



الجمهورية العربية السورية
وزارة التعليم العالي والبحث العلمي
جامعة تشرين
هندسة الاتصالات والإلكترونيات

برمجة الشبكات الوظيفة الثانية

إعداد الطالب:

زين العابدين حسن بربهان

(2680)

المشرف:

د.مهند عيسى

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.

Server

```

.py x vc.py x client2.py x
import socket, threading
import json

questions = {...}

result = {}

1 usage
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()

1 usage
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)
try:
    ss.bind(('127.0.0.1', 8585))
except OSError as err_msg:
    print(err_msg)
ss.listen(5)
print(" Exam Server is Ready and waiting for clients .")
while True:
    cs, cadd = ss.accept()
    print(f"[+] tcp server is waiting for new connection....' {cadd}")
    client = handle_client_thread(cs, cadd)
    client.start()

```

Client

```

import socket
import json

cs = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
cs.settimeout(5)

try:
    cs.connect(('127.0.0.1', 8585))
    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())
except socket.error as err_msg:
    print(err_msg)
final_score = cs.recv(1024).decode()
print(f"Final score: {final_score}")

```

عند تشغيل السيرفر

```

C:\Users\Windows.10\AppData\Local\Programs\Python\Python39\python.exe C:\Users\Windows.10\Desktop\cv.py
Exam Server is Ready and waiting for clients .
[+] tcp server is waiting for new connection....' ('127.0.0.1', 50413)
[+] tcp server is waiting for new connection....' ('127.0.0.1', 50428)

```

الزبون الثاني

```
C:\Users\Windows.10\AppData\Local\Programs\Python\Python38-32\Scripts>python client2.py
what is the answer of 3*3: 9
what is the answer of 3*5: 15
what is the answer of 3+3 : 6
what is the answer of 31-3 : 28
what is the answer of 3-3 : 0
what is the answer of 3**3 : 27
what is the answer of 4*5 : 20
what is the answer of 9*9 : 81
what is the answer of 4+4 : 8
what is the answer of 5*5 : 25
what is the answer of 3+13 : 16
what is the answer of 2+5 : 7
what is the answer of 8+3 : 11
what is the answer of 3*7 : 27
what is the answer of 3*4: 12
what is the answer of 3*8 : 24
what is the answer of 5*6 : 30
what is the answer of 3-2 : 1
what is the answer of 3+61: 64
what is the answer of 5+9 : 14
Final score: Score: 17/20
```

Process finished with exit code 0

الزبون الأول

```
C:\Users\Windows.10\AppData\Local\Programs\Python\Python38-32\Scripts>python client1.py
what is the answer of 3*3: 9
what is the answer of 3*5: 15
what is the answer of 3+3 : 6
what is the answer of 31-3 : 28
what is the answer of 3-3 : 0
what is the answer of 3**3 : 27
what is the answer of 4*5 : 20
what is the answer of 9*9 : 81
what is the answer of 4+4 : 8
what is the answer of 5*5 : 25
what is the answer of 3+13 : 16
what is the answer of 2+5 : 7
what is the answer of 8+3 : 11
what is the answer of 3*7 : 27
what is the answer of 3*4: 12
what is the answer of 3*8 : 24
what is the answer of 5*6 : 30
what is the answer of 3-2 : 1
what is the answer of 3+61: 64
what is the answer of 5+9 : 14
Final score: Score: 18/20
```

Process finished with exit code 0

شرح الكود الأول:

هذا الكود يمثل خادماً (Server) يقوم بإجراء اختبار (Exam) على المتصلين به. يستخدم البروتوكول TCP/IP لإرسال واستقبال البيانات من وإلى المتصلين. يتم فتح المنفذ 8585 للاستماع على الاتصالات الواردة.

