Homework 9 – Programming Assignment

CS5700

Get HTML from Web Server

Points Possible: 20

Problem Statement:

Using the server.py and client.py code on Canvas as a guide, modify them to accommodate the below use case.

High-Level Requirements:

- The client will send an HTTP GET request to the server to retrieve the contents of an HTML file on that server.
- The server will respond back with the connection status and HTML contents.

Detailed Requirements:

- When executing the server.py, you will pass in a port number such that the server will bind to that port number. In other words, the server's port number should not be static or hard coded.
- The server's IP address must not be hard coded in any way in either the client.py or server.py code. The IP address must be retrieved dynamically and printed to the console. For example, you are not allowed to hard-code 127.0.0.1 or 0.0.0.0 anywhere in your code. Doing so will result in a lot of points being deducted.
- When executing the client.py, you will pass in the server's IP address, port number, and the file name of the html file to which you would like to retrieve the contents of. client.py's input parameters must be in that order (IP address, then Port Number, then HTML File Name).
- If a connection is successfully made to the server and the file exists on the server, the server will then respond back to the client indicating the connection was successful ("Connection Successful!"), the success HTTP status ("HTTP/1.1 200 OK"), and the contents of the HTML file on that server. See example section below.
- If a connection is successfully made to the server, but the file doesn't exist on the server, the server will then respond back indicating the connection was successful ("Connection Successful!"), but with a not found error ("HTTP/1.1 404 Not Found"). See example section below.
- If a connection is not successfully made to the server (e.g. if the wrong port number is sent), then "Error while connecting!" should be displayed. See example section below.
- Once server.py has been executed, the connection should remain open between requests. It should not close after each request.
- In addition, the server.py should output the following:
    - Server IP address
    - Sever port number
- server.py, client.py, and the html file should all exist in the same folder.
- Three sample HTML files have been uploaded to Canvas for your reference (helloworld.html, testhtmlfile01.html, testhtmlfile02.html). You can use those for your testing.

What to Submit:

- client.py
- server.py

Submission Requirements:

1. Write your name as a comment at the top of your code.
2. You should submit two files: client.py and server.py
    a. Using any other file name will get you points deducted.
3. Your program must be able to be executed via the command prompt in Windows or the Terminal on the Mac. See the examples below.
4. You can work with other students or individually, up to you. However, you must submit your assignment individually on Canvas.
5. You must submit your file(s) on Canvas. Any other submission method (such as email) will be rejected and you will receive zero credit.
6. This assignment must be done using the same Python version as mentioned in the syllabus.
7. The format of the output after you execute the client.py and server.py should match exactly to the format that you see in the example below. Anything different or any other result that's outputted via our test cases will result in point deductions.
8. Canvas will automatically add a hyphen and a number to the file name if you upload the same file more than once (e.g. client-2.py or server-3.py). That's fine and don't worry about it, I will not deduct points for that.
9. Do not have your code in any class (e.g. "`class Solution:`"). You will get points deducted if you do.
10. Any deviations from the requirements and provided examples below will result in point deductions, up to and including receiving zero credit.

Example:

The following is run from the command prompt. This is how your code will be graded. Your program absolutely needs to be able to be run from the command prompt or Terminal, otherwise you will get zero credit. No exceptions. The output of your code should display in the command prompt or Terminal console. Red font color indicates text from the command prompt.

First, open a command prompt window and execute server.py. Be sure to pass in a port number. When your code is being graded, we may use a different port number than what's shown in the examples below.

>python server.py 8521

server IP address: 192.168.254.30

server port number: 8521

Ready to serve...

Next, open a command prompt window to execute client.py. As mentioned in the Detailed Requirements above, you would need to pass in the server's IP address, port number, and HTML file name to client.py.

The following examples assumes your inputted server IP address, port number, and HTML file names are all correct.

```
>python client.py 192.168.254.30 8521 helloworld.html

Connection Successful!


----------HTTP RESPONSE----------

HTTP/1.1 200 OK


<html>
<header><title>CS5700 Is Awesome!</title></header>
<body>
Hello world
</body>
</html>
----------END OF HTTP RESPONSE----------


>python client.py 192.168.254.30 8521 testhtmlfile01.html

Connection Successful!


----------HTTP RESPONSE----------

HTTP/1.1 200 OK


<!DOCTYPE html>
<html>
<body>

<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>

</body>
</html>

----------END OF HTTP RESPONSE----------
```

```
>python client.py 192.168.254.30 8521 testhtmlfile02.html

Connection Successful!


----------HTTP RESPONSE----------

HTTP/1.1 200 OK


<!DOCTYPE html>

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>Test HTML File 02</title>
  <LINK href="styles.css" rel="stylesheet" type="text/css">
</head>

<body>

<!-- Feel free to change this text here -->
<p>
  Testing 1 2 3 4.
</p>
<p>
  Additional Text.
</p>

</body>
</html>
----------END OF HTTP RESPONSE----------
```

The following example assumes your inputted server IP address and port number are correct, but the HTML file name is incorrect (i.e. doesn't exist on the server).

```
>python client.py 192.168.254.30 8521 doesntexist.html

Connection Successful!


----------HTTP RESPONSE----------

HTTP/1.1 404 Not Found


----------END OF HTTP RESPONSE----------
```

The following examples assumes your inputted server IP address is correct, but the port number is incorrect. In this scenario, it doesn't matter if you provide the correct HTML file name, the output from the client should still be "Error while connecting!".

>python client.py 192.168.254.30 8520 doesntexist.html

Error while connecting!

>python client.py 192.168.254.30 8520 testhtmlfile01.html

Error while connecting!

Wrong IP address should also output "Error while connecting":

>python client.py 192.168.254.31 8520 testhtmlfile01.html

Error while connecting!

Constraints:

- Your code should accommodate both success scenarios and failure scenarios, as mentioned above.
- Library restrictions: you can only import the following libraries:
    - client.py:
        - from socket import *
        - import sys
    - server.py:
        - from socket import *
        - import sys
    - As you can see, both client.py and server.py have the same library restrictions.

Grading Guidelines:

- Does the program meet the requested requirements/criteria?
- Are the submission instructions followed?
- Does your code compile and execute?
- Does your code pass my test cases?