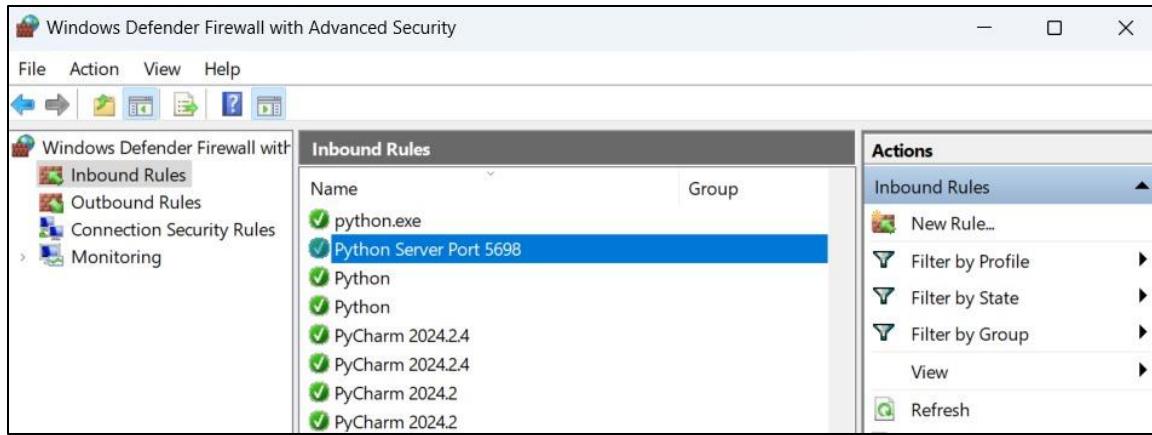


Figure 17:other device.

First we make the firewall Allows Incoming Connections by Create a new inbound rule:

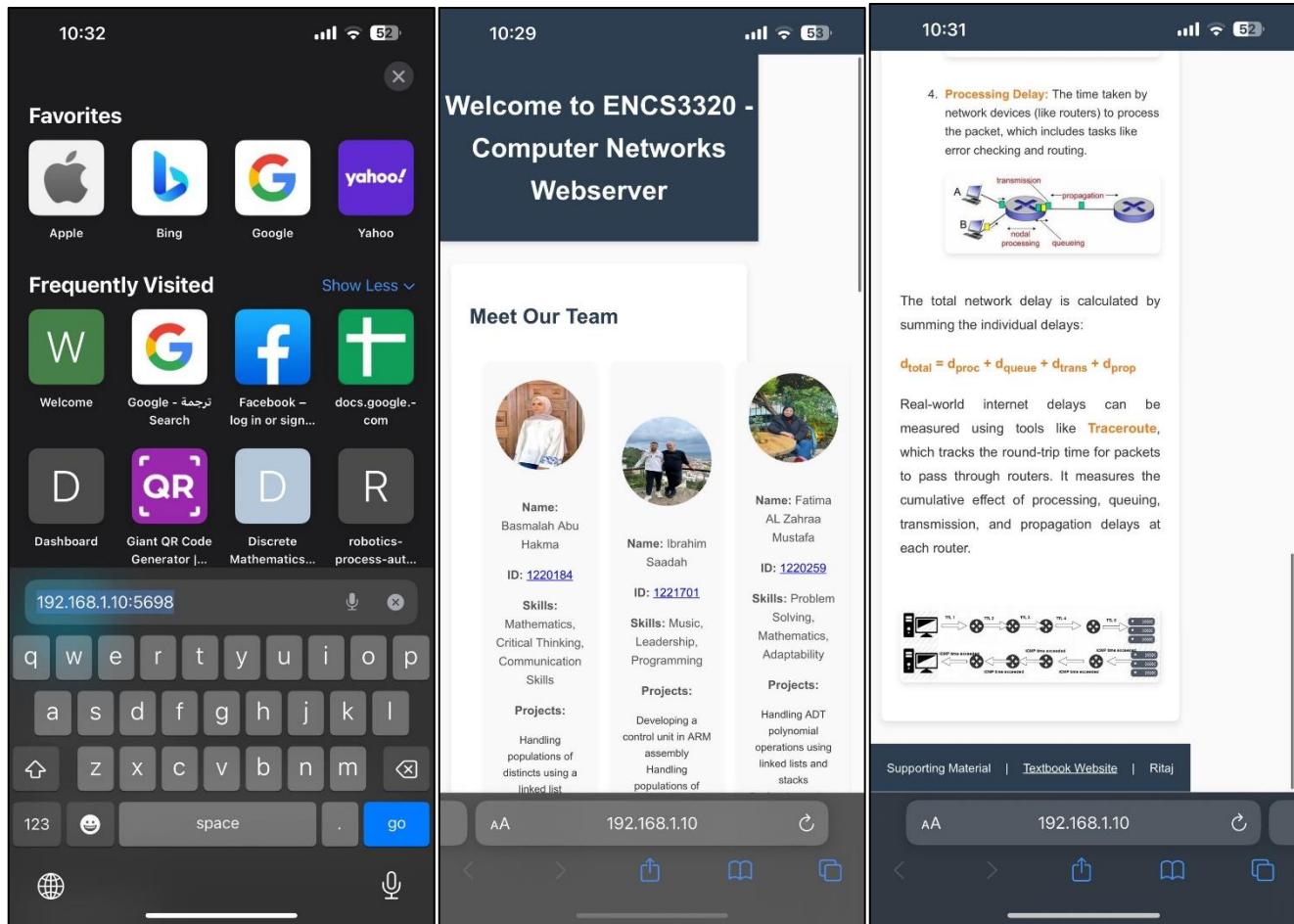
Rule Type: Port

Port: 5698

Allow the connection.

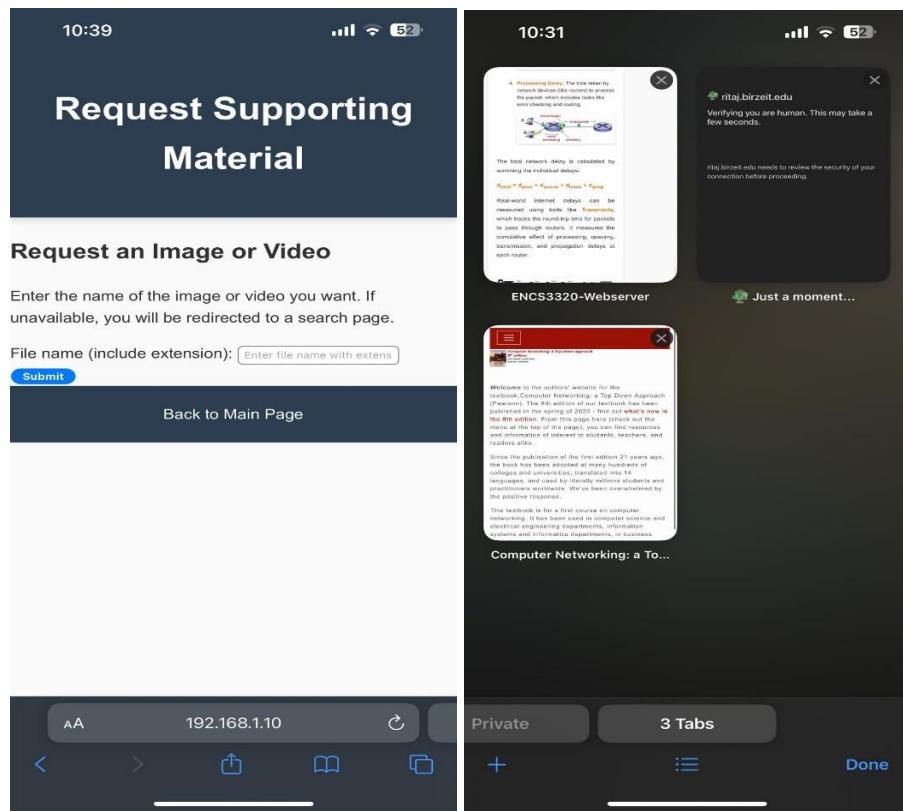
```
1 import socket
2 import os
3
4 # Set the server address and port
5 serverAddress = ('0.0.0.0', 5698)
6 # Create a TCP socket
7 serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8 # Bind the server socket to the address and port
9 serverSocket.bind(serverAddress)
10 # Listen for incoming connections
11 serverSocket.listen(10)
12
```

Note: In order to access The web page from other devices, we changed the IP to 0.0.0.0.

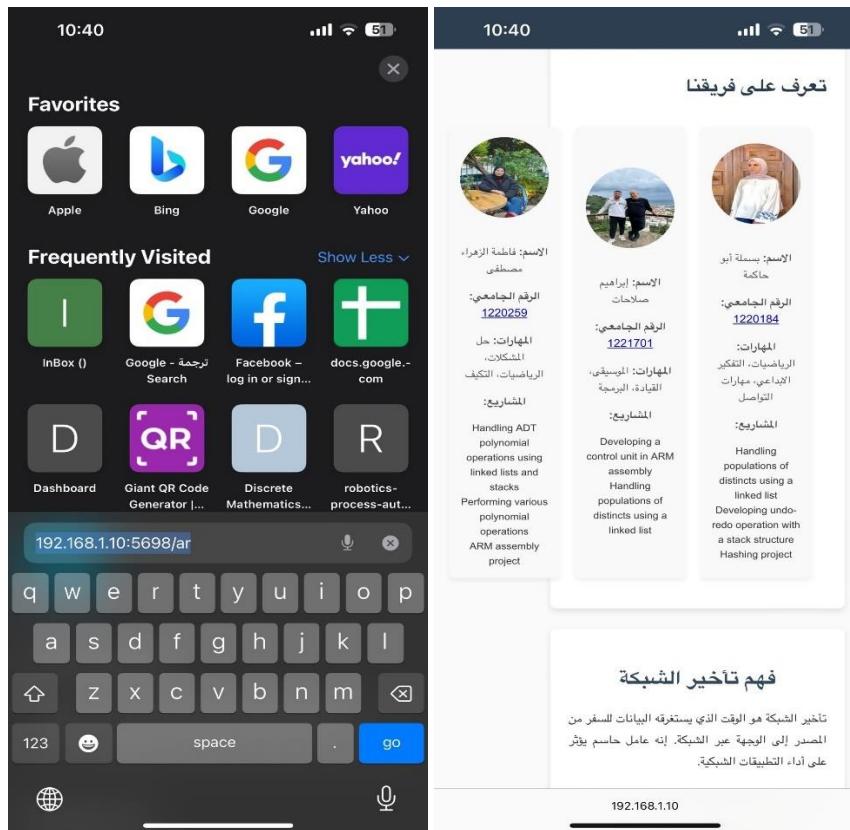


The main English File look

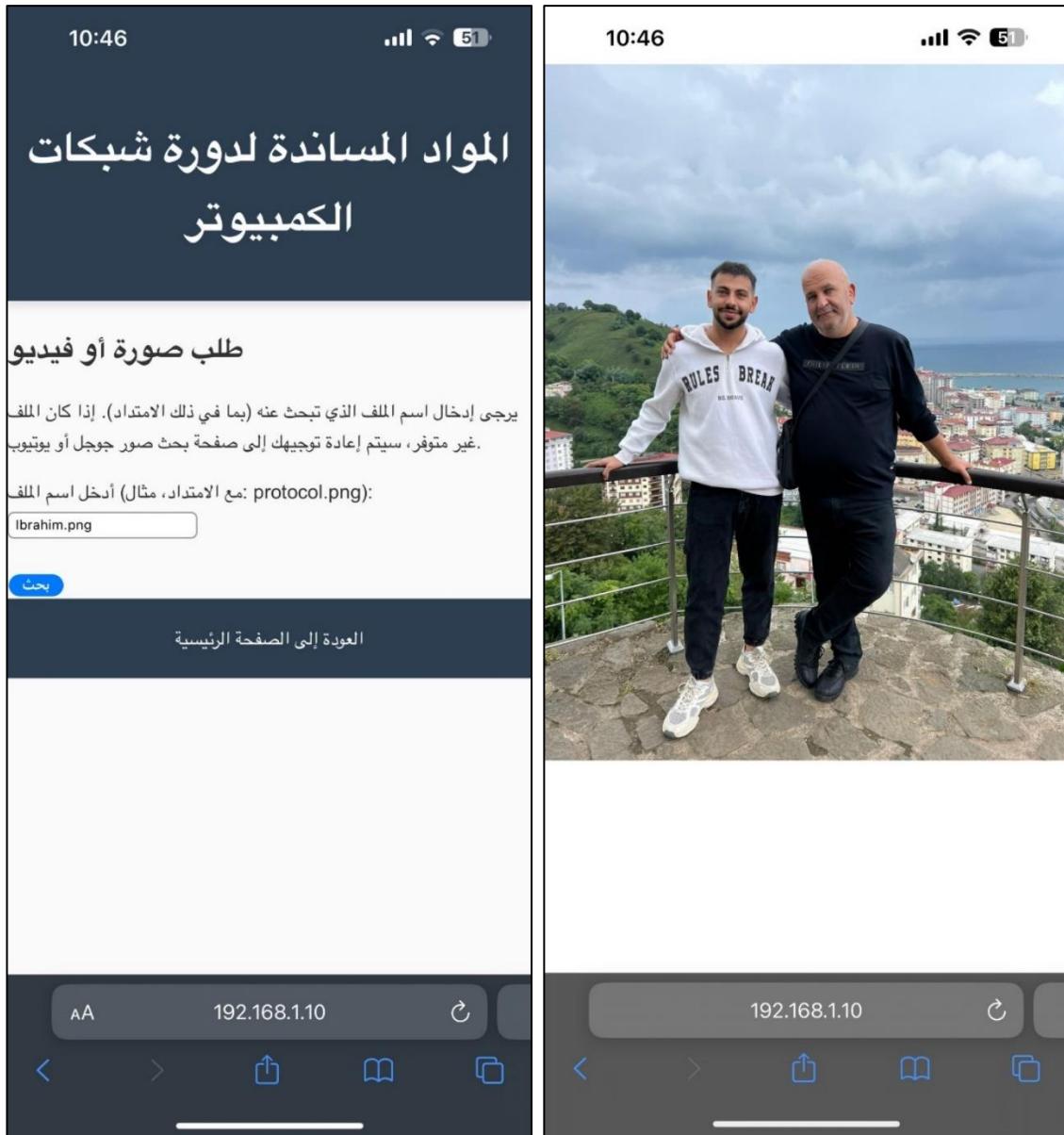
We used the router's IP and the port with it to access the site from the mobile.



