



Name: Rahaf maimon mlaika __, Number: __2639____, Submitted To GitHub: _RahafMlaika

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.

Requirements:

- The server should be able to handle multiple client connections concurrently.
- The quiz should consist of a set of pre-defined questions stored on the server.
- Each client should connect to the server and receive the quiz questions.
- Clients should send their answers to the server.
- The server should keep track of the scores for each client.
- 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 3rd-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.

الشرح :

قامت بتخزين الأسئلة التي سوف يطرحها ال server على ال clients في ملف json, ثم قمت باستيراد لمحتويات هذا الملف ضمن كود ال server في متحول اسمه dictionary, ثم قمت ببناء Tcp multi clients server يعمل على العنوان 127.0.0.1 لأنه localhost وهو يتتصت على ال port 8888, ثم تعريف تابع بارامتراتة هي client_socket وعنوان هذا المستخدم و رقمه في تسلسل ال clients المشاركين في الاختبار, ضمن حلقة while قمت بتعريف متحول score لتخزين علامة ال client وبعدها حلقة for للمرور على عناصر ال dictionary الذي خزنت به الأسئلة وتتعامل هذه الحلقة مع ال key وال value حيث أنني سوف أحتاج ال key لأنه يعبر عن السؤال وال value لأنه يعبر عن الجواب الذي سوف نقارن جواب العميل به لمعرفة فيما إذا كان صحيحاً أم لا, يرسل ال server السؤال إلى كل عميل بتعليمة send ومن ثم يستقبل إجابته بتعليمة receive بحجم buffer مقداره 1024 مع ترميز الرسالة المرسله وفك ترميز الرسالة المستقبلية, ثم يتم مقارنة الإجابة مع الجواب الصحيح المخزن في ال dictionary وفي حال كانت صحيحة نضيف علامة إلى ال score.

الجزء التالي في ال code يبدأ بتعريف متحول قيمته الابتدائية هي الواحد يعبر عن رقم العميل في تسلسل الانضمام, ثم تعليمة ال accept التي تعطي ال client_socket و client_addrese, ثم إنشاء thread لكل عميل خاص به, وبعدها إعطاء الأمر start لل thread الذي تم إنشاؤه, وتحديث الرقم التسلسلي للعملاء بإضافة 1.

Multi clients Tcp server code:



Name: Rahaf maimon mlaika __, Number: __2639 ____, Submitted To GitHub: _RahafMlaika

```

1 import socket, threading
2 import json
3 file=open("D:\\project\\network programming\\homework2\\env\\data_for_server.json", 'r')
4 dectionary=json.load(file)
5 file.close()
6
7 server_socket=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8 server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
9 server_socket.bind(("127.0.0.1", 8888))
10 server_socket.listen(5)
11
12 def requests(num_of_client, client_socket, client_address):
13     print("accepting client...", num_of_client, client_address)
14     j=len(dectionary)
15     k=0
16     while True:
17         storing_dic={}
18         score=0
19         for key,value in dectionary.items():
20             client_socket.send(key.encode())
21             client_answer=client_socket.recv(1024).decode()
22             answer=int(client_answer)

```

```

23         storing_dic[key]=[answer]
24         k+=1
25
26         if answer==(dectionary[key]):
27             score+=1
28
29         if k==j:
30             print("client ", num_of_client, " his score ", score)
31             break
32
33     num_of_client+=1
34
35 while True:
36
37     print("waitting for client...")
38     client_socket, client_address=server_socket.accept()
39     client_thread=threading.Thread(target=requests, args=(num_of_client, client_socket, client_address))
40     client_thread.start()
41     num_of_client+=1

```

في كود العميل :

في البداية إنشاء socket للعميل بالاعتماد على Tcp , ثم يحاول هذا العميل الاتصال مع ال server على العنوان ورقم المنفذ المخصصين له , ضمن حلقة while يقوم باستقبال السؤال من ال server وفك ترميزه وإدخال إجابته



Name: Rahaf maimon mlaika __, Number: __2639 ____, Submitted To GitHub: _RahafMlaika

وإرسالها إلى ال server بعد ترميزها، يقوم باختبار ال exception وطباعة الخطأ في حال وجوده ومن ثم يغلق ال socket .

```

1 import socket
2 client_socket=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
3 client_socket.settimeout(100)
4
5 try:
6     client_socket.connect(("127.0.0.1",8888))
7
8     while True:
9         question=client_socket.recv(1024).decode()
10        print(question)
11        answer=input("write your answer here")
12        client_socket.sendall(answer.encode())
13
14 except socket.error as err:
15     print(err)
16
17 finally:
18     client_socket.close()
19

