Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No2

الجمهورية العربية السورية

اللاذقية -جامعــة تشريـــن

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

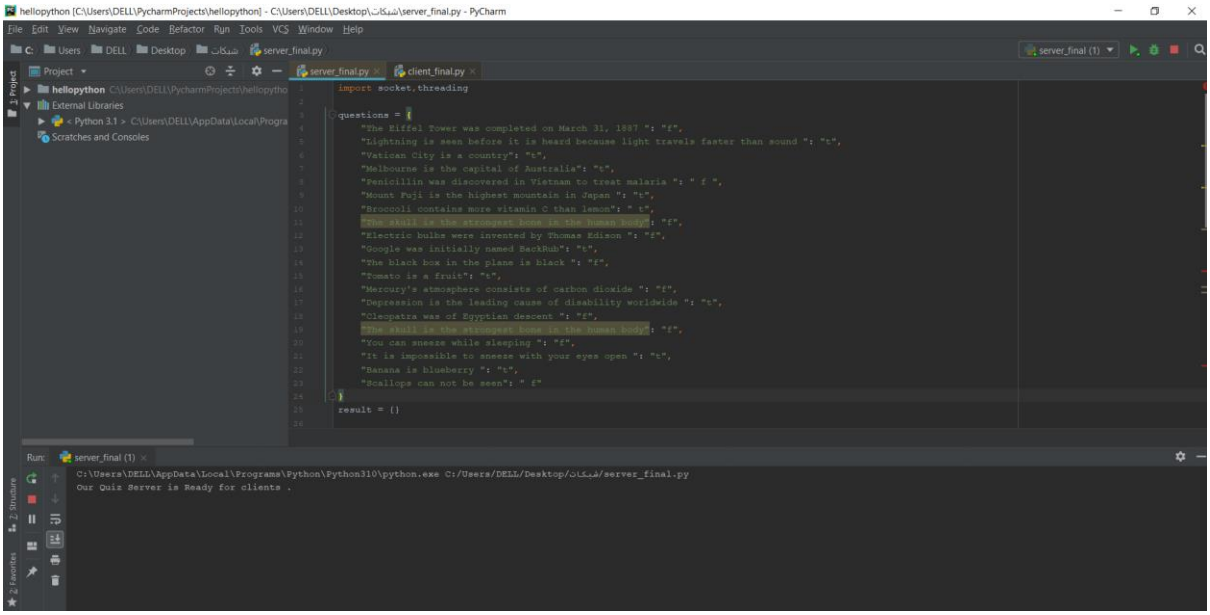
السنة الخامسة: وظيفة 2 برمجة شبكات

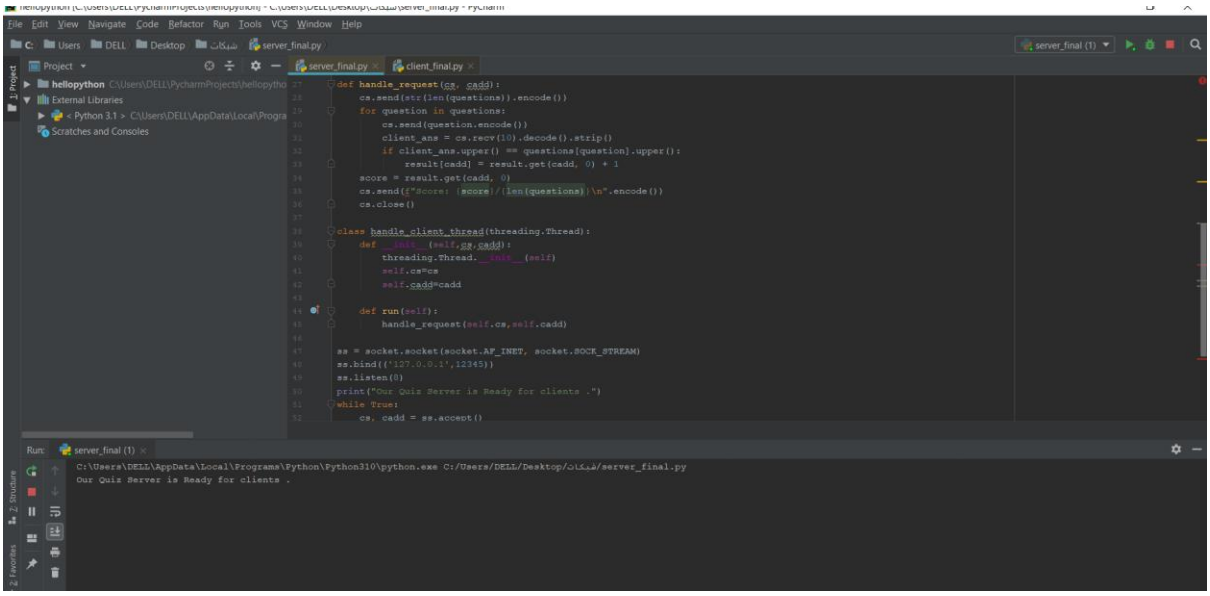
Name:_Rama assaf, Number:2732, To GitHub:__no_____Submitted