The main Arabic File look



Requesting The supporting material



The style.css File

```
network_project > # style.css > ...
1  /* General Styles */
2  body {
3      font-family: 'Arial', sans-serif;
4      background-color: #fafafa;
5      color: #333;
6      margin: 0;
7      padding: 0;
8      line-height: 1.6;
9  }
10
11 /* Header Styling */
12 header {
13     background-color: #2c3e50;
14     color: #fff;
15     text-align: center;
16     padding: 40px 0;
17     box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
18 }
19
20     header h1 {
21         font-size: 36px;
22         margin: 0;
23     }
24
25 /* Team Section Styling */
26 .team {
27     margin: 30px 15px;
28     padding: 25px;
29     background-color: #fff;
30     border-radius: 8px;
31     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
32 }
33
34     .team h2 {
35         font-size: 28px;
36         color: #2c3e50;
37         margin-bottom: 20px;
38     }
39
40 table {
41     width: 100%;
42     border-collapse: separate;
43     border-spacing: 20px;
44 }
45
46 td.team-member {
47     width: 30%;
48     text-align: center;
49     padding: 15px;
50     background-color: #f9f9f9;
51     border-radius: 8px;
52     box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
53 }
54
55     td.team-member img {
56         width: 120px;
57         height: 120px;
58         border-radius: 50%;
59         margin-bottom: 15px;
60     }
61
62     td.team-member p {
63         font-size: 16px;
64         color: #555;
65     }
66
67     td.team-member ul {
68         list-style-type: none;
69         padding: 0;
70         margin: 0;
71     }
72
73         td.team-member ul li {
74             font-size: 14px;
75             color: #333;
76         }
```

```

77
78 /* Topic Section Styling */
79 .topic {
80     margin: 30px 15px;
81     padding: 25px;
82     background-color: #fff;
83     border-radius: 8px;
84     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
85 }
86
87 .topic h2 {
88     font-size: 30px;
89     color: #34495e;
90     margin-bottom: 20px;
91     text-align: center;
92 }
93
94 .topic p {
95     font-size: 18px;
96     color: #333;
97     margin-bottom: 20px;
98     line-height: 1.8;
99     text-align: justify;
100 }
101
102 .topic strong {
103     font-weight: bold;
104     color: #e67e22;
105 }
106
107 .topic em {
108     font-style: italic;
109     color: #8e44ad;
110 }
111
112 .topic ul, .topic ol {
113     margin-left: 20px;
114     margin-bottom: 20px;
115     font-size: 16px;
116     color: #333;
117 }
118
119 .topic ul li, .topic ol li {
120     margin-bottom: 10px;
121 }
122
123         .topic ul li strong, .topic ol li strong {
124             font-weight: bold;
125         }
126
127 .topic img {
128     width: 100%;
129     max-width: 700px;
130     height: auto;
131     margin: 20px 0;
132     border-radius: 8px;
133     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
134 }
135
136 /* Footer styling */
137 footer {
138     background-color: #2c3e50;
139     color: #fff;
140     text-align: center;
141     padding: 20px 0;
142     position: relative;
143     bottom: 0;
144     width: 100%;
145 }
146
147 footer nav {
148     margin: 0;
149 }
150
151     footer nav a {
152         color: #fff;
153         text-decoration: none;
154         padding: 0 15px;
155         font-size: 16px;
156     }
157
158         footer nav a:hover {
159             text-decoration: underline;
160         }

```

Figure 18 style.css

When the server run :

The HTTP request in command line:

```
network_project > driver.py > ...
//                                     title_path = file_path_base + "style.css"
78                                     content_type = "text/css"
79                                     elif requestUri.endswith(".png") or requestUri.endswith(".jpg"):
80                                         # Serve image files
81                                         file_path = file_path_base + requestUri.strip("/")
82                                         content_type = "image/png" if requestUri.endswith(".png") else "image/jpeg"
83
84                                     # Check if the file exists and send it if found
85                                     if file_path and os.path.exists(file_path):
86                                         # Open and read the file in binary mode
87                                         with open(file_path, "rb") as f:
88                                             connection.sendall(f"HTTP/1.1 200 OK\r\nContent-Type: {content_type}\r\n\r\n".encode() + f.read())
89                                     else:
90                                         # If the file is not found, send a 404 error page
91                                         error_message = (
92                                             f"<html><head><meta charset='UTF-8'><title>Error 404</title></head>"
93                                             f"<body style='color: red;'><h1>The file is not found</h1>"
94                                             f"<p>Client IP: {clientAddress[0]}</p>"
95                                             f"<p>Client Port: {clientAddress[1]}</p>"
96                                             f"</body></html>"
97                                         )
98                                         connection.sendall(b"HTTP/1.1 404 Not Found\r\nContent-Type: text/html\r\n\r\n" + error_message.encode())
99
100                                     except Exception as e:
101                                         # Print any errors for debugging
102                                         print(f"An error occurred: {e}")
103                                     finally:
104                                         # Close the connection
105                                         connection.close()
106                                         print("Connection closed")
107
108                                     PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS
109                                     PS C:\Users\HP\Desktop\my> & C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/Users/HP/Desktop/my/network_project/driver.py
110                                     The server is ready to receive
```

Figure 19 The http request

1-When the request / the browser displays:

Welcome to ENCS3320 - Computer Networks Webserver

Meet Our Team

Name	ID	Skills	Projects
Basmalah Abu Hakma	1220184	Mathematics, Critical Thinking, Communication Skills	Handling populations of distincts using a linked list Developing undo-redo operation with a stack structure Hashing project
Ibrahim Saadah	1221701	Music, Leadership, Programming	Developing a control unit in ARM assembly Handling populations of distincts using a linked list
Fatima AL Zahraa Mustafa	1220259	Problem Solving, Mathematics, Adaptability	Handling ADT polynomial operations using linked lists and stacks Performing various polynomial operations ARM assembly project

The HTTP request in command line

```
The server is ready to receive
IP: 127.0.0.1, Port: 59493
GET /basmala.png HTTP/1.1
Host: 127.0.0.1:5698
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 Edg/131.0.0.0
sec-ch-ua: "Microsoft Edge";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://127.0.0.1:5698/
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9

Connection closed
```

Figure 20 HTTP / request in command line

2-When the request /en the browser displays:

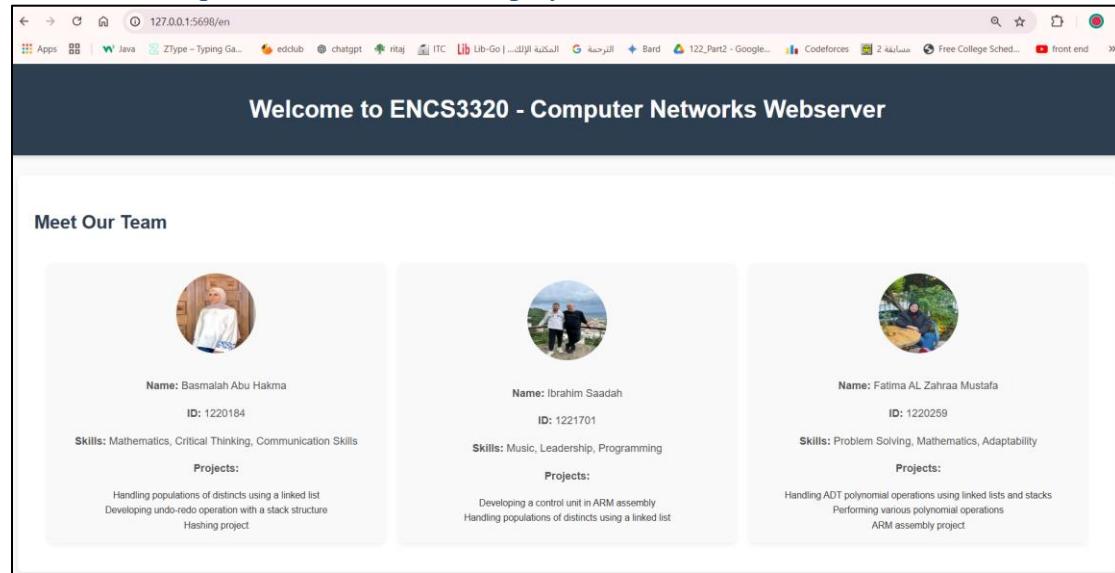


Figure 21 /en request

The HTTP request in command line

```
IP: 127.0.0.1, Port: 55506
GET /traceroute.jpg HTTP/1.1
Host: 127.0.0.1:5698
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 Edg/131.0.0.0
sec-ch-ua: "Microsoft Edge";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://127.0.0.1:5698/en
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9

Connection closed
The server is ready to receive
```

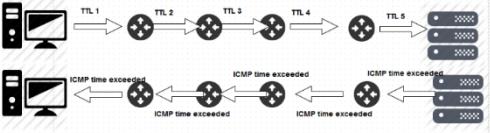
Figure 22HTTP /en request in command line

Request Ritaj

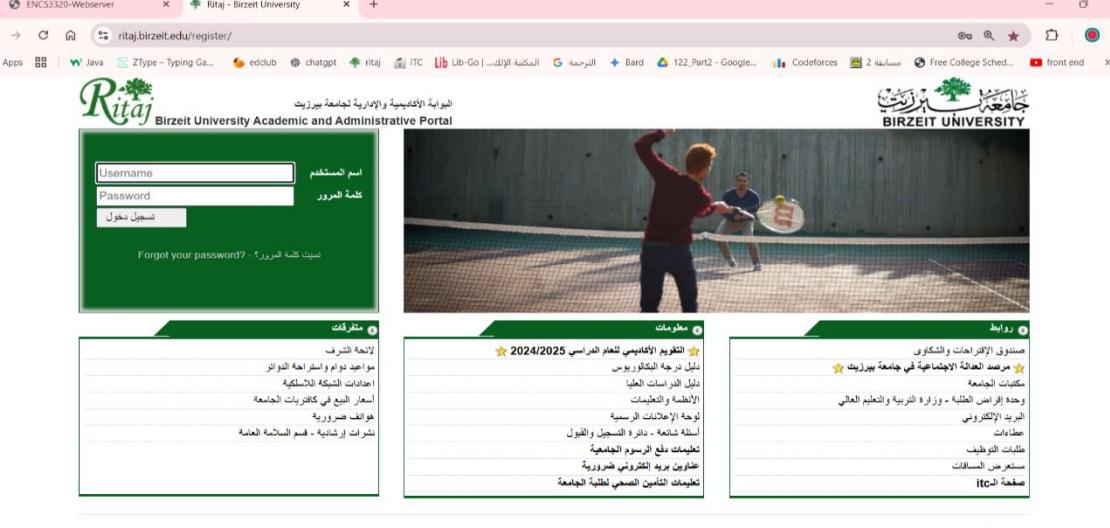
The total network delay is calculated by summing the individual delays:

$$d_{\text{total}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

Real-world internet delays can be measured using tools like [Traceroute](#), which tracks the round-trip time for packets to pass through routers. It measures the cumulative effect of processing, queuing, transmission, and propagation delays at each router.



Supporting Material | Textbook Website | Ritaj

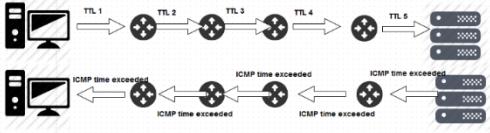


Request Textbook website

The total network delay is calculated by summing the individual delays:

$$d_{\text{total}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

Real-world internet delays can be measured using tools like [Traceroute](#), which tracks the round-trip time for packets to pass through routers. It measures the cumulative effect of processing, queuing, transmission, and propagation delays at each router.



