



Question 1: TCP Server/Client Quiz App with Multi-threading?

As an improvement to previous first homework, build a TCP server and client quiz application using Python. The server should handle multiple client connections simultaneously using multi-threading. The application should allow clients to connect, participate in a quiz, and receive their quiz scores upon completion.

Requirements:

- A. The server should be able to handle multiple client connections concurrently.
- B. The quiz should consist of a set of pre-defined questions stored on the server.
- C. Each client should connect to the server and receive the quiz questions.
- D. Clients should send their answers to the server.
- E. The server should keep track of the scores for each client.
- F. At the end of the quiz, the server should send the final scores to each client.

Solution:

An overview on the program

- ___ first we turn on the server and let it wait for a client
- ___ when a client connects with server the program will create a new thread to the client so that the server can serve multiple clients **simultaneously**
- ___ the client enters his name and then the server sends the questions to the client in sequence to answer them
- ___ finally the server shows the name of the client and his result on the screen
- ___ **handle_client** is used to handle clients
- ___ before the data is sent, it is encoded and then it is sent using the **send()** method
- ___ the data is also received using the **recv()** method
- ___ the server waits for client using the **accept()** method
- ___ server works on the **loop back address (127.0.0.1)** and the **port (5555)**

Server's Code :

server1.py	client1.py	client2.py
<pre>1 import socket 2 import threading 3 import random 4 5 server_ip = '127.0.0.1' 6 server_port = 5555 7 8 QUIZ_QUESTIONS = [9 {'question': 'What is the capital of France?', 'answer': 'Paris'}, 10 {'question': 'What is the largest planet in our solar system?', 'answer': 'Jupiter'}, 11 {'question': 'What is the boiling point of water?', 'answer': '100 degrees Celsius'} 12] 13 14 client_scores = {} 15 16 # handle clients connection 17 def handle_client(client_socket): 18 19 client_name = client_socket.recv(1024).decode() 20 21 22 client_socket.send(f'Welcome to the quiz, {client_name}!\n'.encode()) 23 24 # send the quiz questions to the client 25 k = 0 26 for i, question in enumerate(QUIZ_QUESTIONS): 27 client_socket.send(f'Question {i+1}: {question["question"]}\n'.encode()) 28 29 client_answer = client_socket.recv(1024).decode().strip() 30 31 # check if the client's answer is correct 32 if client_answer.lower() == question['answer'].lower(): 33 client_socket.send('Correct!\n'.encode()) 34</pre>		

```

35         client_scores[client_name] = client_scores.get(client_name, 0) + 1
36         k = k + 1
37     else:
38         client_socket.send(f'Incorrect! The correct answer is {question["answer"]}\n'.encode())
39     if k == 0 :
40         client_scores[client_name] = client_scores.get(client_name, 0) + 0
41     # send the client's score to the client
42     client_socket.send(f'Thank you for playing, {client_name}! Your score is {client_scores[client_name]}\n'.encode())
43
44     client_socket.close()
45
46 # Create the server socket
47 server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
48
49 server_socket.bind((server_ip, server_port))
50 server_socket.listen()
51
52 print(f'Server listening on {server_ip}:{server_port}')
53
54 while True:
55     client_socket, client_address = server_socket.accept()
56
57     # create a new thread to handle the client connection
58     thread = threading.Thread(target=handle_client, args=(client_socket,))
59     thread.start()
60