Second Network programming Homework

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

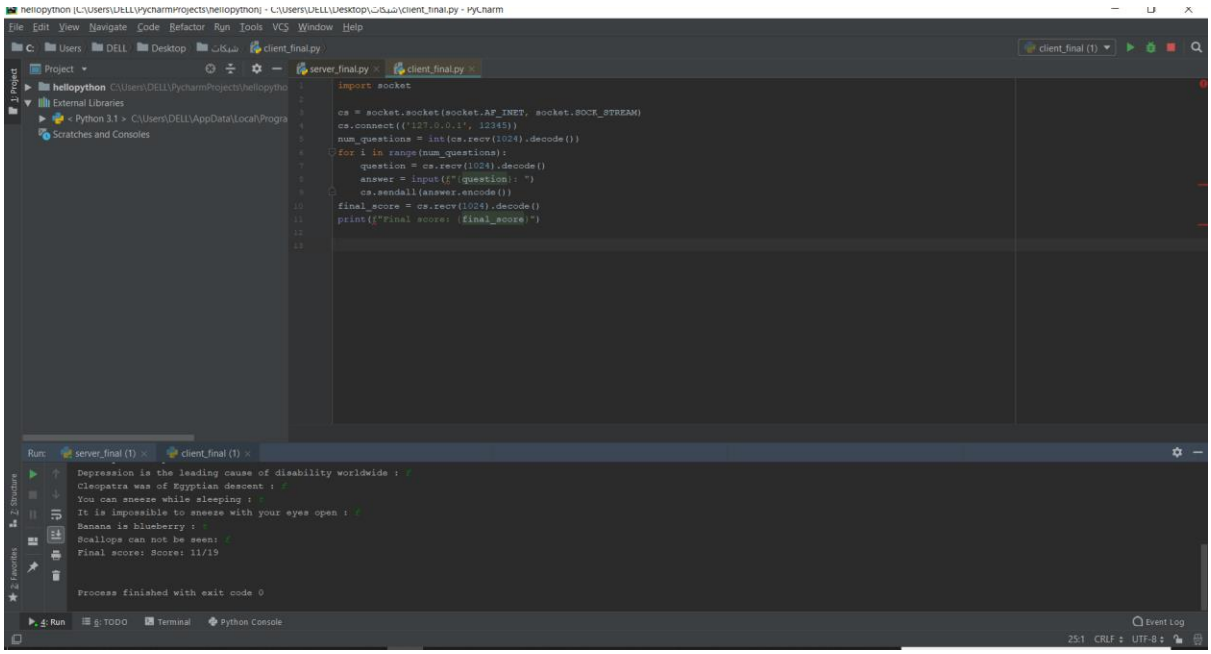
As an improvement to previous first homework, build a TCP server and client quiz application using Python. The server should handle multiple client connections simultaneously using multi-threading. The application should allow clients to connect, participate in a quiz, and receive their quiz scores upon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