Supporting Material | Textbook Website | Ritaj

Request supporting material :

The total network delay is calculated by summing the individual delays:

$$d_{\text{total}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

Real-world internet delays can be measured using tools like **Traceroute**, which tracks the round-trip time for packets to pass through routers. It measures the cumulative effect of processing, queuing, transmission, and propagation delays at each router.

Request Supporting Material

Request an Image or Video

Enter the name of the image or video you want. If unavailable, you will be redirected to a search page.

file name (include extension): Enter file name with extension

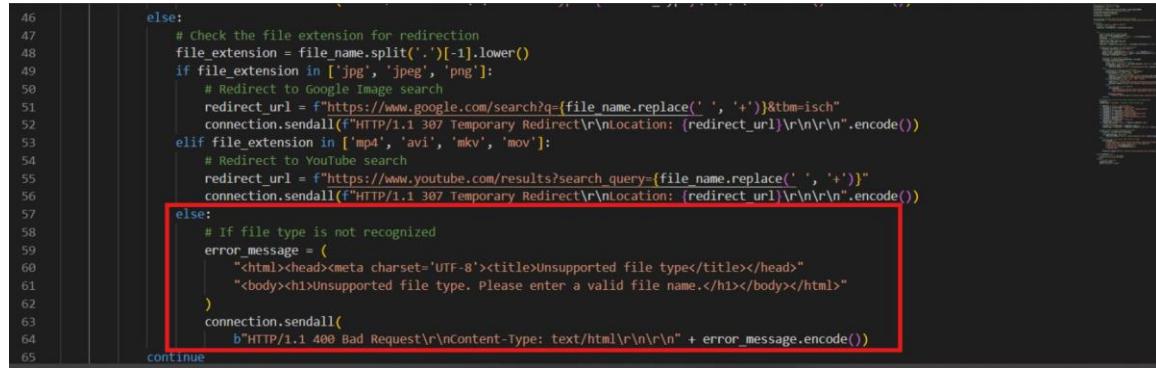
[Back to Main Page](#)

The HTTP request in command line

```
The server is ready to receive
IP: 127.0.0.1, Port: 58313
GET /style.css HTTP/1.1
Host: 127.0.0.1:5698
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 Edg/131.0.0.0
sec-ch-ua: "Microsoft Edge";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://127.0.0.1:5698/supporting_material_en.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9
```

Figure 23 HTTP /supporting_material_en.html request in command line

If we search for the name of a photo or video, like basmala.png, and it exists in the file, the result will be the photo or the video in the file . But if we search for something like car.png, which isn't in the file , it will search on the internet instead.



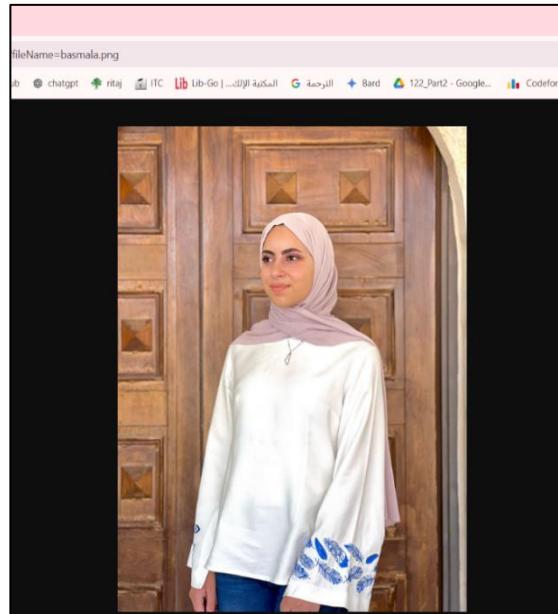
```
46
47     else:
48         # Check the file extension for redirection
49         file_extension = file_name.split('.')[ -1 ].lower()
50         if file_extension in [ 'jpg', 'jpeg', 'png' ]:
51             # Redirect to Google image search
52             redirect_url = f" https://www.google.com/search?q={file_name.replace(' ', '+')}&tbo=isch"
53             connection.sendall( f"HTTP/1.1 307 Temporary Redirect\r\nLocation: {redirect_url}\r\n\r\n".encode() )
54         elif file_extension in [ 'mp4', 'avi', 'mkv', 'mov' ]:
55             # Redirect to YouTube search
56             redirect_url = f" https://www.youtube.com/results?search_query={file_name.replace(' ', '+')}"
57             connection.sendall( f"HTTP/1.1 307 Temporary Redirect\r\nLocation: {redirect_url}\r\n\r\n".encode() )
58         else:
59             # If file type is not recognized
60             error_message = (
61                 "<html><head><meta charset='UTF-8'><title>Unsupported file type</title></head>"
62                 "<body><h1>Unsupported file type. Please enter a valid file name.</h1></body></html>"
63             )
64             connection.sendall(
65                 b"HTTP/1.1 400 Bad Request\r\nContent-Type: text/html\r\n\r\n" + error_message.encode() )
66         continue
```

Figure 24 Error message

If we search for something that doesn't exist neither in the file nor on the internet, it will return a 404 error page.

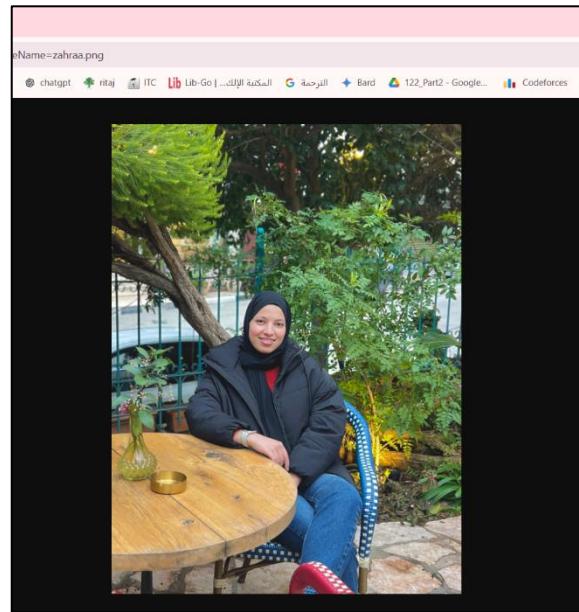
When search for basmala.png

A screenshot of a web browser window titled "Supporting Material". The URL is 127.0.0.1:5698/supporting_material_en.html. The page has a dark blue header with the word "Request" in white. Below it is a light gray section titled "Request an Image or Video". It contains the text "Enter the name of the image or video you want. If unavailable, you will be redirected to a search engine." followed by a form field labeled "File name (include extension)" with the value "basmala.png" and a "Submit" button. A red box highlights the input field.



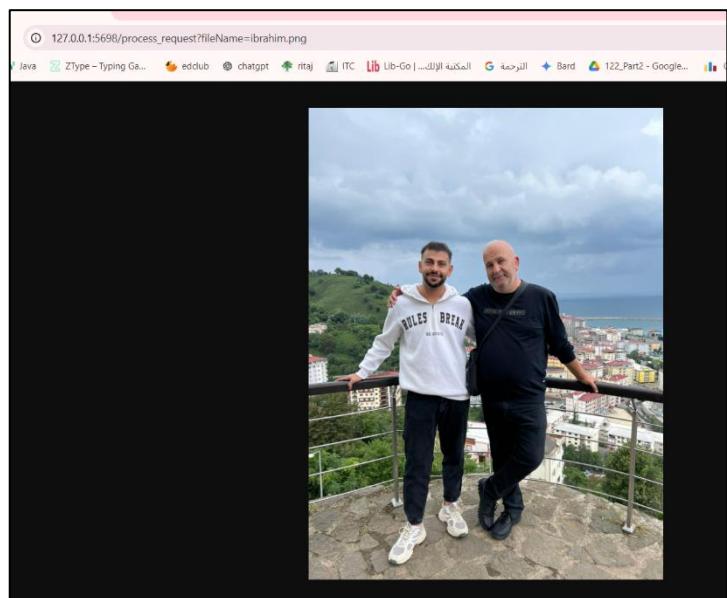
When search for Zahraa.png

A screenshot of a web browser window titled "Supporting Material". The URL is 127.0.0.1:5698/supporting_material_en.html. The page has a dark blue header with the word "Request Sup" (partially visible). Below it is a light gray section titled "Request an Image or Video". It contains the text "Enter the name of the image or video you want. If unavailable, you will be redirected to a search engine." followed by a form field labeled "File name (include extension)" with the value "zahraa.png" and a "Submit" button. A red box highlights the input field. At the bottom right, there is a link "Back to...".



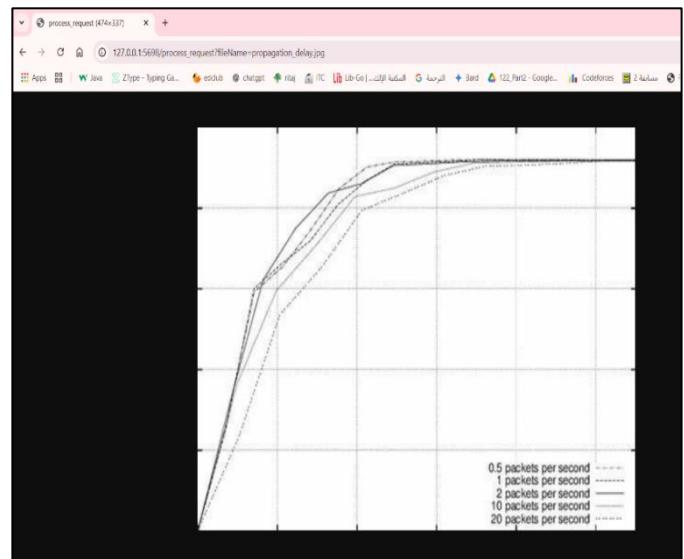
When search for ibraheem.png

The screenshot shows a web browser window titled "Supporting Material". The URL is 127.0.0.1:5698/supporting_material_en.html. The page content is a "Request Support" form. It includes a title "Request Support", a section "Request an Image or Video" with instructions, and a file input field labeled "File name (include extension)" containing "ibraheem.png". A red box highlights the file input field. A "Submit" button is next to it.

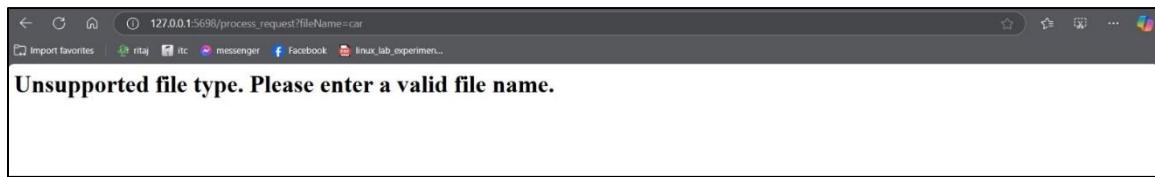
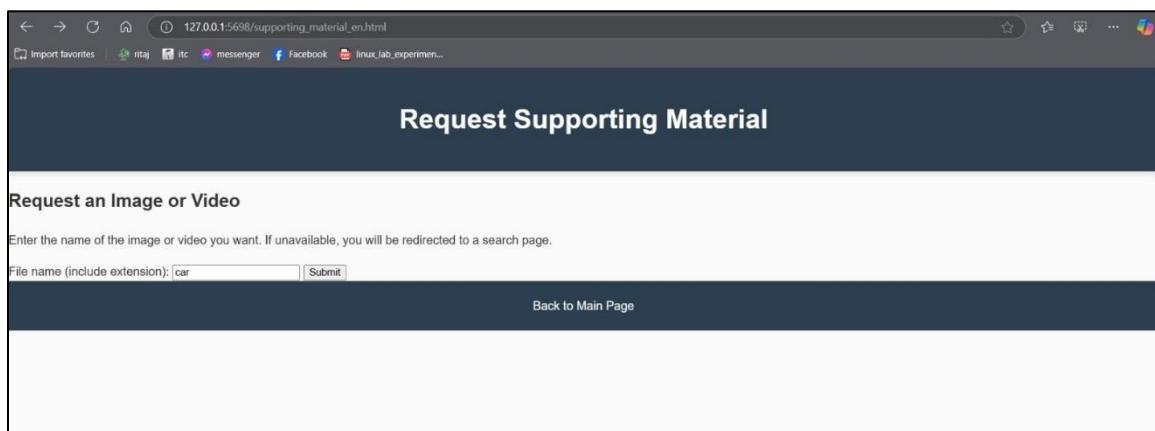


When search for propagation_delay.jpg

The screenshot shows a web browser window titled "process.request [474x337]". The URL is 127.0.0.1:5698/process_request?fileName=propagation_delay.jpg. The page content is a "Request Support" form. It includes a title "Request Support", a section "Request an Image or Video" with instructions, and a file input field labeled "File name (include extension)" containing "propagation_delay.jpg". A red box highlights the file input field. A "Submit" button is next to it.

