Lab 3: Web Client Lab

In this lab, you will develop a web client that connects to the web server in Lab 1 and download the `index.html` file and displays it. I If you run both on the same computer, use different terminals or environments for the client and server, e.g., running the server in Visual Studio Code, and the client on the command prompt. (You can also connect to some other web server online by modifying the `serverName` and `serverPort` in the code.) If the requested file is not present in the server, the client should get an HTTP "404 Not Found" message.

Code

Below you will find the skeleton code for the web client. You are to complete the skeleton code. The places where you need to fill in code are marked with **`#Fill in start`** and **`#Fill in end`**. Each place may require one or more lines of code.

Running the Client

Put an HTML file (e.g., index.html) in the same directory that the server is in. Run the server program. Determine the IP address of the host that is running the server (e.g., 128.238.251.26), or use `localhost` if both client and server are on the same machine. Your server is already working by testing it with a web browser. Now you need to make the client work.

From another host or another terminal on the same host, run your client. In the client code, provide the server name and port. 'index.html' is the name of the file you placed in the server directory. The client should then display the contents of index.html. If you omit ":6789", the browser will assume port 80 and you will get the web page from the server only if your server is listening at port 80. Then try to get a file that is not present at the server, and the client should get a "404 Not Found" message.

What to Hand in

You will hand in the complete client code and server code, along with the screen shots of your client, verifying that you actually receive the contents of the HTML file from the server.

Skeleton Python Code for the Web Client

```
from socket import *

# Server details
serverName = 'localhost'  # or the server IP address if the server is
on a different machine
serverPort = 6789  # Make sure this matches the server's port

# Create a TCP client socket
#Fill in start
#Fill in end

# Connect to the server
#Fill in start
#Fill in end


# Prepare the HTTP GET request. The .format(serverName) method call at
# the end of the string is used to insert the value of the serverName #
# variable into the placeholder {} in the Host header.
request = "GET /index.html HTTP/1.1\r\nHost:
{}\r\n\r\n".format(serverName)

try:
    # Send the request to the server
    #Fill in start
    #Fill in end

    # Receive and print the server's response
    response = clientSocket.recv(1024).decode()
    print("Server response:")
    print(response)

    # Receive and print the content (if any)
    while True:
        data = #Fill in start   #Fill in end
        if not data:
            break
        print(data)

except Exception as e:
    print("An error occurred:", str(e))

finally:
    # Close the client socket
    #Fill in start
    #Fill in end
```