```

```

1 import socket
2 client_socket=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
3 client_socket.settimeout(100)
4
5 try:
6     client_socket.connect(("127.0.0.1",8888))
7
8     while True:
9         question=client_socket.recv(1024).decode()
10        print(question)
11        answer=input("write your answer here")
12        client_socket.sendall(answer.encode())
13
14 except socket.error as err:
15     print(err)
16
17 finally:
18     client_socket.close()
19

```

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

Requirements:



Name: Rahaf maimon mlaika __, Number: __2639____, Submitted To GitHub: _RahafMlaika

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

الشرح:

في البداية قمت بإنشاء web server بالاعتماد على flask

عند الدخول إلى الموقع تظهر واجهة تحتوي على خيارات متعددة كل منها عبارة عن link مع صفحة html, الصفحات مأخوذة من نماذج bootstrap, هذه النماذج مبنية بالاعتماد على html و css.

في حال دخلنا إلى الخيار Home :

سوف تظهر صفحة بعنوان (getting started to html), وهي تحتوي على معلومات عامة عن ال html و ال tags.

في حال دخلنا إلى (getting start) هنا سوف نجد طريقة تضمين تنسيقات css ضمن ملف مكتوب بال html. وعند الدخول إلى (courses) سوف نجد مجالات أخرى يمكن أن يكون المستخدم مهتماً بها مثل تعلم البايثون أو AI أو opencv

Flask web server:

```

1 from flask import Flask, render_template
2 app=Flask(__name__)
3 @app.route('/')
4 def info():
5     return (render_template('index.html'))
6 @app.route('/about')
7 def about():
8     return (render_template('about.html'))
9 @app.route('/blogger')
10 def blogger():
11     return (render_template('blogger.html'))
12 if __name__ == '__main__':
13     app.run(debug=True)
14 
```

Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical
engineering

5th , Network Programming : Homework No2



الجمهورية العربية السورية

الملاذقية - جامعة تشرين

كلية الهندسة الكهربائية والميكانيكية

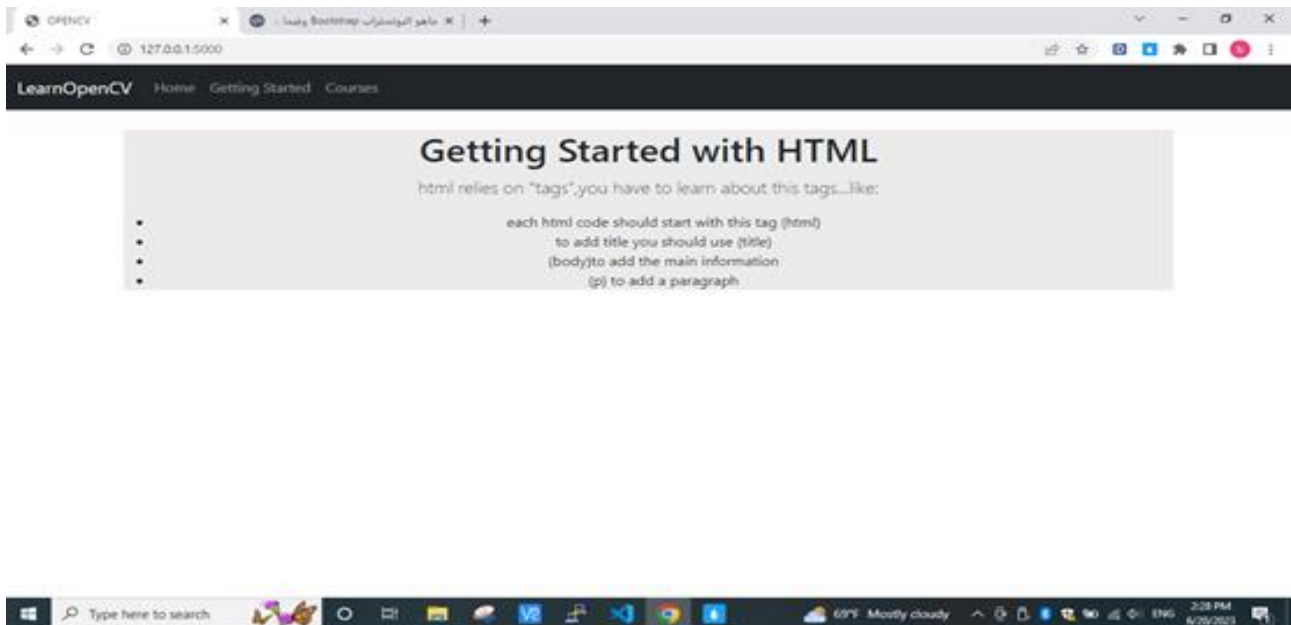
قسم هندسة الاتصالات والإلكترونيات

السنة الخامسة: وظيفة 2 برمجة شبكات

Name: Rahaf maimon mlaika __, Number: __2639 ____, Submitted To GitHub: _RahafMlaika