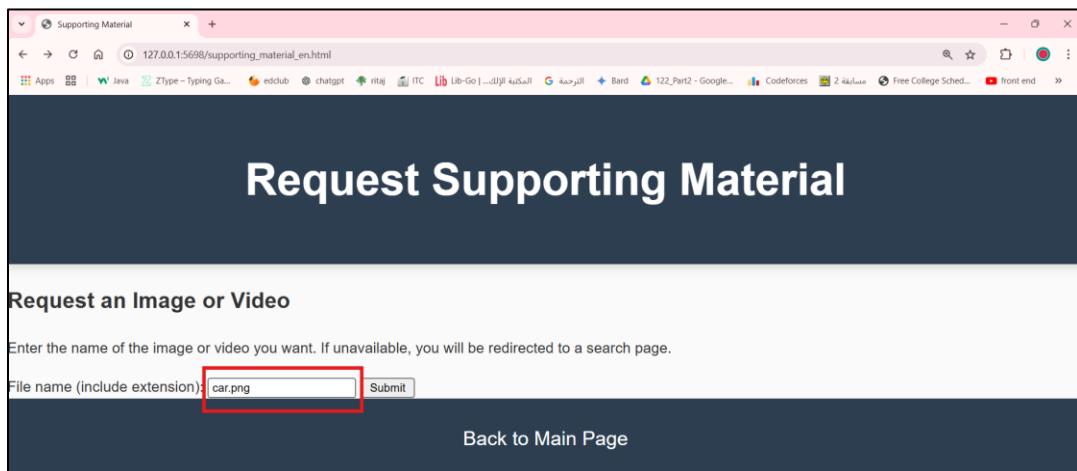


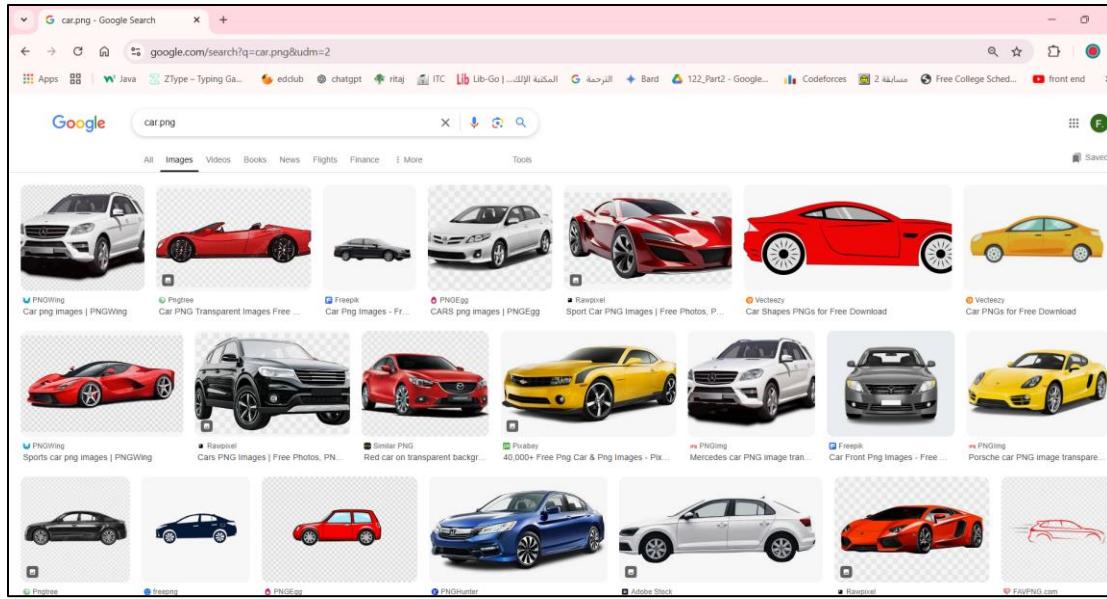
When search for “Car”



When searching for a car, it gave 'unsupported file type' because not including the file format. Instead of writing just car, we should write car.png or car.jpg.

When we search for car.png





When search for cat.mp4

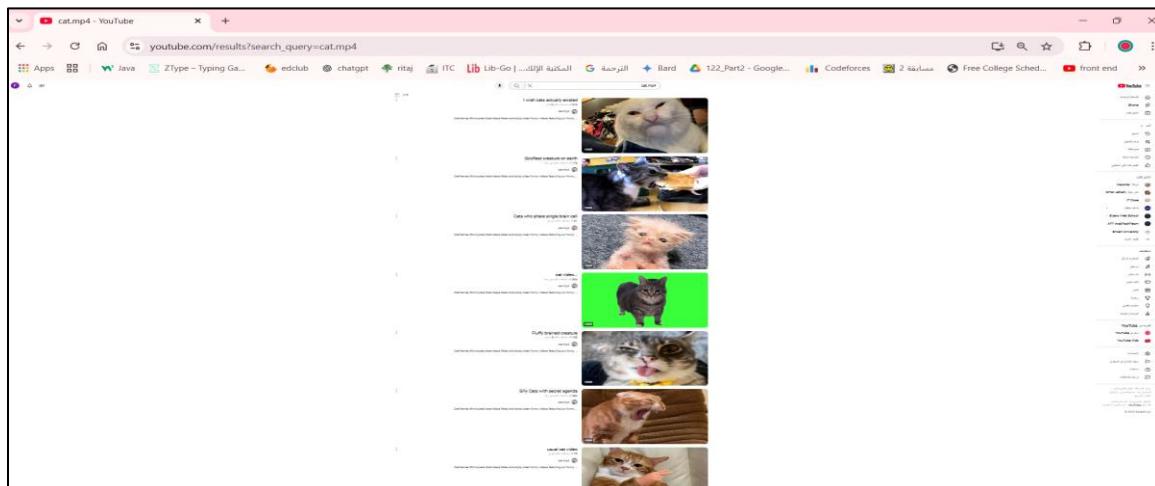
Request Supporting Material

Request an Image or Video

Enter the name of the image or video you want. If unavailable, you will be redirected to a search page.

File name (include extension)

[Back to Main Page](#)



3-When the request /ar the browser displays:

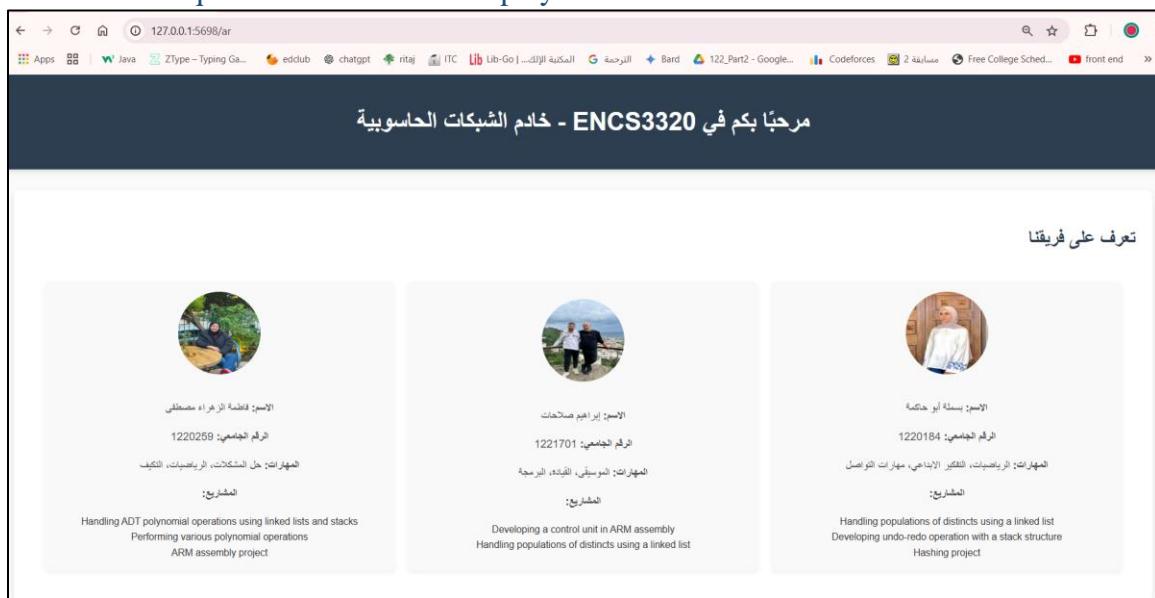


Figure 25 /ar request

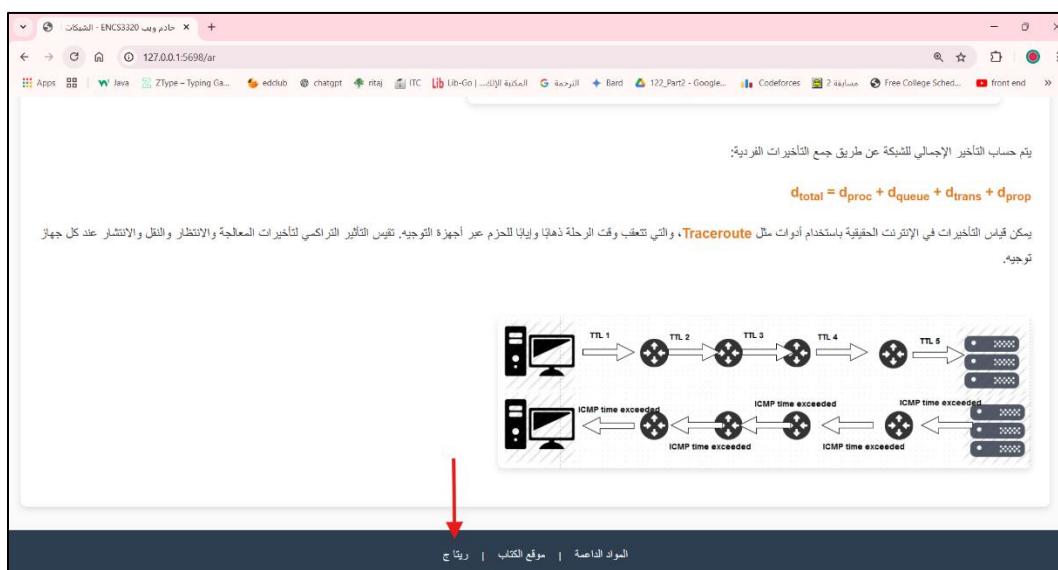
The HTTP request in command line :

```
The server is ready to receive
IP: 127.0.0.1, Port: 55335
GET /traceroute.jpg HTTP/1.1
Host: 127.0.0.1:5698
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 Edg/131.0.0.0
sec-ch-ua: "Microsoft Edge";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?
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://127.0.0.1:5698/ar
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9

Connection closed
The server is ready to receive
```

Figure 26 The HTTP /ar request in command line

Requeste Ritaj



Request Textbook website

يتم حساب التأثير الإجمالي للشبكة عن طريق جمع التأخيرات الفردية:

$$d_{\text{total}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

يمكن قياس التأخيرات في الإنترن트 الحقيقة باستخدام أدوات مثل **Traceroute**, والتي تتبع وقت الرحلة ذهاباً وإياباً للحزم عبر أجهزة التوجيه. تقيس التأثير التراكمي للتأخيرات المعالجة والانتظار والتقلل والانتشار عند كل جهاز توجيه.

Welcome to the authors' website for the textbook, Computer Networking: A Top Down Approach (Pearson). The 8th edition of our textbook has been published in the spring of 2020 - find out what's new in the 8th edition. From this page here (check out the menu at the top of the page), you can find resources and information of interest to students, teachers, and readers alike.

Since the publication of the first edition 21 years ago, the book has been adopted at many hundreds of colleges and universities, translated into 14 languages, and used by literally millions of students and practitioners worldwide. We've been overwhelmed by the positive response.

This textbook is for a first course on computer networking. It has been used in computer science and electrical engineering departments, information systems and informatics departments, in business schools, and elsewhere - at both the undergraduate and graduate levels. It should also be of interest to practitioners in industry as well. Find out more about the textbook [here](#).

You can't buy a hard copy of the 8th edition, but instead can rent (and then choose/pay to keep the hardcopy if you want a hard copy book). You can [rent a copy or subscribe to Pearson+](#) from our publisher, or [rent a hard copy or purchase a Kindle version from Amazon](#), or [rent a hard copy from VitalSource](#). The ISBNs are: Print rental: 9780136681557, Pearson+ access: 9780135928615.

We gratefully acknowledge the programming and problem design work of John Broderick (UMass '21), which has really helped to substantially improve this site.

Copyright © 2010-2024 J.F. Kurose, K.W. Ross
Comments welcome and appreciated: kurose@cs.umass.edu

Request supporting material

يتم حساب التأخير الإجمالي للشبكة عن طريق جمع التأخيرات الفردية:

$$d_{\text{total}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

يمكن قياس التأخيرات في الإنترن트 الحقيقية باستخدام أدوات مثل **Traceroute**, والتي تتبع وفت الرحلة ذهاباً وإياباً للحزم عبر أجهزة التوجيه. تقيس التأثير التراكمي للتأخيرات المعالجة والانتظار والتقليل والانتشار عند كل جهاز توجيه.