تم تعريف قائمة الأسئلة (Questions) التي سيتم استخدامها في الاختبار، مع الإجابات الصحيحة المقابلة لكل سؤال. يتم إرسال الأسئلة إلى المتصلين ويطلب منهم إرسال الإجابات. يتم حساب النتيجة النهائية لكل متصل عن طريق عدد الإجابات الصحيحة التي قدمها.

تم إنشاء فئتين، الأولى `handle_request` تستخدم لإجراء عملية التحقق من الإجابات المقدمة من المتصل، وتحسب النتيجة النهائية. الثانية `handle_client_thread` تستخدم لإنشاء خيوط (Threads) جديدة لكل متصل يتصل بالخادم.

يتم إنشاء كائن `socket` باستخدام `AF_INET` و `SOCK_STREAM`، والذي يتم استخدامه لإنشاء الاتصالات بين المتصلين والخادم. تم استخدام استثناء الـ `OSError` للتحقق من وجود أية أخطاء في عملية الربط. أخيراً، يتم استدعاء `accept` للانتظار على المتصلين الجدد وقبول الاتصالات الواردة. عند وصول متصل جديد، تم إنشاء خيط جديد للتعامل مع المتصل.

شرح الكود الثاني:

هذا الكود يمثل عميل (Client) يتصل بالخادم السابق ويتفاعل معه لإجراء الاختبار.

تم استخدام مكتبة `socket` لإنشاء كائن `Socket`، وهو ما سيستخدم لإنشاء الاتصال بالخادم. تم استخدام `AF_INET` و `SOCK_STREAM` للاتصال باستخدام بروتوكول TCP/IP.

يتم تعيين قيمة الـ `settimeout` إلى 5 ثوانٍ، وهو ما يعني أنه إذا لم يتم الرد على الاتصال خلال هذه الفترة، سيتم إثارة استثناء `socket.timeout` وإيقاف الاتصال.

بعد الاتصال بالخادم، يتم استقبال عدد الأسئلة الذي سيتم طرحها في الاختبار، ثم يتم طرح الأسئلة على المستخدم من خلال استخدام `input` وإرسال الإجابات المقدمة من المستخدم إلى الخادم باستخدام `sendall`.

يتم استقبال النتيجة النهائية للمتصل من الخادم باستخدام `recv` ويتم طباعة النتيجة النهائية للمستخدم. يتم التحقق من وجود أية أخطاء في عملية الاتصال باستخدام استثناء `socket.error` وإظهار رسالة الخطأ المناسبة إذا حدث خطأ.

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, render_template

app = Flask(__name__)

@app.route('/')
def index():
    return render_template("index.html")

@app.route('/about')
def about():
    return render_template("about.html")

@app.route('/contacts')
def contacts():
    return render_template("contacts.html")

@app.route('/certificates')
def certificates():
    return render_template('certificates.html')

if __name__ == "__main__":
    app.run(debug = True)
```

شرح الكود :

هذا الكود يمثل تطبيق Flask الذي يقدم عددًا من الصفحات التي يمكن الوصول إليها عن طريق العناوين المحددة.

تم استيراد الوحدات `Flask` و `render_template` من إطار عمل Flask. تم إنشاء كائن Flask باستخدام اسم التطبيق `__name__`، ويتم تعيين كائن `app` إلى هذا الكائن.

يتم استخدام الزخم `app.route@` لتحديد عناوين URL المرتبطة بكل صفحة. على سبيل المثال، يتم استخدام `app.route@` لتحديد عنوان URL للصفحة الرئيسية، و `app.route('/about')@` لصفحة المعلومات، وهكذا.

تم إنشاء دالة `index` للصفحة الرئيسية و `about` و `contacts` و `certificates` لصفحات الأخرى. يتم استخدام `render_template` للعثور على ملفات HTML المرتبطة بكل صفحة وإرجاعها كمحتوى استجابة.

أخيرًا، يتم استخدام `__main__ == "__if __name__":` للتأكد من أن الكود يتم تنفيذه فقط عند تشغيل الملف كبرنامج مستقل، ويتم تشغيل الخادم باستخدام `app.run`، مع تمكين وضع التصحيح `debug=True` لتشغيل الخادم في وضع التصحيح.

ويعطي عند التشغيل :