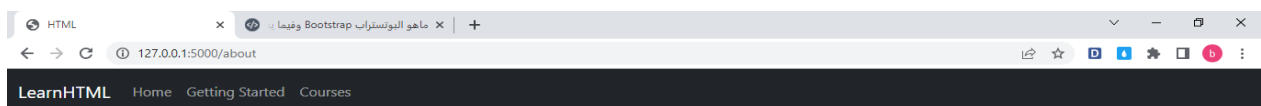
Home :

هذا الخيار هو ارتباط مع الصفحة index.html



Getting started :

هذا الخيار مرتبط مع الصفحة about.html



matching between HTML and CSS simply

when we want to build an html page and make it's style from css file we should make two files, first one is "css file" and the second is the "html file", we need to put all styles in css file as class so we can use it in html file by this link : (link rel="stylesheet" href="styles.css") and this link we should put it in the (head) in html code

Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical
engineering

5th , Network Programming : Homework No2



الجمهورية العربية السورية

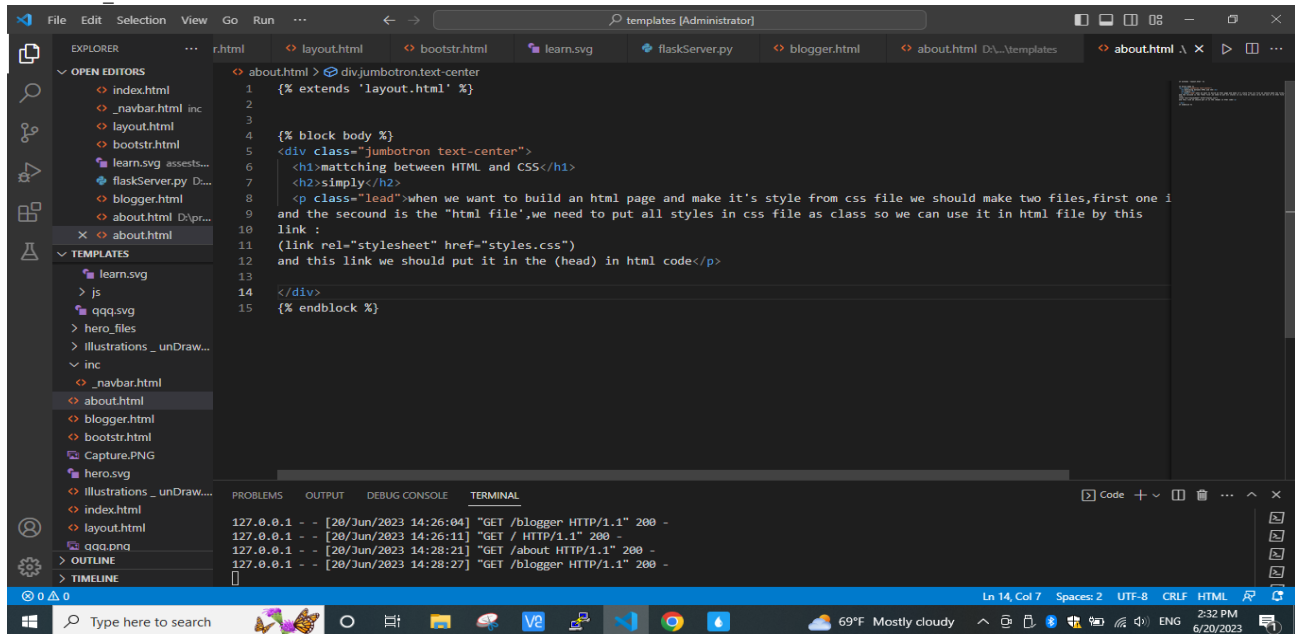
الملاذقية - جامعة تشرين

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والإلكترونيات

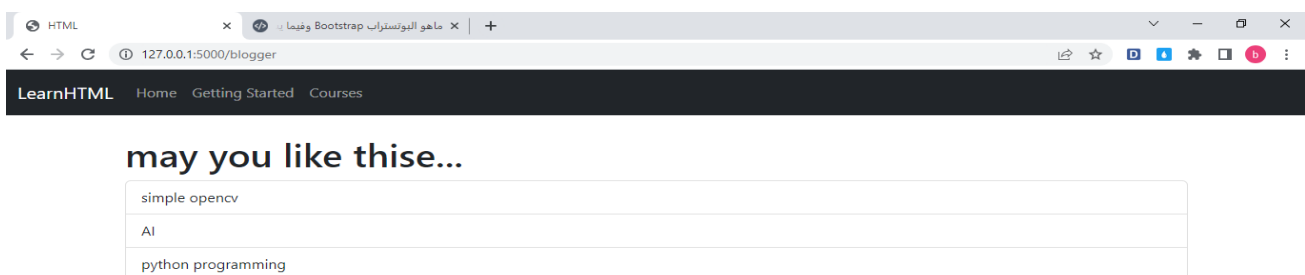
السنة الخامسة: وظيفة 2 برمجة شبكات

Name: Rahaf maimon mlaika __, Number: __2639 ____, Submitted To GitHub: _RahafMlaika



Courses :

هذا الخيار مرتبط مع blogger.html





Name: Rahaf maimon mlaika __, Number: __2639 ____, Submitted To GitHub: _RahafMlaika

```

1  {% extends 'layout.html' %}
2
3
4  {% block body %}
5
6  <h1>may you like thise...</h1>
7  <ul class="list-group">
8  <li class="list-group-item">simple openvc</li>
9  <li class="list-group-item">AI</li>
10 <li class="list-group-item">python programming</li>
11 </ul>
12 {% endblock %}

```

```

127.0.0.1 - - [20/Jun/2023 14:26:04] "GET /blogger HTTP/1.1" 200 -
127.0.0.1 - - [20/Jun/2023 14:26:11] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [20/Jun/2023 14:28:21] "GET /about HTTP/1.1" 200 -
127.0.0.1 - - [20/Jun/2023 14:28:27] "GET /blogger HTTP/1.1" 200 -

```

إضافةً إلى كودين باسم layout و nav الذين يساعدان في إعطاء الشكل العام للموقع

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5   <meta name="description" content="" />
6   <meta name="author" content="" />
7   <meta name="viewport" content="user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, width=device-width" />
8   <title>HTML</title>
9   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" />
10 </head>
11 <body>
12   {% include 'inc/_navbar.html' %}