الموقع الداعمة | موقع الكتاب | زيارة

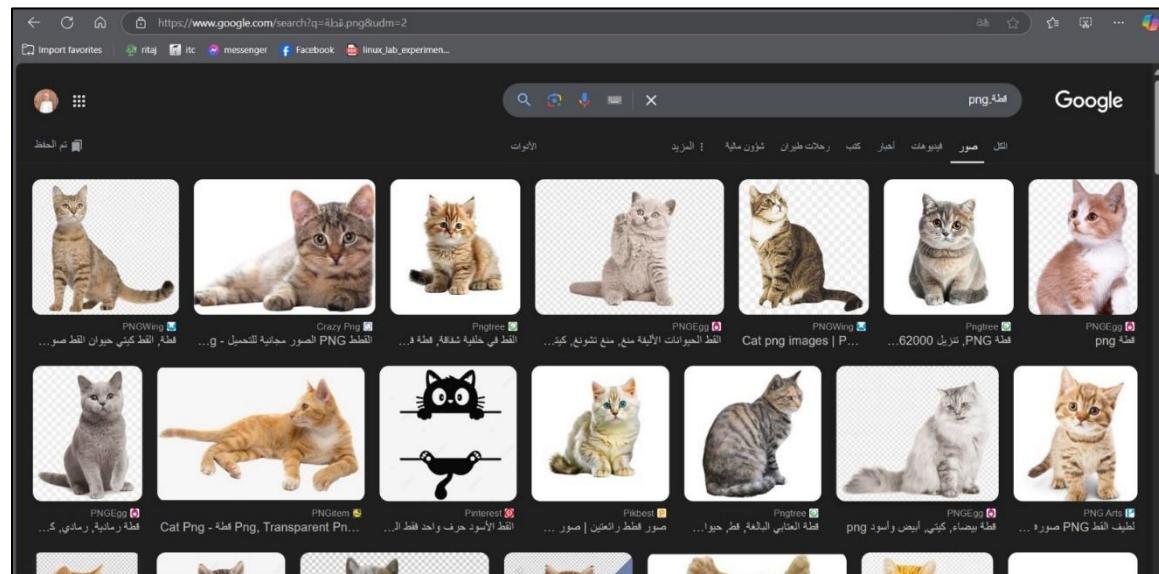
The HTTP request in command line

```
The server is ready to receive
IP: 127.0.0.1, Port: 58326
GET /style.css HTTP/1.1
Host: 127.0.0.1:5698
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 Edg/131.0.0.0
sec-ch-ua: "Microsoft Edge";v="131", "Chromium";v="131", "Not_A_Brand";v="24"
sec-ch-ua-mobile: ?
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: http://127.0.0.1:5698/supporting_material_ar.html
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9

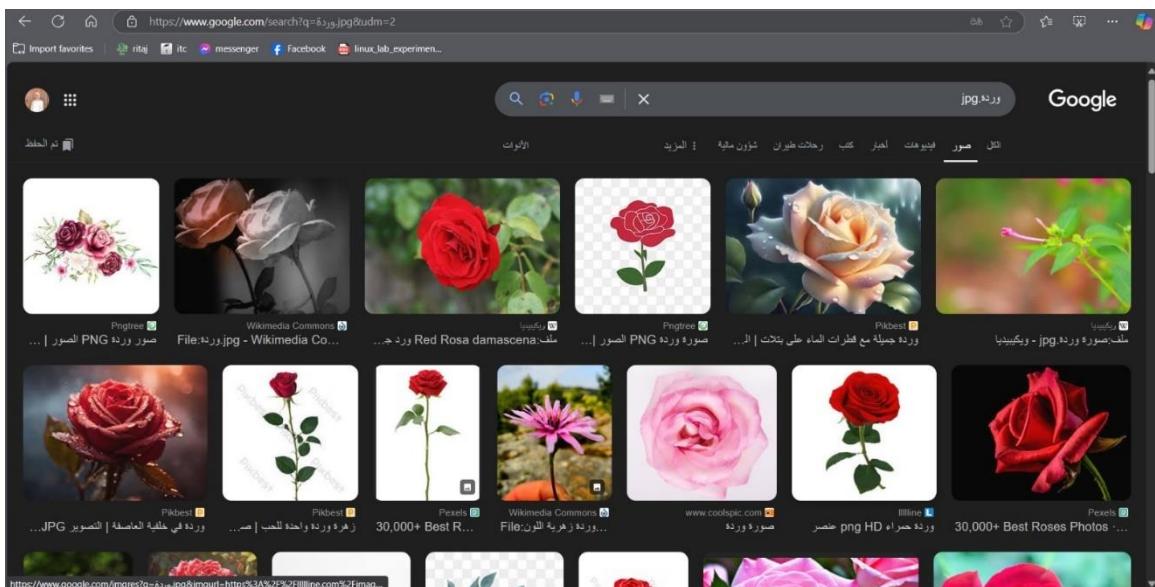
Connection closed
The server is ready to receive
```

Figure 27 The HTTP/supporting_material_ar.html request in command line

When search for “قطة.png”

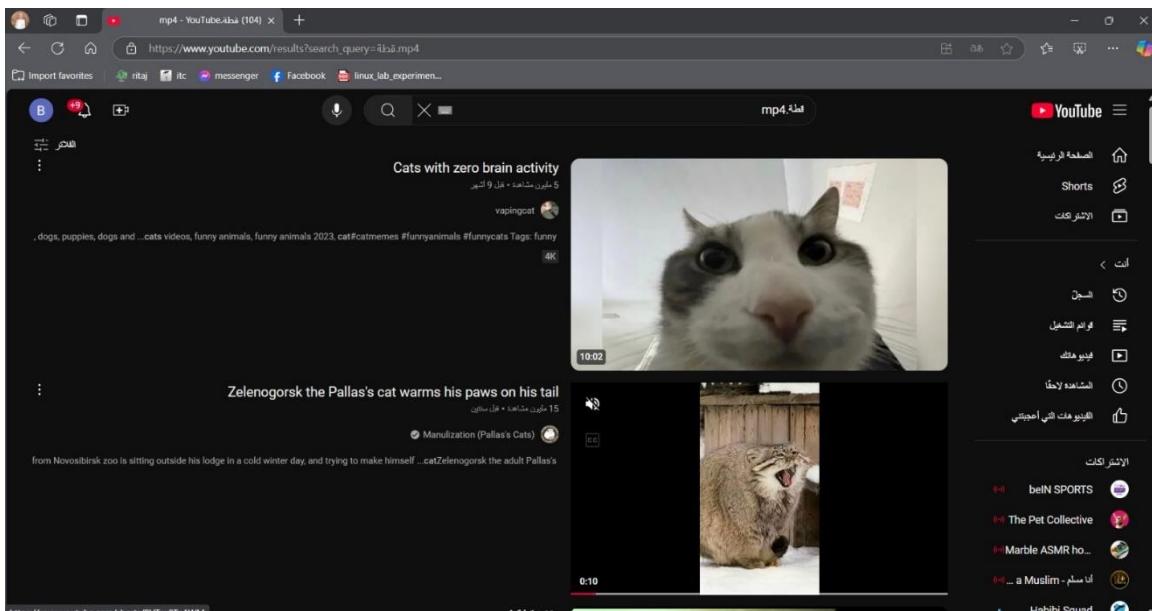


When search for "وردة.jpg"



When search for "قطة.mp4"





When the request /.css the browser displays

```

/* General Styles */
body {
    font-family: 'Arial', sans-serif;
    background-color: #fafafa;
    color: #333;
    margin: 0;
    padding: 0;
    line-height: 1.6;
}

/* Header Styling */
header {
    background-color: #2c3e50;
    color: #fff;
    text-align: center;
    padding: 40px 0;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

    header h1 {
        font-size: 36px;
        margin: 0;
    }

/* Team Section Styling */
.team {
    margin: 30px 15px;
    padding: 25px;
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

    .team h2 {
        font-size: 28px;
        color: #2c3e50;
        margin-bottom: 20px;
    }

table {
    width: 100%;
    border-collapse: separate;
    border-spacing: 20px;
}

```

The HTTP request in command line

```
The server is ready to receive
IP: 127.0.0.1, Port: 55361
GET /.css HTTP/1.1
Host: 127.0.0.1:5698
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";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 Edg/131.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9

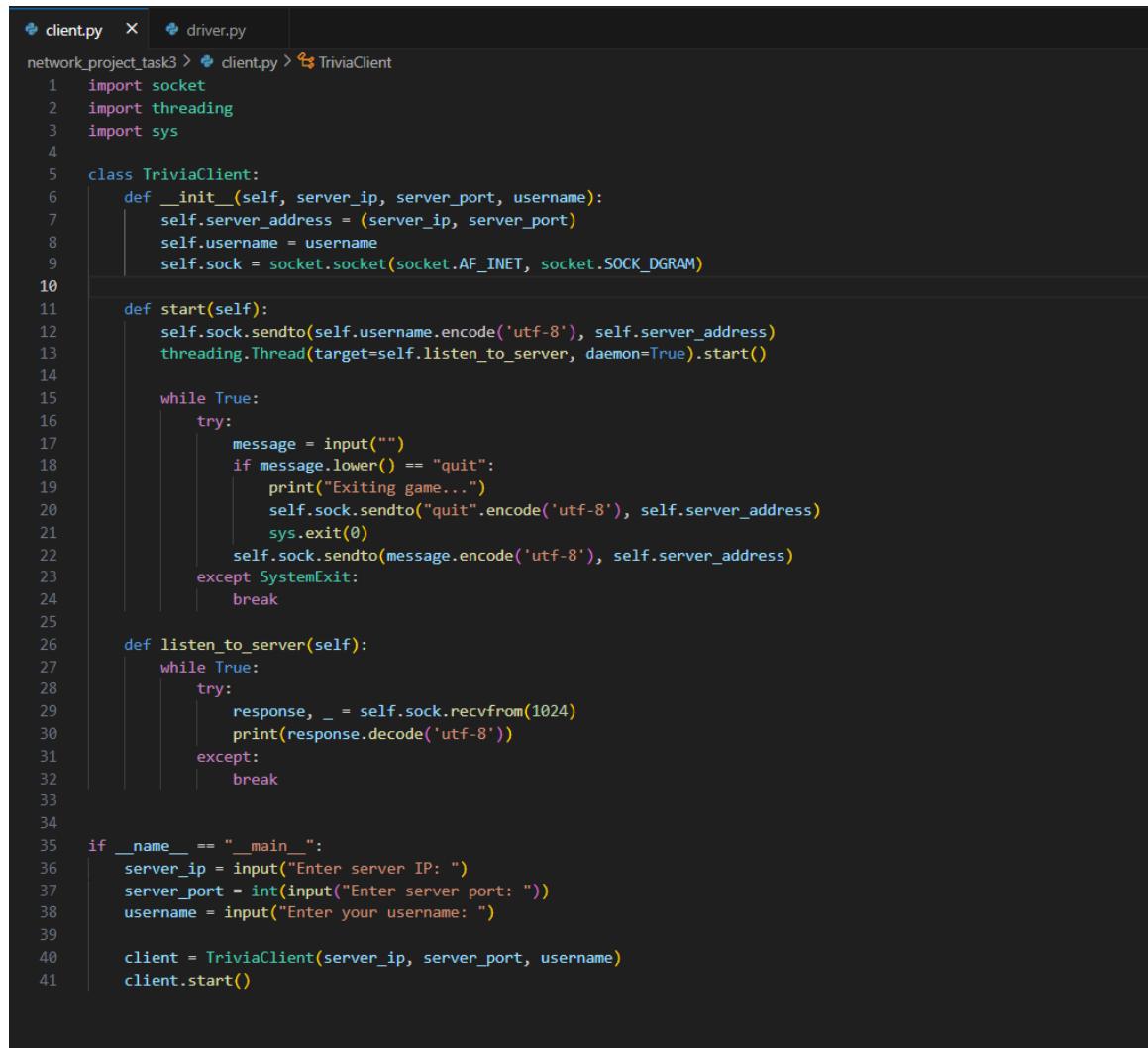
Connection closed
```

Figure 28 HTTP /.css request in command line

Task 3 UDP Client-Server Trivia Game Using Socket Programming

In the trivia competition, each question has a specific time limit for answering. The game requires at least two participants. For example, if there are three players, the first player to answer correctly earns points based on the remaining time: the point value is determined by subtracting the time they took to answer from the total time limit. The second player to answer correctly will receive the remaining points after the first player's time is deducted, and the third player will get the last portion of the points based on their response time. If any player answers incorrectly or does not answer at all, they will receive 0 points for that question. The game continues through multiple rounds and ends when the number of participants drops below the required minimum.

client code :



```
client.py  x  driver.py
network_project_task3 > client.py > TriviaClient
1  import socket
2  import threading
3  import sys
4
5  class TriviaClient:
6      def __init__(self, server_ip, server_port, username):
7          self.server_address = (server_ip, server_port)
8          self.username = username
9          self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
10
11     def start(self):
12         self.sock.sendto(self.username.encode('utf-8'), self.server_address)
13         threading.Thread(target=self.listen_to_server, daemon=True).start()
14
15     while True:
16         try:
17             message = input("")
18             if message.lower() == "quit":
19                 print("Exiting game...")
20                 self.sock.sendto("quit".encode('utf-8'), self.server_address)
21                 sys.exit(0)
22             self.sock.sendto(message.encode('utf-8'), self.server_address)
23         except SystemExit:
24             break
25
26     def listen_to_server(self):
27         while True:
28             try:
29                 response, _ = self.sock.recvfrom(1024)
30                 print(response.decode('utf-8'))
31             except:
32                 break
33
34
35 if __name__ == "__main__":
36     server_ip = input("Enter server IP: ")
37     server_port = int(input("Enter server port: "))
38     username = input("Enter your username: ")
39
40     client = TriviaClient(server_ip, server_port, username)
41     client.start()
```

Figure 29 client code

This Python script implements a UDP-based trivia game client that communicates with a trivia server. The client allows users to enter their username, connect to the server, and send messages (such as trivia answers or commands) while listening for responses from the server in real time. The client uses the socket and threading modules to handle communication and concurrency. The TriviaClient class is the core of the client, managing the connection and interaction with the server.