```

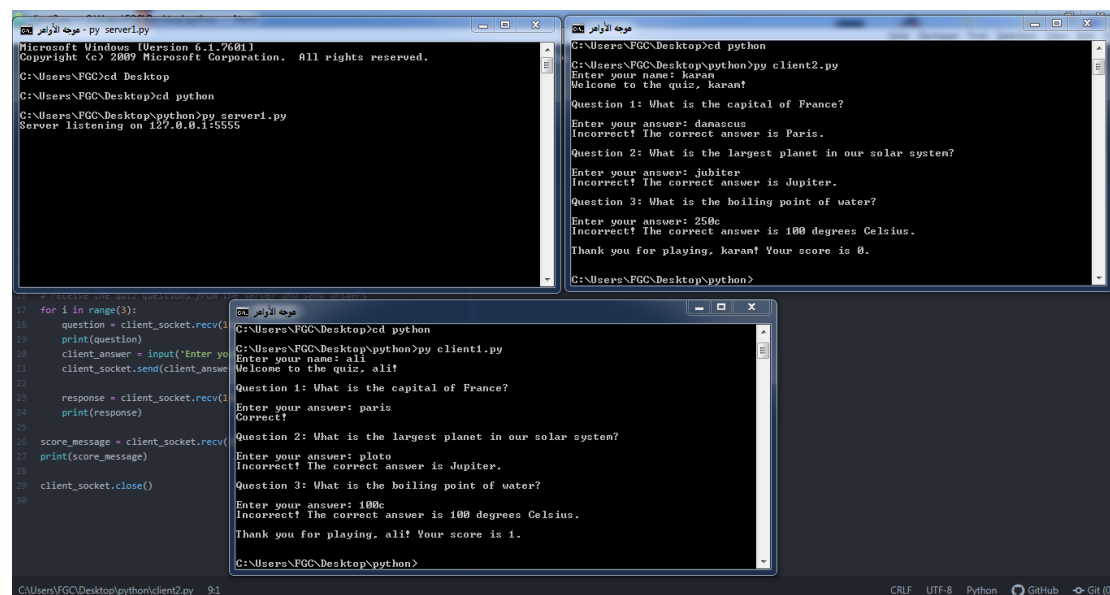
Client1 code :

server1.py	client1.py	client2.py
<pre> 1 import socket 2 3 server_ip = '127.0.0.1' 4 server_port = 5555 5 6 client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM) 7 8 client_socket.connect((server_ip, server_port)) 9 10 client_name = input('Enter your name: ') 11 client_socket.send(client_name.encode()) 12 13 welcome_message = client_socket.recv(1024).decode() 14 print(welcome_message) 15 16 # receive the quiz questions from the server and send answers 17 for i in range(3): 18 question = client_socket.recv(1024).decode() 19 print(question) 20 client_answer = input('Enter your answer: ') 21 client_socket.send(client_answer.encode()) 22 23 response = client_socket.recv(1024).decode() 24 print(response) 25 26 score_message = client_socket.recv(1024).decode() 27 print(score_message) 28 29 client_socket.close() 30 </pre>		

Client2 code :

server1.py	client1.py	client2.py
<pre>1 import socket 2 3 server_ip = '127.0.0.1' 4 server_port = 5555 5 6 client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM) 7 8 client_socket.connect((server_ip, server_port)) 9 10 client_name = input('Enter your name: ') 11 client_socket.send(client_name.encode()) 12 13 welcome_message = client_socket.recv(1024).decode() 14 print(welcome_message) 15 16 # receive the quiz questions from the server and send answers 17 for i in range(3): 18 question = client_socket.recv(1024).decode() 19 print(question) 20 client_answer = input('Enter your answer: ') 21 client_socket.send(client_answer.encode()) 22 23 response = client_socket.recv(1024).decode() 24 print(response) 25 26 score_message = client_socket.recv(1024).decode() 27 print(score_message) 28 29 client_socket.close() 30</pre>		

Program execution :



Question 2: Simple Website with Python Flask Framework

Create a simple website with multiple pages using Flask, HTML, CSS, and Bootstrap. The website should demonstrate your understanding of web design principles

. Requirements:

- A. Set up a local web server using XAMPP, IIS, or Python's built-in server (using Flask)
- . B. Apply CSS and Bootstrap to style the website and make it visually appealing.
- C. Ensure that the website is responsive and displays correctly on different screen sizes.
- D. Implement basic server-side functionality using Flask to handle website features.