```
IPython 7.29.0 -- An enhanced Interactive Python.
In [1]:
In [1]: runfile('C:/Users/Windows.10/Desktop/net_hom/zein/main.py', wdir='C:/Users/Windows.10/Desktop/net_hom/zein')
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
```

محتوى صفحات الويب:

index.html

```
> Users > Windows.10 > Desktop > net_hom > zein > templates > <> index.html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>CV - Index</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <!-- Bootstrap CSS -->
8      <link rel="stylesheet" href={{url_for('static', filename="css/bootstrap.min.css')}}>
9  </head>
10 <body>
11     <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
12         <a class="navbar-brand" href="#">CV</a>
13         <ul class="navbar-nav">
14             <li class="nav-item">
15                 <li class="nav-item">
16                     <a class="nav-link" href={{url_for('index')}}>Index</a>
17                 </li>
18                 <a class="nav-link" href={{url_for('about')}}>About</a>
19             </li>
20             <li class="nav-item">
21                 <a class="nav-link" href={{url_for('contacts')}}>Contacts</a>
22             </li>
23             <li class="nav-item">
24                 <a class="nav-link" href={{url_for('certificates')}}>certificates</a>
25             </li>
26         </ul>
27     </nav>
28     <div class="container mt-3">
29         <h1>Welcome to my CV</h1>
30         <p>This is the index page of my CV. Please click on the links above to navigate to other pages.</p>
31     </div>
```

About.html

```
about.html X
> Users > Windows.10 > Desktop > net_hom > zein > templates > <> about.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>CV - About</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <!-- Bootstrap CSS -->
8      <link rel="stylesheet" href={{url_for('static', filename="css/bootstrap.min.css')}}>
9  </head>
10 <body>
11     <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
12         <a class="navbar-brand" href="#">CV</a>
13         <ul class="navbar-nav">
14             <li class="nav-item">
15                 <li class="nav-item">
16                     <a class="nav-link" href={{url_for('index')}}>Index</a>
17                 </li>
18                 <a class="nav-link" href={{url_for('about')}}>About</a>
19             </li>
20             <li class="nav-item">
21                 <a class="nav-link" href={{url_for('contacts')}}>Contacts</a>
22             </li>
23             <li class="nav-item">
24                 <a class="nav-link" href={{url_for('certificates')}}>certificates</a>
25             </li>
26         </ul>
27     </nav>
28     <div class="container mt-3">
29         <h1>About me</h1>
30         <img src={{url_for('static', filename="5.jpg')}} alt="My Picture" class="img-fluid rounded-circle mb-3" width="200">
31         <p>Here's some information about me:</p>
```


contacts.html

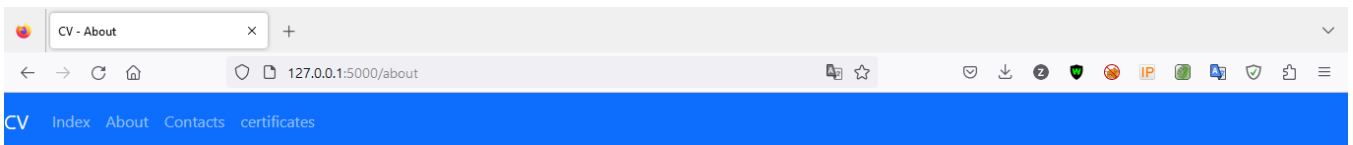
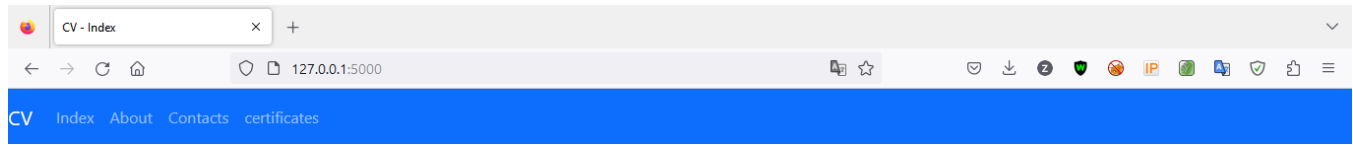
```
<> about.html <> contacts.html X
C: > Users > Windows.10 > Desktop > net_hom > zein > templates > <> contacts.html > ...
1  |<!DOCTYPE html>
2  |<html lang="en">
3  |<head>
4  |  <title>CV - Contacts</title>
5  |  <meta charset="utf-8">
6  |  <meta name="viewport" content="width=device-width, initial-scale=1">
7  |  <!-- Bootstrap CSS -->
8  |  <link rel="stylesheet" href="{{url_for('static', filename="css/bootstrap.min.css")}}">
9  |</head>
10 |<body>
11 |  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
12 |    <a class="navbar-brand" href="#">CV</a>
13 |    <ul class="navbar-nav">
14 |      <li class="nav-item">
15 |        <li class="nav-item">
16 |          <a class="nav-link" href="{{url_for('index')}}>Index</a>
17 |        </li>
18 |        <a class="nav-link" href="{{url_for('about')}}>About</a>
19 |      </li>
20 |      <li class="nav-item">
21 |        <a class="nav-link" href="{{url_for('contacts')}}>Contacts</a>
22 |      </li>
23 |      <li class="nav-item">
24 |        <a class="nav-link" href="{{url_for('certificates')}}>certificates</a>
25 |      </li>
26 |    </ul>
27 |  </nav>
28 |  <div class="container mt-3">
29 |    <h1>Contact me</h1>
30 |    <p>Here's how you can contact me:</p>
31 |    <ul>
```

certificates.html