The TriviaClient class has several important attributes and methods. In the `__init__` method, it initializes the connection details, including the server's IP address and port as a tuple (`server_address`), and stores the user's chosen username. A UDP socket is created using `socket.socket(socket.AF_INET, socket.SOCK_DGRAM)`, which establishes a connectionless communication channel between the client and the server. UDP is chosen because it is lightweight and suitable for real-time applications like a trivia game, where speed is prioritized over guaranteed delivery.

The start method is the entry point of the client. It first sends the username to the server to identify the client. Then, it creates a new thread using `threading.Thread(target=self.listen_to_server, daemon=True)` to listen for responses from the server continuously in the background. The main thread waits for user input, which is sent to the server as a message. If the user types "quit", the client sends a "quit" message to the server and terminates the connection gracefully using `sys.exit(0)`.

The `listen_to_server` method runs in a separate thread and listens for incoming messages from the server. When a message is received, it decodes and prints the response to the console, allowing the user to see the server's replies, such as questions or results. If an error occurs (e.g., the server stops sending data), the loop breaks, and the client stops listening.

The important APIs and modules used in this script are:

`socket.socket()`: This is used to create a UDP socket for communication between the client and server.

`sendto()`: This method sends data to the specified server address over the socket.

`recvfrom()`: This method listens for incoming data from the server.

`threading.Thread`: This is used to create a separate thread to handle server communication, allowing the client to send messages while simultaneously receiving responses.

In summary, the client establishes a real-time communication channel with the trivia server using UDP sockets, handles user input, and listens for responses asynchronously in a background thread. This allows the trivia game to function interactively and efficiently.

Server code :

```
client.py  server.py  x
network_project_task3 > server.py > ...
1  import socket
2  import threading
3  import time
4  import random
5
6  # Predefined set of trivia questions and answers
7  QUESTIONS = [
8      ("What is the capital of France?", "Paris"),
9      ("Who wrote 'To Kill a Mockingbird?'", "Harper Lee"),
10     ("What is the smallest prime number?", "2"),
11     ("Which planet is known as the Red Planet?", "Mars"),
12     ("What is the chemical symbol for water?", "H2O"),
13     ("What color is the sky on a clear day?", "Blue"),
14     ("What is 5 + 3?", "8"),
15     ("Is the sun a star? (True/False)", "True"),
16     ("What is the opposite of 'hot'?", "Cold"),
17     ("Do fish live in water? (True/False)", "True"),
18     ("How many legs does a spider have?", "8"),
19     ("Is an elephant bigger than a mouse? (True/False)", "True"),
20     ("What fruit is known for having seeds on the outside?", "Strawberry"),
21     ("Is chocolate made from cocoa beans? (True/False)", "True"),
22     ("What is the first letter of the English alphabet?", "A"),
23     ("Do birds have feathers? (True/False)", "True"),
24     ("How many days are in a week?", "7"),
25     ("Is 2 + 2 equal to 5? (True/False)", "False"),
26     ("What shape has three sides?", "Triangle"),
27     ("Does the moon produce its own light? (True/False)", "False"),
28     ("What is 10 divided by 2?", "5"),
29     ("Is water wet? (True/False)", "True"),
30     ("What do cows drink?", "Water"),
31     ("Is fire cold? (True/False)", "False"),
32     ("What is the color of bananas?", "Yellow"),
33     ("Is Earth a planet? (True/False)", "True"),
34     ("What is 1 + 1?", "2"),
35     ("Is ice hot? (True/False)", "False"),
36     ("What color are most leaves?", "Green"),
37     ("Do cats bark? (True/False)", "False"),
38     ("Which animal is known as 'man's best friend'?", "Dog"),
39     ("Can humans breathe underwater without equipment? (True/False)", "False"),
40     ("What do bees make?", "Honey"),
41     ("Is the moon closer to Earth than the sun? (True/False)", "True"),
42     ("What is 6 - 4?", "2"),
43     ("Is snow white? (True/False)", "True"),
44     ("What do you call a baby dog?", "Puppy"),
45     ("Can fish fly? (True/False)", "False"),
46     ("What is the number after 9?", "10"),
47     ("Do apples grow on trees? (True/False)", "True"),
48     ("Is the ocean salty? (True/False)", "True"),
49     ("What is the color of the sun?", "Yellow"),
50     ("What is 3 x 3?", "9"),
51     ("Is chocolate sweet? (True/False)", "True"),
52     ("Do chickens lay eggs? (True/False)", "True"),
53     ("What animal says 'meow'?", "Cat"),
54     ("Is fire orange? (True/False)", "True"),
55     ("How many fingers does a human hand have?", "5"),
56     ("Can airplanes fly? (True/False)", "True"),
57 ]
58 ]
```

Figure 30 The question area

These are the predefined questions that will be used in the trivia competition. Each question has a corresponding correct answer. Each participant's score will be based on how quickly and accurately they answer the questions.

Note : The questions may appear more than once during the competition.

```

58
59     MIN_PLAYERS = 2
60     ROUND_DELAY = 10
61     ANSWER_TIME = 30
62
63
64     class TriviaServer:
65         def __init__(self, host='127.0.0.1', port=5689):
66             self.server_address = (host, port)
67             self.clients = {}
68             self.scores = {}
69             self.round_answers = {}
70             self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
71             self.sock.bind(self.server_address)
72             print(f"Server started on {host}:{port}")
73             self.current_question = None
74             self.current_question_time = None
75             self.running = True
76
77         def start(self):
78             threading.Thread(target=self.listen_for_clients, daemon=True).start()
79             print(f"Waiting for at least {MIN_PLAYERS} players to join...")
80             while len(self.clients) < MIN_PLAYERS:
81                 time.sleep(1)
82             print(f"{MIN_PLAYERS} players have joined. Starting the game!")
83             try:
84                 while self.running:
85                     if len(self.clients) >= MIN_PLAYERS:
86                         self.play_round()
87                     else:
88                         print("Not enough players. Exiting server... ")
89                         self.broadcast_message("Server is shutting down. Not enough players.")
90                         self.running = False
91                         self.sock.close()
92                         exit()
93             except KeyboardInterrupt:
94                 print("\nServer shutting down...")
95                 self.broadcast_message("Server is shutting down. Goodbye!")
96                 self.running = False
97                 self.sock.close()
98
99         def listen_for_clients(self):
100            while self.running:
101                try:
102                    message, client_address = self.sock.recvfrom(1024)
103                    message = message.decode('utf-8').strip()
104                    if client_address in self.clients:
105                        if message.lower() == "quit":
106                            self.remove_client(client_address, reason="Quit the game")
107                        else:
108                            threading.Thread(
109                                target=self.process_answer, args=(message, client_address), daemon=True
110                            ).start()
111                    elif message.lower() != "quit":
112                        threading.Thread(
113                            target=self.add_client, args=(message, client_address), daemon=True
114                        ).start()
115                except Exception as e:
116                    print(f"Error in listen_for_clients: {e}")
117
118            def add_client(self, message, client_address):
119                self.clients[client_address] = message
120                self.scores[client_address] = 0
121                print(f"New client joined: {client_address} ({message})")
122                self.sock.sendto("Welcome to the trivia game!".encode('utf-8'), client_address)
123                self.broadcast_message(f"{message} has joined the game!")
124
125            def process_answer(self, message, client_address):
126                if self.round_answers.get(client_address): # Check if the client has already answered
127                    self.sock.sendto("You have already answered this question.".encode('utf-8'), client_address)
128                    return
129
130                client_answer = message
131                print(f"Answer received from {self.clients[client_address]} {client_address}: {client_answer}")
132
133                if self.current_question and (time.time() - self.current_question_time <= ANSWER_TIME):
134                    if client_answer.lower() == self.current_question[1].lower():
135                        self.scores[client_address] += int(ANSWER_TIME - (time.time() - self.current_question_time))
136                        self.sock.sendto("Correct!".encode('utf-8'), client_address)
137                    else:
138                        self.sock.sendto("Wrong answer.".encode('utf-8'), client_address)
139
140                self.round_answers[client_address] = True # Mark client as having answered
141

```

```

142     def play_round(self):
143         print("Starting a new round...")
144         self.broadcast_message("A new round is starting soon!")
145         time.sleep(ROUND_DELAY)
146
147         # Select a random set of questions for this round
148         questions = random.sample(QUESTIONS, 4)
149
150         # Track players who didn't answer any question in this round
151         unresponsive_clients = set(self.clients.keys())
152
153         for idx, (question, answer) in enumerate(questions, start=1):
154             # Reset round_answers at the start of each question
155             self.round_answers = {client: False for client in self.clients}
156
157             # Set the current question and start the timer
158             self.current_question = (question, answer)
159             self.current_question_time = time.time()
160
161             # Broadcast the question to all clients
162             self.broadcast_message(f"Question {idx}: {question}")
163             print(f"Broadcasted question {idx}: {question}")
164
165             # Wait for the answer time to elapse
166             time.sleep(ANSWER_TIME)
167
168             # Remove players who responded to the question from the unresponsive set
169             for client, answered in self.round_answers.items():
170                 if answered:
171                     unresponsive_clients.discard(client)
172
173             # Broadcast the correct answer and display scores
174             self.broadcast_message(f"\nThe correct answer was: {answer}")
175             print(f"The correct answer was: {answer}")
176             self.display_scores()
177
178             # If there are not enough players left after a question, stop the game
179             if len(self.clients) < MIN_PLAYERS:
180                 print("Not enough players to continue. Server shutting down.")
181                 self.broadcast_message("Not enough players to continue. Server shutting down.")
182                 self.running = False
183                 self.sock.close()
184                 exit()
185
186             # After all questions, kick unresponsive clients
187             for client in unresponsive_clients:
188                 self.remove_client(client, reason="Did not respond during the entire round")
189
190     def remove_client(self, client_address, reason="Quit the game"):
191         if client_address in self.clients:
192             client_name = self.clients[client_address]
193             print(f"Removing client {client_name} ({client_address}): {reason}")
194             del self.clients[client_address]
195             del self.scores[client_address]
196             if client_address in self.round_answers:
197                 del self.round_answers[client_address]
198             self.sock.sendto("You have been removed from the game.".encode('utf-8'), client_address)
199             self.broadcast_message(f"{client_name} has left the game!")
200
201     def broadcast_message(self, message):
202         for client in self.clients.keys():
203             self.sock.sendto(message.encode('utf-8'), client)
204
205     def display_scores(self):
206         leaderboard = "\nLeaderboard:\n"
207         for client, score in sorted(self.scores.items(), key=lambda x: x[1], reverse=True):
208             leaderboard += f"{self.clients[client]}: {score} points\n"
209         print(leaderboard)
210         self.broadcast_message(leaderboard)
211
212
213     if __name__ == "__main__":
214         server = TriviaServer()
215         server.start()

```

Figure 31 Server code

This Python code implements a trivia game server that uses the UDP protocol to handle communication between multiple clients. The server facilitates the trivia game by accepting player connections, broadcasting questions, receiving answers, and updating scores. It uses the socket, threading, and time modules to handle networking, concurrent player interactions, and timing functionality. The trivia questions are predefined in a list, and the server randomly selects a set of questions for each round. It ensures that at least two players are connected before starting the game and enforces time limits for answering questions.

The server's core functionality is encapsulated in the TriviaServer class. This class initializes the server's address and binds it to a UDP socket (socket.socket(socket.AF_INET, socket.SOCK_DGRAM)) on the specified host and port. It then listens for incoming connections from clients and manages the trivia game logic. A separate thread (threading.Thread(target=self.listen_for_clients, daemon=True)) is used to handle incoming messages from clients, allowing the server to receive and process data asynchronously. The server handles a variety of client commands, including new player registration, answers to questions, and player disconnections.

The trivia game is structured into rounds, where each round consists of multiple questions (randomly selected from a predefined set). The play_round method broadcasts a question to all connected clients, waits for answers for a specified amount of time (ANSWER_TIME), and then evaluates the answers. If a player answers correctly, they earn points based on how quickly they responded. The server also keeps track of the players' scores in a dictionary (self.scores). If any players do not respond within the allotted time, they are considered unresponsive and may be removed from the game. The server maintains a leaderboard that is periodically updated and sent to all players.