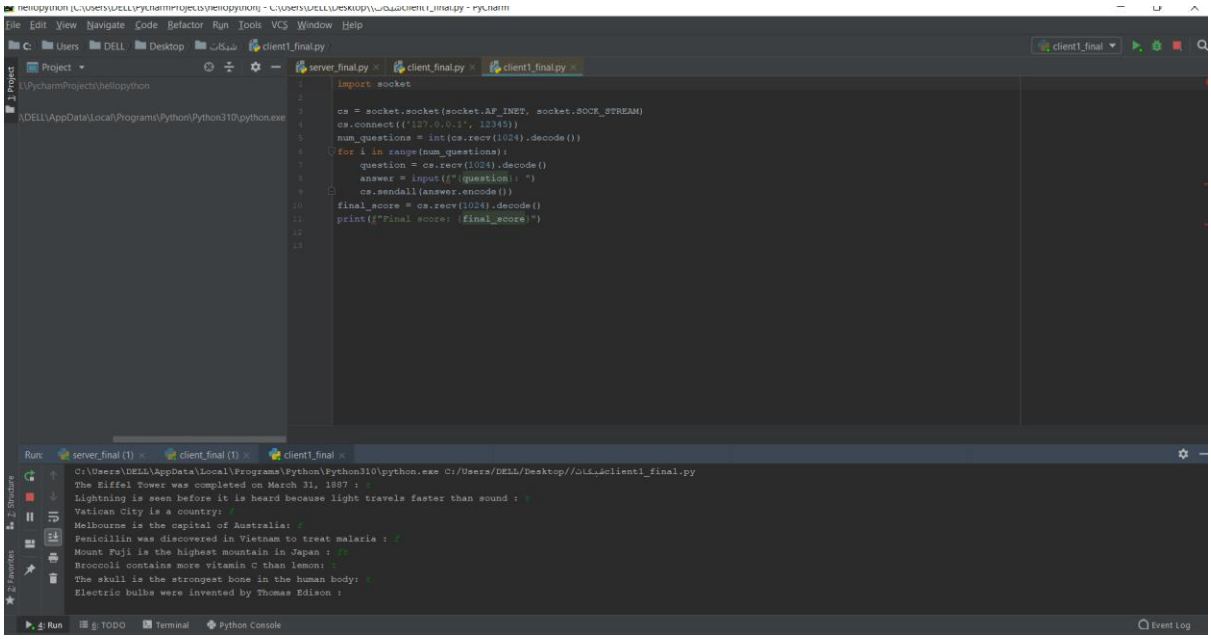
```
def handle_request(cs, cadd):
    cs.send(str(len(questions)).encode())
    for question in questions:
        cs.send(question.encode())
        client_ans = cs.recv(10).decode().strip()
        if client_ans.upper() == questions[question].upper():
            result[cadd] = result.get(cadd, 0) + 1
    score = result.get(cadd, 0)
    cs.send(f"Score: {score}/{len(questions)}\n".encode())
    cs.close()

class handle_client_thread(threading.Thread):
    def __init__(self,cs,cadd):
        threading.Thread.__init__(self)
        self.cs=cs
        self.cadd=cadd

    def run(self):
        handle_request(self.cs,self.cadd)

ss = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
ss.bind(('127.0.0.1',12345))
ss.listen(8)
print("Our Quiz Server is Ready for clients .")
while True:
    cs, cadd = ss.accept()
```

```
C:\Users\DELL\AppData\Local\Programs\Python\Python310\python.exe C:/Users/DELL/Desktop/شبكات/server_final.py
Our Quiz Server is Ready for clients .
```