```
<> about.html <> contacts.html <> certificates.html X
C: > Users > Windows.10 > Desktop > net_hom > zein > templates > <> certificates.html > ...
1  |<!DOCTYPE html>
2  |<html lang="en">
3  |<head>
4  |  <title>CV - certificates</title>
5  |  <meta charset="utf-8">
6  |  <meta name="viewport" content="width=device-width, initial-scale=1">
7  |  <!-- Bootstrap CSS -->
8  |  <link rel="stylesheet" href="{{url_for('static', filename="css/bootstrap.min.css")}}">
9  |</head>
10 |<body>
11 |  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
12 |    <a class="navbar-brand" href="#">CV</a>
13 |    <ul class="navbar-nav">
14 |      <li class="nav-item">
15 |        <li class="nav-item">
16 |          <a class="nav-link" href="{{url_for('index')}}>Index</a>
17 |        </li>
18 |        <a class="nav-link" href="{{url_for('about')}}>About</a>
19 |      </li>
20 |      <li class="nav-item">
21 |        <a class="nav-link" href="{{url_for('contacts')}}>Contacts</a>
22 |      </li>
23 |      <li class="nav-item">
24 |        <a class="nav-link" href="{{url_for('certificates')}}>certificates</a>
25 |      </li>
26 |    </ul>
27 |  </nav>
28 |  <div class="container mt-3">
29 |    <h1>My certificates</h1>
30 |    <p>These are some of the certificates I have :</p>
31 |    <div class="row">
```

الآن نذهب للمتصفح ونكتب 127.0.0.1:5000

فيظهر:



About me



Here's some information about me:

- Name: Zein Alabden Hasan Barbahan
- Email: zeinbarbahan@gmail.com
- Phone: +963991874061
- Education: Communication and electronics engineering
- Skills: I have experience in communications networks. I have strong skills and extensive knowledge in the design, development and maintenance of communications networks. I have an excellent ability to understand complex technical processes and solve problems that you encounter. Have experience using the latest networking technologies, including wireless networks, wired networks, and their connected devices.

