Computer Networks Project

Names & IDs:

Aly Ahmed Abouelnasr 6511

Maged Magdy Zearban 6395

Fares Mohamed ElSayed Ramdan 6537

Test Cases:

```
GET [Server]_Data.txt 127.0.0.1
GET /hypertext/WWW/TheProject.html info.cern.ch 80
POST [Client]_Data.txt 127.0.0.1
GET [Server]_Data.txt 127.0.0.1
```

```
Case 1: GET [Server]_Data.txt 127.0.0.1
```

Server :

```
[('127.0.0.1', 63222)]
GET [Server]_Data.txt 127.0.0.1

[SERVER] Local Searching........
[SERVER] I am here in GET condition......
[SERVER] Done
```

Client :

```
[CLIENT] Command  : GET [Server]_Data.txt 127.0.0.1
[CLIENT] File : [Server]_Data.txt
FIRST TIME..............
Command Sent

Data Received .....

This is the Data in the Server File .
```

```
Case 2: GET /hypertext/WWW/TheProject.html info.cern.ch
80
```

Server :

```
[('127.0.0.1', 63222)]
GET /hypertext/WWW/TheProject.html info.cern.ch 80

[SERVER] Global Searching........
[SERVER] I am here in GET condition......
[SERVER] Done
```

Client :

```
FIRST TIME..............

Command Sent

Data Received .....

HTTP/1.1 200 OK
Date: Sun, 15 May 2022 19:10:26 GMT
Server: Apache
Last-Modified: Thu, 03 Dec 1992 08:37:20 GMT
ETag: "8a9-291e721905000"
Accept-Ranges: bytes
Content-Length: 2217
Connection: close
Content-Type: text/html
```

Case 3: POST [Client]_Data.txt 127.0.0.1

Server :

```
[('127.0.0.1', 63222)]
POST [Client]_Data.txt 127.0.0.1

This is the Data in the Client File .

[SERVER] Local Searching........
[SERVER] I am here in POST condition......
[SERVER] Done
```

Client :

```
[CLIENT] Command  : POST [Client]_Data.txt 127.0.0.1
[CLIENT] File : [Client]_Data.txt

FIRST TIME..............

Command Sent

[SERVER] 200\ok.........
```

```
Case 4: POST [Client]_Data.txt 127.0.0.1
```

Server :

No Server Logs as the command was fulfilled from **CLIENT CACHE**

Client :

```
[CLIENT] Command  : GET [Server]_Data.txt 127.0.0.1
[CLIENT] File : [Server]_Data.txt

FROM CACHE.............

This is the Data in the Server File .
```

Persistent:

```
if (time.time() - PresistantTime) > 5 * No_Commands:
    print("Time Has Ran Out")
    print(time.time() - PresistantTime)
    break
```

Time Out is Determined By the Number Of Commands A Client Sends , Ex. If client sends 3 Commands wait time is 15 Seconds

Pipelining:

```
def handle_client(conn: object, addr):
    print(f"[NEW CONNECTION] {addr} connected\n")
    connected = True
    PresistantTime = 0
    No_Commands = 0
    while connected:

        msg = conn.recv(1024).decode(FORMAT)

        if msg:
            No_Commands = No_Commands + 1
            PresistantTime = time.time()
            if msg == DISCONNECT:
                print("I found a disconnect")
                break
            print(f"[{addr}] \n{msg}\n")
            ClientThread = threading.Thread(target=check, args=(msg, conn))
            ClientThread.start()
```

Server Creates a thread to run the client command while listening to the new Commands coming in .