13   <div class="container">
14     {% block body %}
15     {% endblock %}
16   </div>
17   <script type="text/javascript" src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"></script>
18 </body>
19 </html>

```

```

127.0.0.1 - - [20/Jun/2023 14:26:04] "GET /blogger HTTP/1.1" 200 -
127.0.0.1 - - [20/Jun/2023 14:26:11] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [20/Jun/2023 14:28:21] "GET /about HTTP/1.1" 200 -
127.0.0.1 - - [20/Jun/2023 14:28:27] "GET /blogger HTTP/1.1" 200 -

```




Name: Rahaf maimon mlaika __, Number: __2639 ____, Submitted To GitHub: _RahafMlaika

```

1 <nav class="navbar navbar-expand-md navbar-dark bg-dark mb-4">
2   <div class="container-fluid">
3     <a class="navbar-brand" href="#">LearnHTML</a>
4   </div>
5   <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse" aria-controls=
6     <span class="navbar-toggler-icon"></span>
7   </button>
8   <div class="collapse navbar-collapse" id="navbarCollapse">
9     <ul class="navbar-nav me-auto mb-2 mb-md-0">
10      <li class="nav-item">
11        <a class="nav-link" aria-current="page" href="/">Home</a>
12      </li>
13      <li class="nav-item">
14        <a class="nav-link" href="/about">Getting Started</a>
15      </li>
16      <li class="nav-item">
17        <a class="nav-link" href="/blogger">Courses</a>
18      </li>
19    </ul>
20  </div>
21 </nav>

```

Notes “! important”

- All Needed information and codes to do your homework are presented in Lecture Slides and reference.
- You have to create **nicely formatted** and well-explained PDF document that includes “Question, Explanation, Code, Result”.
- **Similar solutions will be rejected and not accepted.**
- The homework is accepted **until the date of 18/6/2023.**
- **Ensure that all code runs on your pc During the interview, you will be required to present your homework and demonstrate the execution of your code.** Please note that submitting a PDF alone will not be accepted unless it includes evidence of successful code execution on your PC.
- Upload your code to your GitHub account {YourNme_YourNumber_H2}.PDF and related codes, then **send the link to promail0101@gmail.com**