Solution :

solution's steps :

- 1_ to work with flask we create a virtual environment that contains the project
- 2_ to create a virtual environment we enter the folder that we want to work in it

Then we write this command in the command line `python -m venv (name)`

3_ to activate the virtual environment we copy the path of the activate.bat file in the scripts folder then **we enter the path in the command** line and we attach to it `\activate.bat`

___ the project that has been done , is a simple website that shows the offers on shoes and it consists of pages

The backend code of the website :

```
flask1.py
1  from flask import Flask, redirect, url_for, render_template
2  import os
3  app = Flask(__name__)
4
5
6  picture = os.path.join('static', 'photo')
7  app.config['UPLOAD_FOLDER'] = picture
8
9  @app.route("/")
10 def index():
11     return render_template("index1.html")
12
13 @app.route("/product")
14 def product():
15     offer = os.path.join(app.config['UPLOAD_FOLDER'], 'offer.jpg')
16     return render_template("index.html", user_image = offer)
17
18 @app.route("/complain")
19 def complain():
20     return render_template("index2.html")
21
22
23 if __name__ == "__main__":
24     app.run(port=8888)
25
```

The frontend code of the website : the home page

```
index.html    index1.html    index2.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" />
6      <link rel="stylesheet" href="static/bootstrap.css" />
7      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
8      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
9      <title>shoe store</title>
10   </head>
11   <body>
12     <nav class="navbar navbar-expand-lg bg-body-tertiary">
13       <div class="container-fluid">
14         <button
15           class="navbar-toggler"
16           type="button"
17           data-bs-toggle="collapse"
18           data-bs-target="#navbarSupportedContent"
19           aria-controls="navbarSupportedContent"
20           aria-expanded="false"
21           aria-label="Toggle navigation"
22         >
23           <span class="navbar-toggler-icon"></span>
24         </button>
25         <div class="collapse navbar-collapse" id="navbarSupportedContent">
26           <ul class="navbar-nav me-auto mb-2 mb-lg-0">
27             <li class="nav-item">
28               <a class="nav-link active" aria-current="page" href="#">Home</a>
29             </li>
30             <li class="nav-item">
31               <a class="nav-link" href="/product" target="_blank">Offers</a>
32             </li>
33           </ul>
34           <form class="d-flex" role="search">
35             <input
36               class="form-control me-2"
37               type="search"
38               placeholder="Search"
39               aria-label="Search"
40             />
41             <button class="btn btn-outline-success" type="submit">
42               Search
43             </button>
44           </form>
45         </div>
46       </div>
47     </nav>
48     <h3>About us</h3>
49     <ul>
50       <li>
51         our main concern is to provide you with the best shoes offers that you
52         can find
53       </li>
54       <li>please click on offers to see the avialable sales</li>
55       <li>
56         if you run into any problem please contact with our support team using
57         the email below
58       </li>
59       <li><a href="/complain">shop.store@gmail.com</a></li>
60     </ul>
61   </body>
62 </html>
63
```

The products page code

index.html	index1.html
<pre>1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <link rel="stylesheet" href="static/style.css" /> 5 <meta charset="UTF-8" /> 6 <meta http-equiv="X-UA-Compatible" content="IE=edge" /> 7 <meta name="viewport" content="width=device-width, initial-scale=1.0" /> 8 <title>Nike Product</title> 9 </head> 10 <body> 11 <div class="main-div"> 12 <h2>Nike Air Max shoes</h2> 13 <div class="div"> 14 22 <div class="div1"> 23 <div class="div1-sub"> 24 <p class="prise">\$65.00</p> 25 <p class="shipping">free shipping</p> 26 </div> 27 <p class="discription">Step into style: Exclusive shoe Deals</p> 28 more information 29 <div> 30 <button class="button button1"></button> 31 <button class="button button2"></button> 32 <button class="button button3"></button> 33 <button class="button button4"></button> 34 <button class="button button5"></button> 35 <button class="button button6"></button></pre>	<pre>35 <button class="button button6"></button> 36 </div> 37 </div> 38 <div class="div2"> 39 <p class="productt">productt details</p> 40 <ul class="u-list"> 41 made with high-quality materials 42 available with customizable options 43 feature a visible Air cushioning 44 45 </div> 46 </div> 47 <button class="last-button">add to cart</button> 48 <p class="sale">sale</p> 49 </div> 50 </body> 51 </html> 52</pre>