In terms of key APIs and modules used:

socket.socket(): This is used to create a UDP socket for communication with the clients. The socket is set up with AF_INET (IPv4) and SOCK_DGRAM (UDP) to establish the connectionless communication.

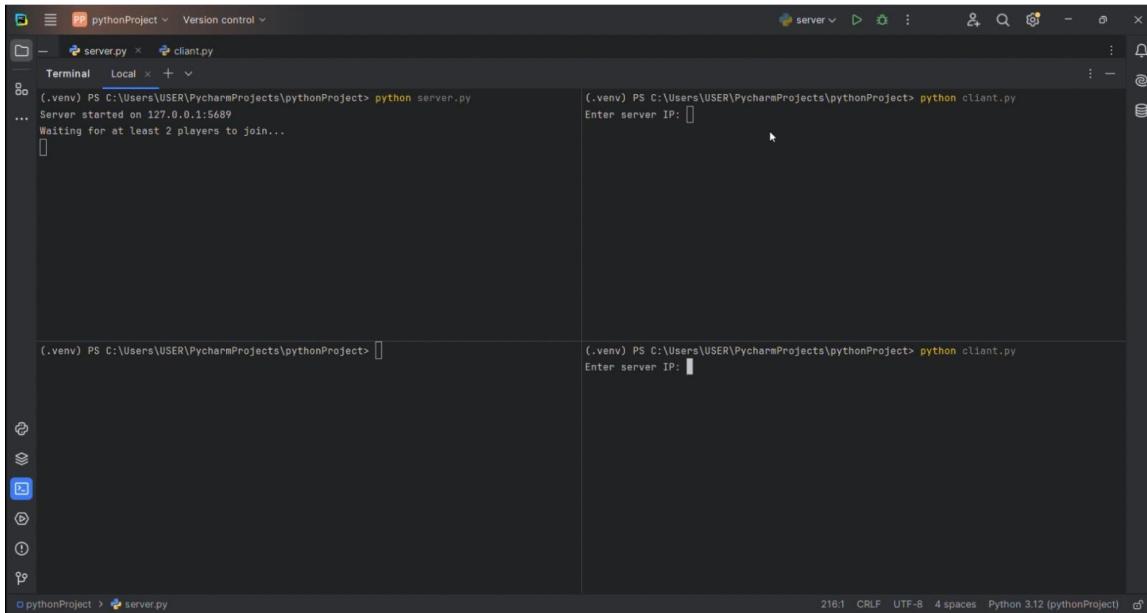
recvfrom() and sendto(): These methods are used for receiving data from and sending data to the clients, respectively. The recvfrom() method is used to listen for incoming messages from any client, while sendto() is used to send messages to specific clients.

threading.Thread(): This is used to handle concurrent operations, such as listening for client messages while simultaneously processing game logic. Each client's actions are handled in separate threads, allowing the server to scale efficiently.

time.sleep(): This is used for introducing delays, such as waiting between rounds or giving players time to answer questions. It also helps control the pacing of the game, such as waiting for players to respond within the allowed time frame.

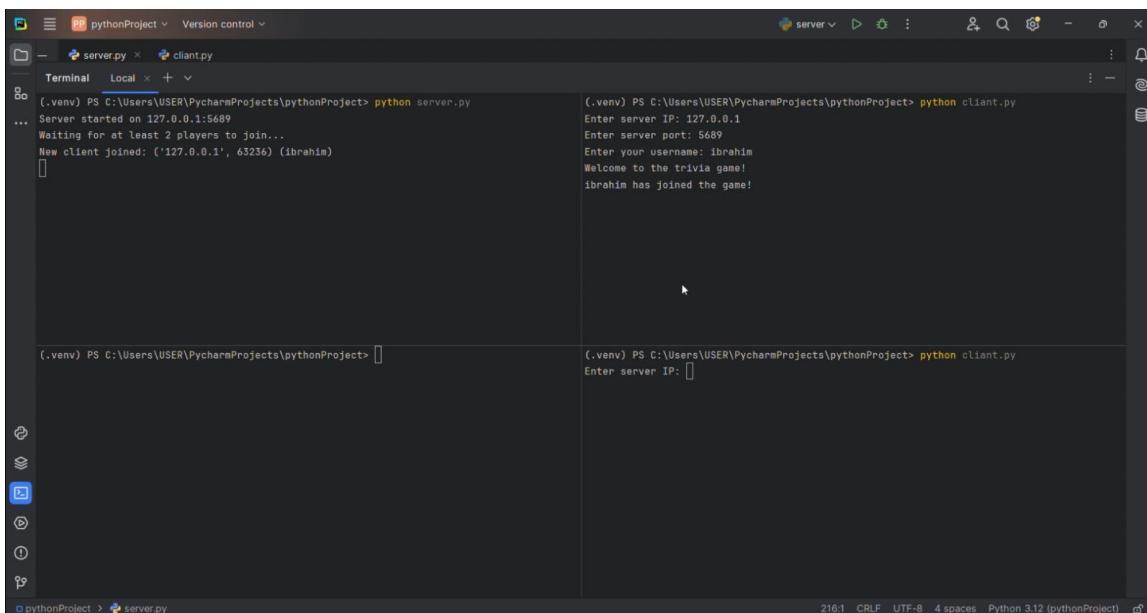
Overall, this trivia game server provides an interactive and scalable solution to host a multiplayer trivia game. It leverages asynchronous communication to handle multiple clients efficiently and uses simple game mechanics to create an engaging user experience.

Traviria Game Example



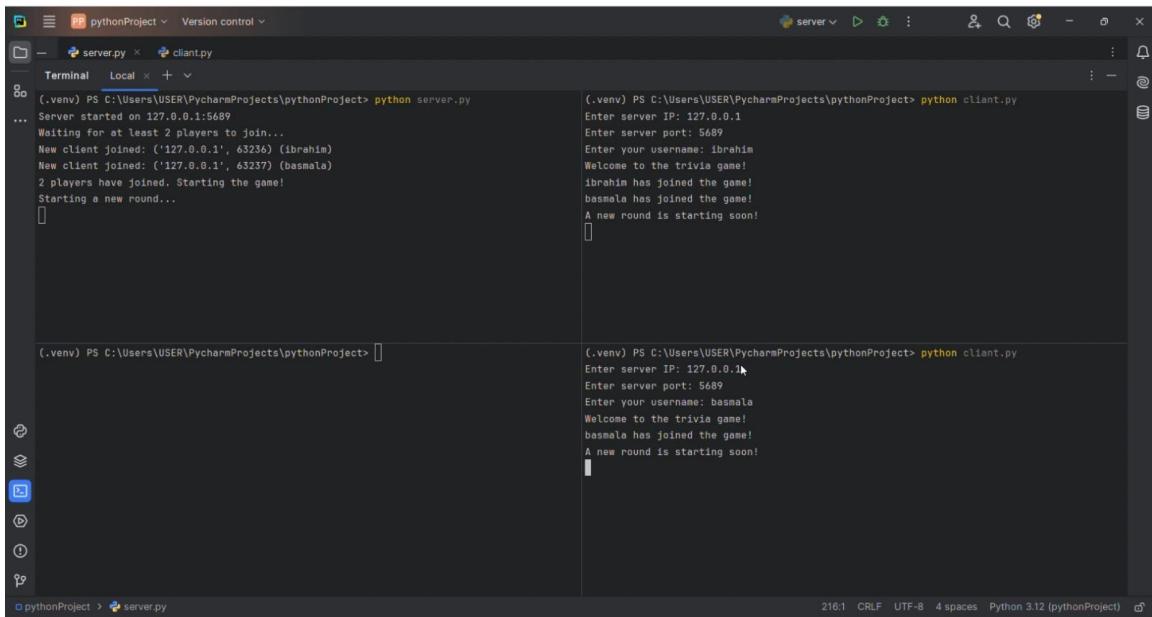
```
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python server.py
...
Server started on 127.0.0.1:5689
Waiting for at least 2 players to join...
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python client.py
Enter server IP:
```

The server is running, but no clients are connected yet.



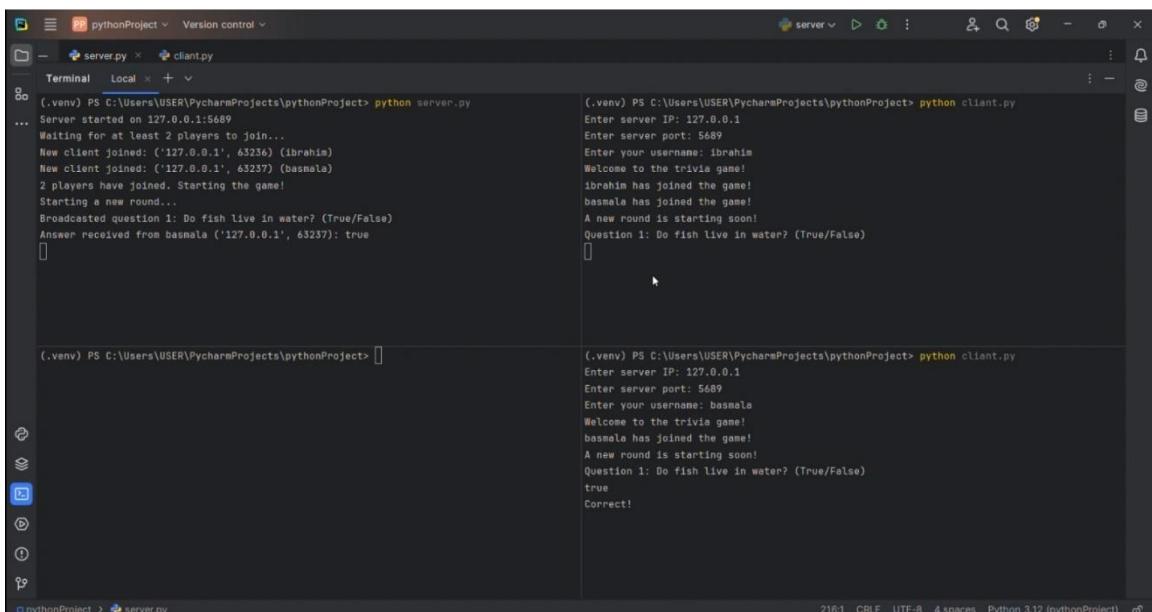
```
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python server.py
...
Server started on 127.0.0.1:5689
Waiting for at least 2 players to join...
New client joined: ('127.0.0.1', 63236) (ibrahim)
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python client.py
Enter server IP: 127.0.0.1
Enter server port: 5689
Enter your username: ibrahim
Welcome to the trivia game!
ibrahim has joined the game!
```

The first client has joined, The game will start once at least two clients are connected.



```
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python server.py
...
Server started on 127.0.0.1:5689
Waiting for at least 2 players to join...
New client joined: ('127.0.0.1', 65236) (ibrahim)
New client joined: ('127.0.0.1', 65237) (basmala)
2 players have joined. Starting the game!
Starting a new round...
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python client.py
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject>
Enter server IP: 127.0.0.1
Enter server port: 5689
Enter your username: ibrahim
Welcome to the trivia game!
ibrahim has joined the game!
basmala has joined the game!
A new round is starting soon!
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python client.py
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject>
Enter server IP: 127.0.0.1
Enter server port: 5689
Enter your username: basmala
Welcome to the trivia game!
basmala has joined the game!
A new round is starting soon!
```

The game started when the second client joined, as the minimum number of players was met.



```
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python server.py
...
Server started on 127.0.0.1:5689
Waiting for at least 2 players to join...
New client joined: ('127.0.0.1', 65236) (ibrahim)
New client joined: ('127.0.0.1', 65237) (basmala)
2 players have joined. Starting the game!
Starting a new round...
Broadcasted question 1: Do fish live in water? (True/False)
Answer received from basmala ('127.0.0.1', 65237): true
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python client.py
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject>
Enter server IP: 127.0.0.1
Enter server port: 5689
Enter your username: ibrahim
Welcome to the trivia game!
ibrahim has joined the game!
basmala has joined the game!
A new round is starting soon!
Question 1: Do fish live in water? (True/False)
true
Correct!
```

The first question was sent to all clients. Basmalah answered correctly before Ibrahim. Therefore, she earned more points.