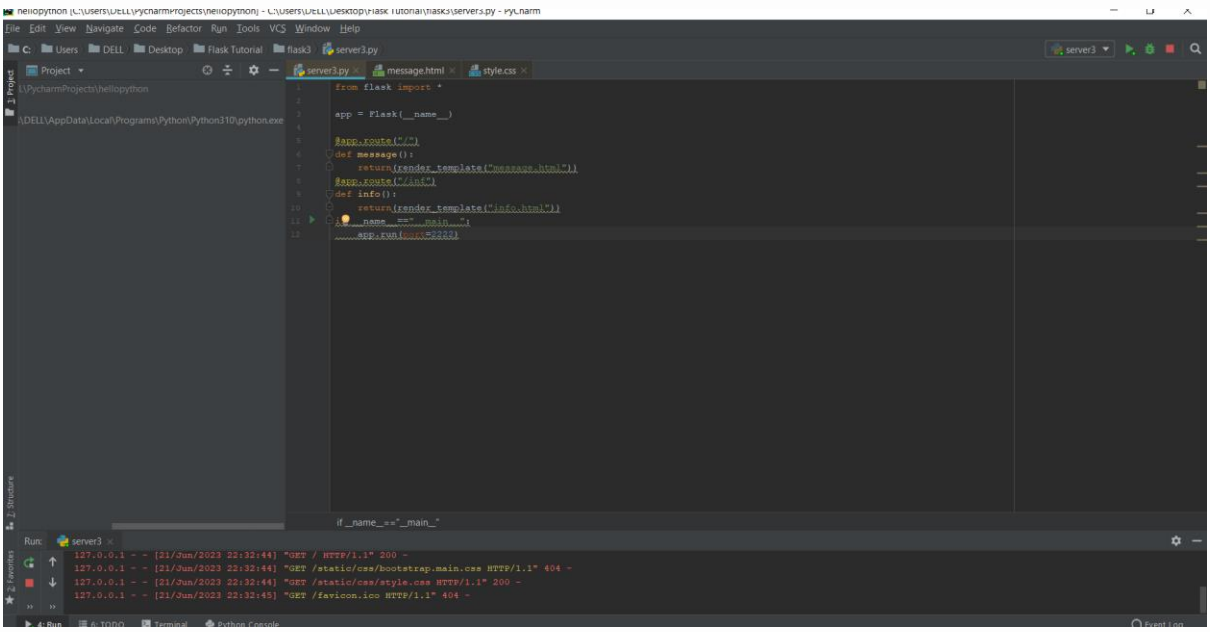
```
import socket

cs = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
cs.connect(('127.0.0.1', 12345))
num_questions = int(cs.recv(1024).decode())
for i in range(num_questions):
    question = cs.recv(1024).decode()
    answer = input(f"{question}: ")
    cs.sendall(answer.encode())
final_score = cs.recv(1024).decode()
print(f"Final score: {final_score}")
```

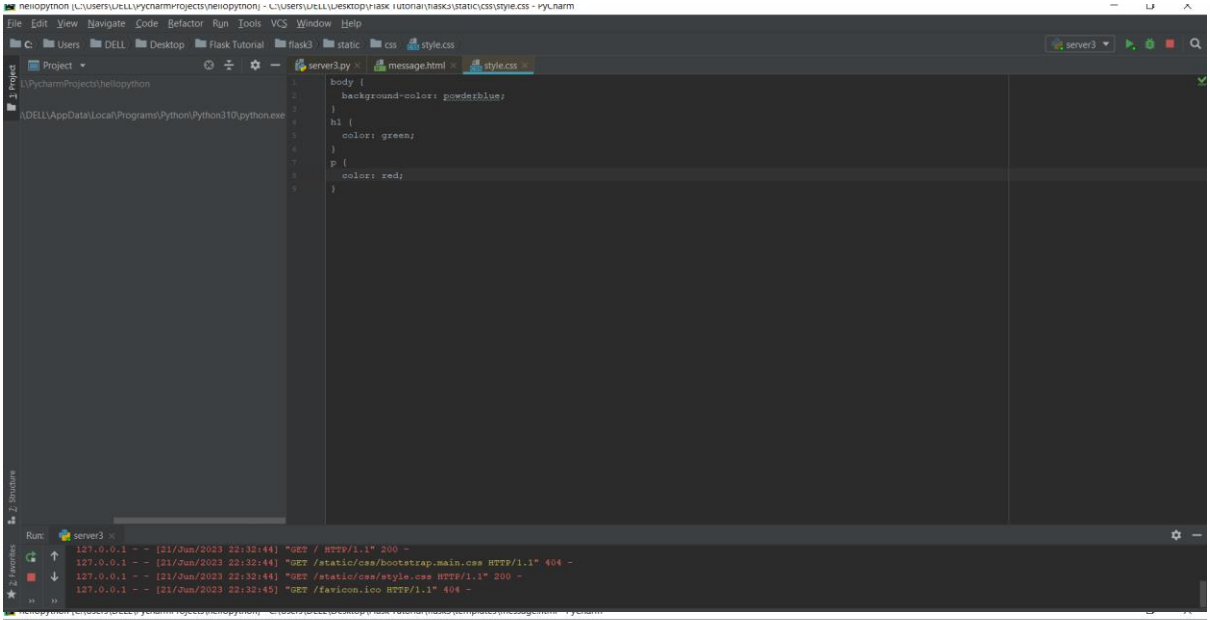
```
Depression is the leading cause of disability worldwide : f
Cleopatra was of Egyptian descent : f
You can sneeze while sleeping : f
It is impossible to sneeze with your eyes open : f
Banana is blueberry : t
Scallops can not be seen: f
Final score: Score: 11/15


Process finished with exit code 0
```