The complain page :

```
index.html  index1.html  index2.html
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6      <link rel="stylesheet" href="static/bootstrap.css" />
7      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8      <title>complains</title>
9    </head>
10   <body>
11     <div class="mb-3">
12       <label for="exampleFormControlInput1" class="form-label">
13         Email address</label>
14       <input
15         type="email"
16         class="form-control"
17         id="exampleFormControlInput1"
18         placeholder="name@example.com"
19       />
20     </div>
21     <div class="mb-3">
22       <label for="exampleFormControlTextarea1" class="form-label">
23         Please Enter your complain</label>
24       <textarea
25         class="form-control"
26         id="exampleFormControlTextarea1"
27         rows="3"
28       />
29     </div>
30   </body>
31 </html>
```

the website's execution :

__ in order to show the website we have first to run the server after we activate the virtual environment

__ then we open a web browser and we type in the search field the ip address of the website

```
C:\Windows\System32\cmd.exe - py flask1.py
h1
Collecting MarkupSafe>=2.1.1 (from Werkzeug)>=2.2.2->flask)
Using cached https://files.pythonhosted.org/packages/9b/c1/9f44da5ca74f95116c6
44892152ca6514ecdc34c8297a3f40d886147863d/MarkupSafe-2.1.3-cp37-cp37m-win_amd64.
whl
Collecting colorama; platform_system == "Windows" (from click>=8.0->flask)
Using cached https://files.pythonhosted.org/packages/d1/d6/3965ed04c63042e047c
b6a3e6ed1a63a35087b6a609aa3a15ed8ac56c221/colorama-0.4.6-py2.py3-none-any.whl
Installing collected packages: zipp, typing-extensions, importlib-metadata, Mark
upSafe, Werkzeug, Jinja2, itsdangerous, colorama, click, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-2.2.3 click-8.1.3
colorama-0.4.6 flask-2.2.5 importlib-metadata-6.7.0 itsdangerous-2.1.2 typing-ex
tensions-4.6.3 zipp-3.15.0
You are using pip version 19.0.3, however version 23.1.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.

(store1) D:\store>py flask1.py
* Serving Flask app 'flask1'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on http://127.0.0.1:8888
Press CTRL+C to quit
```


shoe store

127.0.0.1:8888

New TabYouTube - قناة النورNetcut Know who(n...New Tab

HomeOffers

Search

Search

About us

- our main concern is to provide you with the best shoes offers that you can find
- please click on offers to see the available sales
- if you run into any problem please contact with our support team using the email below
- shop.store@gmail.com


shoe store

Nike Productt

127.0.0.1:8888/product

New TabYouTube - قناة النورNetcut Know who(n...New Tab

SALE



NIKE AIR MAX SHOES

\$65.00 *FREE SHIPPING*

Step into style: Exclusive shoe Deals

MORE INFORMATION

PRODUCTT DETAILS

- MADE WITH HIGH-QUALITY MATERIALS
- AVAILABLE WITH CUSTOMIZABLE OPTIONS
- FEATURE A VISIBLE AIR CUSHIONING

ADD TO CART

complaints

Nike Productt

127.0.0.1:8888/complain

New TabYouTube - قناة النورNetcut Know who(n...New Tab

Email address

name@example.com

Please Enter your complain