The screenshot shows a PyCharm interface with two terminal windows. The left window is for the server (server.py) and the right window is for a client (client.py). The server logs show two clients joining: 'ibrahim' and 'basmala'. It then starts a new round and asks a question about fish living in water. Both clients respond with 'true', which is marked as correct. The server calculates points based on time and updates the leaderboard. The client then answers the second question correctly, increasing its score.

```

pythonProject Version control
server.py client.py
Terminal Local + v
...
New client joined: ('127.0.0.1', 63236) (ibrahim)
...
New client joined: ('127.0.0.1', 63237) (basmala)
2 players have joined. Starting the game!
Starting a new round...
Broadcasted question 1: Do fish live in water? (True/False)
Answer received from basmala ('127.0.0.1', 63237): true
Answer received from ibrahim ('127.0.0.1', 63236): true
The correct answer was: True

Leaderboard:
basmala: 22 points
ibrahim: 18 points

Broadcasted question 2: What is the capital of France?
[]

(.venv) PS C:\Users\USER\PycharmProjects\pythonProject>

```

```

ibrahim has joined the game!
basmala has joined the game!
A new round is starting soon!
Question 1: Do fish live in water? (True/False)
true
Correct!

The correct answer was: True

Leaderboard:
basmala: 22 points
ibrahim: 18 points

Question 2: What is the capital of France?
[]

Welcome to the trivia game!
basmala has joined the game!
A new round is starting soon!
Question 1: Do fish live in water? (True/False)
true
Correct!

The correct answer was: True

Leaderboard:
basmala: 22 points
ibrahim: 18 points

Question 2: What is the capital of France?
[]

216:1 CRLF UTF-8 4 spaces Python 3.12 (pythonProject)

```

The points are calculated based on the remaining time for the question, which is:

Question time – time taken by the client to answer .

So, the faster a client answers, the higher their score.

This screenshot continues the game session. The server asks a question about the capital of France. Ibrahim answers quickly with 'paris', which is marked as correct and increases his score to 36. Basmalah takes longer and answers with 'h', which is marked as a wrong answer. The server then asks another question about the number of days in a week, and both clients respond with '7', which is marked as correct. Ibrahim's score remains at 36, while Basmalah's score increases to 22.

```

pythonProject Version control
server.py client.py
Terminal Local + v
...
Leaderboard:
...
basmala: 22 points
ibrahim: 18 points

Broadcasted question 2: What is the capital of France?
Answer received from ibrahim ('127.0.0.1', 63236): paris
Answer received from basmala ('127.0.0.1', 63237): h
The correct answer was: Paris

Leaderboard:
ibrahim: 36 points
basmala: 22 points

Broadcasted question 3: How many days are in a week?
[]

(.venv) PS C:\Users\USER\PycharmProjects\pythonProject>

```

```

basmala: 22 points
ibrahim: 18 points

Question 2: What is the capital of France?
paris
Correct!

The correct answer was: Paris

Leaderboard:
ibrahim: 36 points
basmala: 22 points

Question 3: How many days are in a week?
[]

basmala: 22 points
ibrahim: 18 points

Question 2: What is the capital of France?
h
Wrong answer.

The correct answer was: Paris

Leaderboard:
ibrahim: 36 points
basmala: 22 points

Question 3: How many days are in a week?
[]

216:1 CRLF UTF-8 4 spaces Python 3.12 (pythonProject)

```

For the second question, Ibrahim answered correctly while Basmalah answered incorrectly. As a result, Basmalah's score still the same, but Ibrahim's score increased, making him the first.

```
pythonProject -- Version control
server v D S : + - @
pythonProject > server.py
ibrahim: 36 points
basmala: 22 points
Question 3: How many days are in a week?
7
Correct!
The correct answer was: 7
Leaderboard:
ibrahim: 50 points
basmala: 33 points
Question 4: Is the ocean salty? (True/False)
[client.py]
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python client.py
Enter server IP: 127.0.0.1
```

For the third question, the third client hasn't joined yet but is about to join in the middle of the game.

```
pythonProject -- Version control
server v D S : + - @
pythonProject > server.py
ibrahim: 36 points
basmala: 22 points
Question 3: How many days are in a week?
7
Correct!
The correct answer was: 7
Leaderboard:
ibrahim: 50 points
basmala: 33 points
Question 4: Is the ocean salty? (True/False)
zahraa has joined the game!
[client.py]
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject> python client.py
Enter server IP: 127.0.0.1
Enter server port: 5689
Enter your username: zahraa
Welcome to the trivia game!
zahraa has joined the game!
```

The third client has joined, and a message was sent to all clients notifying them of the new arrival.

```

pythonProject > Terminal Local + v
server.py x client.py
Terminal Local + v
80 basmala: 33 points
...
Broadcasted question 4: Is the ocean salty? (True/False)
New client joined: ('127.0.0.1', 62938) (zahraa)
Answer received from basmala ('127.0.0.1', 63237): false
Answer received from ibrahim ('127.0.0.1', 63236): false
The correct answer was: True

Leaderboard:
ibrahim: 50 points
basmala: 33 points
zahraa: 0 points

Starting a new round...
[redacted]
Enter server IP: 127.0.0.1
Enter server port: 5689
Enter your username: zahraa
Welcome to the trivia game!
zahraa has joined the game!
The correct answer was: True

Leaderboard:
ibrahim: 50 points
basmala: 33 points
zahraa: 0 points

zahraa: 0 points
A new round is starting soon!
[redacted]
Question 4: Is the ocean salty? (True/False)
zahraa has joined the game!
false
Wrong answer.

The correct answer was: True

Leaderboard:
ibrahim: 50 points
basmala: 33 points
zahraa: 0 points

A new round is starting soon!
[redacted]

```

The first round has ended. The second round is about to start. The third client's score started at zero, and the server began sending them questions just like the other clients.

```

pythonProject > Terminal Local + v
server.py x client.py
Terminal Local + v
80 Answer received from ibrahim ('127.0.0.1', 63236): 8
...
Answer received from basmala ('127.0.0.1', 63237): 4
Answer received from zahraa ('127.0.0.1', 62938): 2
The correct answer was: 8

Leaderboard:
ibrahim: 60 points
basmala: 46 points
zahraa: 34 points

Broadcasted question 4: Is water wet? (True/False)
Answer received from basmala ('127.0.0.1', 63237): false
Answer received from zahraa ('127.0.0.1', 62938): yes
Answer received from ibrahim ('127.0.0.1', 63236): true
[redacted]

Question 3: How many legs does a spider have?
2
Wrong answer.

The correct answer was: 8

Leaderboard:
ibrahim: 60 points
basmala: 46 points
zahraa: 34 points

[redacted]
Question 4: Is water wet? (True/False)
true
Correct!
[redacted]

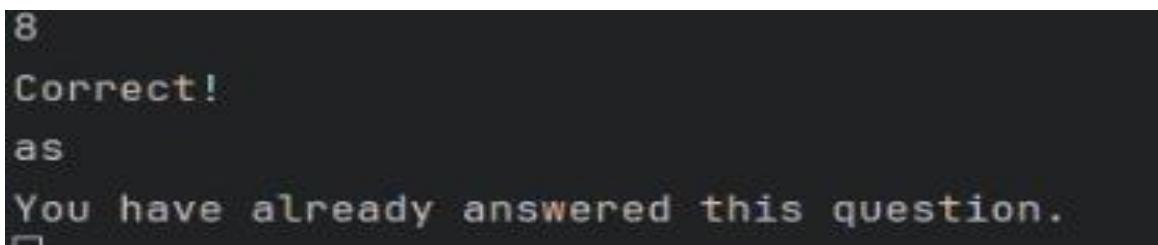
Question 3: How many legs does a spider have?
4
Wrong answer.

The correct answer was: 8

Leaderboard:
ibrahim: 60 points
basmala: 46 points
zahraa: 34 points

[redacted]
Question 4: Is water wet? (True/False)
false
Wrong answer.
[redacted]
```

We played the second round (which is why the results changed), and the ranking remained from highest to lowest.



This message will appear if you attempt to submit more than one answer.

The terminal window shows a game session between three participants: basmala, ibrahim, and zahraa. The session starts with a broadcasted question about the number of fingers on a human hand. The correct answer is 5. Ibrahim answers correctly, while basmala and zahraa answer incorrectly. The session then moves to a new round where a question about water being wet is asked. The correct answer is False. Ibrahim answers correctly, while basmala and zahraa answer incorrectly.

```
Answer received from basmala ('127.0.0.1', 63237): false
...
Answer received from zahraa ('127.0.0.1', 62938): yes
Answer received from ibrahim ('127.0.0.1', 63236): true
The correct answer was: True

Leaderboard:
ibrahim: 70 points
basmala: 46 points
zahraa: 34 points

Starting a new round...
Broadcasted question 1: How many fingers does a human hand have?
Answer received from ibrahim ('127.0.0.1', 63236): 5
Answer received from zahraa ('127.0.0.1', 62938): 4
[ ]
yes
Wrong answer.

The correct answer was: True

Leaderboard:
ibrahim: 70 points
basmala: 46 points
zahraa: 34 points

[ ] A new round is starting soon!
Question 1: How many fingers does a human hand have?
5
Correct!

Leaderboard:
ibrahim: 70 points
basmala: 46 points
zahraa: 34 points

A new round is starting soon!
Question 4: Is water wet? (True/False)
false
Wrong answer.

The correct answer was: True

Leaderboard:
ibrahim: 70 points
basmala: 46 points
zahraa: 34 points

[ ] A new round is starting soon!
Question 1: How many fingers does a human hand have?
[ ]
```

A new round has started. We will not let Basmalah answer any questions in this round (so she will be excluded).

```
ibrahim: 94 points
basmala: 46 points
zahraa: 34 points [ ]
```

Question 2: Is Earth a planet? (True/False)

The correct answer was: True

Leaderboard:

```
ibrahim: 94 points
basmala: 46 points
zahraa: 34 points
```

Question 3: How many legs does a spider have?

```
[ ]
```

```

Terminal Local + v
ibrahim: 94 points
basmala: 46 points
zahraa: 34 points

Broadcasted question 4: Can fish fly? (True/False)
The correct answer was: False

Leaderboard:
ibrahim: 94 points
basmala: 46 points
zahraa: 34 points

Removing client basmala (('127.0.0.1', 63237)): Did not respond during the entire round
Starting a new round...
[...]
basmala: 46 points
zahraa: 34 points

Question 4: Can fish fly? (True/False)

The correct answer was: False

Leaderboard:
ibrahim: 94 points
basmala: 46 points
zahraa: 34 points

basmala has left the game!
A new round is starting soon!
[...]
You have been removed from the game.

pythonProject > server.py
216:1 CRLF UTF-8 4 spaces Python 3.12 (pythonProject) ⌂

```

Since Basmalah didn't answer any questions in this round, she has been excluded.

```

Terminal Local + v
... Removing client basmala (('127.0.0.1', 63237)): Did not respond during the entire round
Starting a new round...
Broadcasted question 1: Is 2 + 2 equal to 5? (True/False)
Answer received from ibrahim ('127.0.0.1', 63236): false
Answer received from zahraa ('127.0.0.1', 62958): false
The correct answer was: False

Leaderboard:
ibrahim: 116 points
zahraa: 49 points

Broadcasted question 2: What shape has three sides?
Removing client ibrahim (('127.0.0.1', 63236)): Quit the game
[...]
basmala has left the game!
A new round is starting soon!
Question 1: Is 2 + 2 equal to 5? (True/False)
false
Correct!
The correct answer was: False

Leaderboard:
ibrahim: 116 points
zahraa: 49 points

Question 2: What shape has three sides?
quit
Exiting game...
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject>
ibrahim: 94 points
basmala: 46 points
zahraa: 34 points

Question 4: Can fish fly? (True/False)

The correct answer was: False

Leaderboard:
ibrahim: 94 points
basmala: 46 points
zahraa: 34 points

You have been removed from the game.

pythonProject > server.py
216:1 CRLF UTF-8 4 spaces Python 3.12 (pythonProject) ⌂

```

Ibraheem has quit, so he was removed from the game and the server.

The screenshot shows a terminal window with two tabs: 'Terminal' and 'Local'. The 'Terminal' tab contains the following text:

```
Leaderboard:  
ibrahim: 116 points  
zahrae: 49 points  
  
Broadcasted question 2: What shape has three sides?  
Removing client ibrahim (('127.0.0.1', 63236)): Quit the game  
The correct answer was: Triangle  
  
Leaderboard:  
zahrae: 49 points  
  
Not enough players to continue. Server shutting down.  
Error in listen_for_clients: [WinError 10038] An operation was attempted on something that is not a socket  
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject>
```

The 'Local' tab shows the game's logic:

```
A new round is starting soon!  
Question 1: Is 2 + 2 equal to 5? (True/False)  
false  
Correct!  
  
The correct answer was: False  
  
Leaderboard:  
ibrahim: 116 points  
zahrae: 49 points  
  
Question 2: What shape has three sides?  
quit  
Exiting game...  
(.venv) PS C:\Users\USER\PycharmProjects\pythonProject>  
ibrahim: 94 points  
basmala: 46 points  
zahrae: 34 points  
  
Question 4: Can fish fly? (True/False)  
  
The correct answer was: False  
  
Leaderboard:  
ibrahim: 94 points  
basmala: 46 points  
zahrae: 34 points  
  
You have been removed from the game.
```

The last client was removed because the number of clients became less than 2, and the server stopped.

References

- Kurose, James F., and Keith W. Ross. *Computer Networking: A Top-Down Approach*. 8th ed., Pearson, 2017.
- <https://www.geeksforgeeks.org/socket-programming-python/>

Team work

We all worked together on each and every task, ensuring that the workload was divided equally among all of us, so that everyone contributed in a fair and balanced way.

