

# Ali Dali Slo ٢٥٧٦

- Question ١: 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 . Guidelines : • Use Python's socket module “don't use ٣rd-party packages”. • Implement multi-threading to handle multiple client connections concurrently . • Store the quiz questions and correct answers on the server side . Notes : • Write brief report describing the design choices you made and any challenges faced during implementation . • You can make a TCP Server/Client of your choice, such as Bank ATM, Chat application, or any other appropriate application that fulfil all requirements.

server.py

X client.py

X

```
1 import socket,threading
2
3 questions = {
4     "Who are real madrid legend? :\na.cr7\nb.messi": "a", "What is the capital of Syria? :\na.Damascus\nb.aleppo": "a", "What is the best department to study? :\na.ECce\nb.medecine": "b", "how many years you have yo study in ece? :\na.4\nb.5": "b", "What is liverpool ? :\na.club\nb.company": "a", "What is the top goal scorer in world? :\na.messi\nb.ronaldo": "a", "What is the currency of Canada? :\na.Dollar\nb.Euro": "a", "What is the currency of France? :\na.Dollar\nb.Euro": "b", "What is the capital of america? :\na.NY\nb.whashinton": "b", "What is the currency of Russia? :\na.Ruble\nb.Euro": "a", "What is the capital of France? :\na.paris\nb.marssiaile": "a", "What is the currency of Saudi Arabia? :\na.Riyal\nb.Dollar": "a", "What is the capital of Lebanon? :\na.tarablus\nb.beirut": "b", "What is the capital of Italy? :\na.roma\nb.milan": "a", "What is the currency of Egypt? :\na.Pound\nb.Dollar": "a", "What is argentina legend? :\na.higuain\nb.messi": "b", "What is the capital of United Kingdom? :\na.london\nb.manchester": "a", "What is the capital of Spain? :\na.madrid\nb.barcelona": "a", "What is the capital of jordan? :\na.amman\nb.irbed": "a", "What is the capital of ksa? :\na.ryiadh\nb.jeddah": "a"}
5
6 result = {}
7
8 def handle_request(cs, cadd):
9     cs.send(str(len(questions)).encode())
10    for question in questions:
11        cs.send(question.encode())
12        client_ans = cs.recv(10).decode().strip()
13        if client_ans.upper() == questions[question].upper():
14            result[cadd] = result.get(cadd, 0) + 1
15    score = result.get(cadd, 0)
16    cs.send(f"Score: {score}/{len(questions)}\n".encode())
17    cs.close()
18
19 class handle_client_thread(threading.Thread):
20     def __init__(self,cs,cadd):
21         threading.Thread.__init__(self)
22         self.cs=cs
23         self.cadd=cadd
24
25     def run(self):
26         handle_request(self.cs,self.cadd)
27
28 ss = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
29 ss.bind(('127.0.0.1',12345))
30 ss.listen(5)
31 print("i am waiting your answer|.")
32 ... =
```

```
server.py client.py
29 ss.bind(('127.0.0.1',12345))
30 ss.listen(5)
31 print("i am waiting your answer|.")
32 while True:
33     cs, cadd = ss.accept()
34     print(f"Connected to {cadd}")
35     client = handle_client_thread(cs, cadd)
36     client.start()
37
38
```

```
Administrator: C:\Windows\System32\cmd.exe - server.py
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\DELL\Desktop\ali>server.py
i am waiting your answer.
```

Administrator: C:\Windows\System32\cmd.exe

```
p.Riyal
p.Dollar: a
What is the capital of lebanon? :
p.tarabulus
p.beirut: a
What is the capital of Italy? :
p.roma
p.milan: a
What is the currency of Egypt? :
p.Pound
p.Dollar: a
What is argentina legend? :
p.higuain
p.messi: a
What is the capital of United Kingdom? :
p.london
p.manchester: a
What is the capital of Spain? :
p.madrid
p.barcelona: a
What is the capital of jordan? :
p.amman
p.irbed: a
What is the capital of ksa? :
p.ryiadh
p.jeddah: a
results: Score: 14/20

C:\Users\DELL\Desktop\ali>
```

```
Administrator: C:\Windows\System32\cmd.exe - serv...
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. 0000 000000 000000.

C:\Users\DELL\Desktop\ali>server.py
i am waiting your answer.
Connected to ('127.0.0.1', 57188)
Connected to ('127.0.0.1', 58072)
```

```
Administrator: C:\Windows\System32\cmd.exe - ...
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. 所有权利保留。

C:\Users\DELL\Desktop\ali>client.py
Who are real madrid legend? :
a.cr7
b.messi:
```

```
Administrator: C:\Windows\System32\cmd.exe - client2.py
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. 所有权利保留。

C:\Users\DELL\Desktop\ali>client2.py
Who are real madrid legend? :
a.cr7
b.messi:
```

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

```
from flask import Flask

ex1 = Flask(__name__)

@ex1.route("/")
def marhaba():
    return("hello bro")

@ex1.route("/name")
def name():
    return("Hello friend")

if __name__=="__main__":
    ex1.run()
```

```
1 from flask import Flask,render_template
2
3 app = Flask(__name__)
4
5
6 @app.route("/")
7 def hello():
8     return("Hello from our website")
9
10 @app.route("/info")
11 def info():
12     return(render_template("info.html"))
13
14 @app.route("/<name>")
15 def name(name):
16     return(f"Hello {name}")
17
18 if __name__=="__main__":
19     app.run(port=3333)
```

```
1|<html>
2|<title> marhaba </title>
3|<body>
4|<h1> hello bro </h1>
5|</body>
6|</html>
```

```
1|body {
2|  background-color: powderblue;
3|}
4|h1 {
5|  color: blue;
6|}
7|p {
8|  color: red;
9|}
```

```
1|<html>
2|<head>
3|  <title>Message</title>
4|  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
5|  <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.main.css') }}">
6|</head>
7|
8|<body>
9|  <h1>hi, welcome to the website</h1>
10|  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
11|  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/js/bootstrap.min.js"></script>
12|</body>
13|</html>
```

```

1<html>
2  <head>
3    <link rel="stylesheet" href="{{url_for('static', filename='css/bootstrap.main.css')}}">
4  </head>
5  <body>
6    <h3>Register the customer, fill the following form.</h3>
7    <form action = "http://localhost:5000/success" method = "POST">
8      <p>Name <input type = "text" name = "name" /></p>
9      <p>Email <input type = "email" name = "email" /></p>
10     <p>Contact <input type = "text" name = "contact" /></p>
11     <p>Pin code <input type = "text" name = "pin" /></p>
12     <p><input type = "submit" value = "submit" /></p>
13   </form>
14 </body>
15 </html>

```

```

1<!doctype html>
2<html>
3  <body>
4    <p><strong>Thanks for the registration. Confirm your details</strong></p>
5    <table border = 1>
6      {% for key, value in result.items() %}
7        <tr>
8          <th> {{ key }} </th>
9          <td> {{ value }} </td>
10        </tr>
11      {% endfor %}
12    </table>
13  </body>
14</html>

```



```
1 from flask import *
2 app = Flask(__name__)
3
4 @app.route('/')
5 def customer():
6     return render_template('customer.html')
7
8 @app.route('/success', methods = ['POST', 'GET'])
9 def print_data():
10     if request.method == 'POST':
11         result = request.form
12         return render_template("result_data.html", result = result)
13
14 if __name__ == '__main__':
15     app.run(port=4444)
16
```