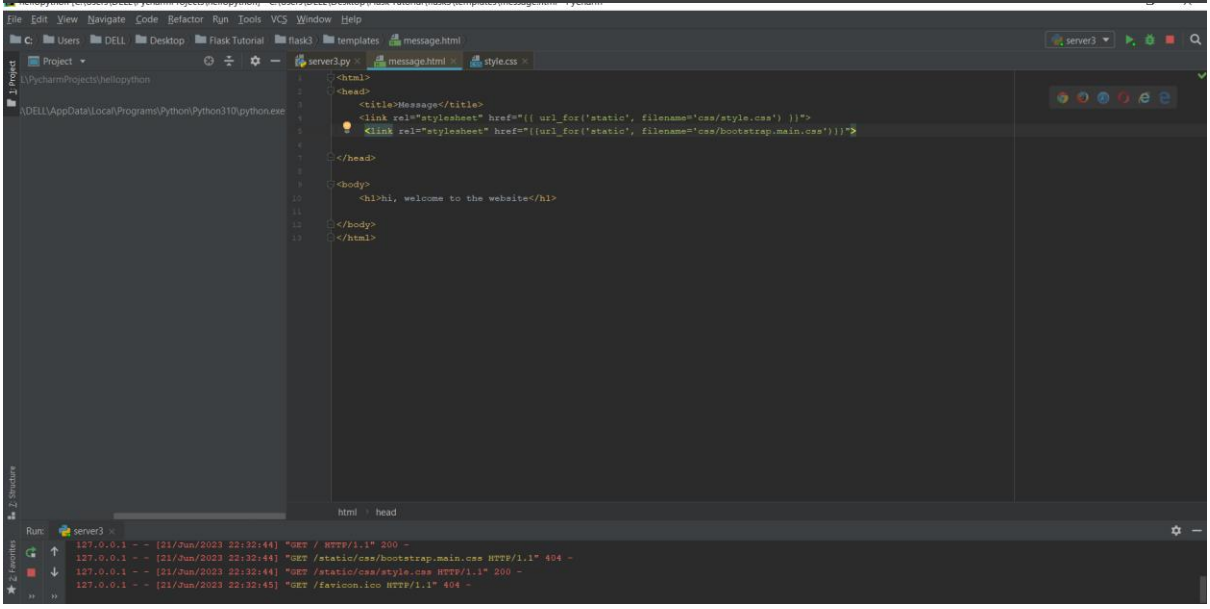


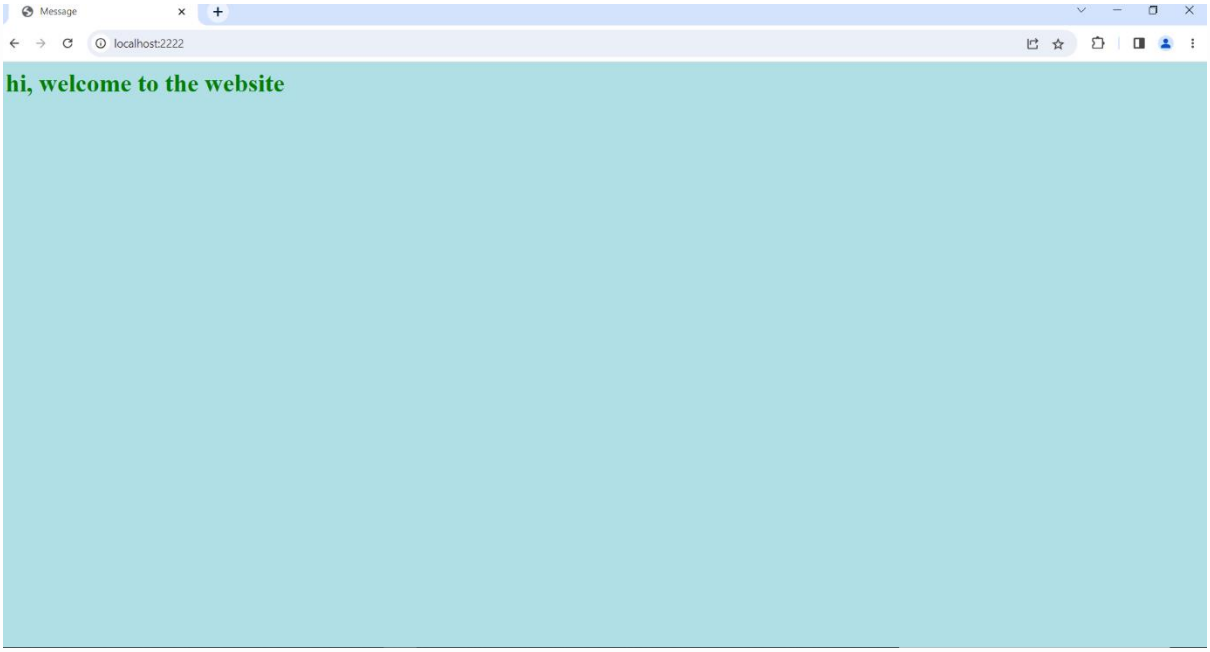
Question 2: Simple Website with Python Flask Framework

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




File Edit View Navigate Code Refactor Run Tools VCS Window Help
server3.py
message.html
style.css
body {
background-color: powderblue;
}
h1 {
color: green;
}
p {
color: red;
}
Run: server3
127.0.0.1 - - [21/Jun/2023 22:32:44] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/Jun/2023 22:32:44] "GET /static/css/bootstrap.main.css HTTP/1.1" 404 -
127.0.0.1 - - [21/Jun/2023 22:32:44] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [21/Jun/2023 22:32:45] "GET /favicon.ico HTTP/1.1" 404 -


File Edit View Navigate Code Refactor Run Tools VCS Window Help
server3.py
message.html
style.css
<html>
<head>
<title>Message</title>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
<link rel="stylesheet" href="{{url_for('static', filename='css/bootstrap.main.css')}}">
</head>
<body>
<h1>hi, welcome to the website</h1>
</body>
</html>
html head
Run: server3
127.0.0.1 - - [21/Jun/2023 22:32:44] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/Jun/2023 22:32:44] "GET /static/css/bootstrap.main.css HTTP/1.1" 404 -
127.0.0.1 - - [21/Jun/2023 22:32:44] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [21/Jun/2023 22:32:45] "GET /favicon.ico HTTP/1.1" 404 -



العائق الذي واجهني عند العمل على برنامج pycharm هو عدم وجود حزمة ال flask وبالتي قمت بتنزيل هذه الحزمة على البرنامج لنجاح الوصول الى صفحة الويب المطلوبة