Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No2

الجمهورية العربية السورية

اللاذقية -جامعـــة تشريـــن

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة2 برمجة شبكات

Name:_ _____, Number:_ _____, Submitted To GitHub:_ _____

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:

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 3thd-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 client.py

server.py > handle_c

```python
import socket,threading

quiz = {
    'BSC performs all the softer hand-over for MSs moving between BTSs in its control.': 'f',
    'Hand over may result in call dropping.': 't',
    'Co-channel reuse ratio depends upon the raduises of the co-channel cells.': 't',
    'There is need to use any mobility managment between in progress calls inside the cell.': 'f',
    'Slow frequency hopper is a part of BSC block diagram.': 'f',
    'The frequency hopping used in GSM may be cyclic hopping.': 't',
    'Any cellular operation starts using BCCH.': 'f',
    'Frequency correction burst contains S bits.': 'f',
    'In GSM system, the access burst has the longest guard bits.': 't',
    'Burst of all users must reach the BS at the same time.': 'f',
    'Burst in GSM can have a different length.': 'f',
    'Any registered user must have a record in HLR.': 't',
    'Using TRAU does not affect the performance of the air interface.': 't',
    'SRES is an essential data of GSN SIM card.': 'f',
    'Authentication keys are not a permanent SIM data.': 'f',
    'GSM specifies four databases.': 't',
    'For GSM system, TDMA is used on the Abis interface.': 'f',
    'In cellular system radio units are housed in BTSs.': 't',
    'Both MAHO and NCHO can be used in GSM.': 'f',
    'A HO operation can be initiated even if the MS is still unmoved.': 't'
}

c_scores = {}

def handle_c(client_soc, client_address):
    try:
        client_soc.send(str(len(quiz)).encode())

        for question in quiz:
            client_soc.send(question.encode())

            client_ans = client_soc.recv(1024).decode().strip()

            if client_ans.lower() == quiz[question].lower():
                c_scores[client_address] = c_scores.get(client_address, 0) + 1
```

Name:_ _____, Number:_ _____, Submitted To GitHub:_ _____

```python

def start_ser():
    server_soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    server_address = ('localhost', 2323)
    server_soc.bind(server_address)

    server_soc.listen(5)
    print("The server has been initiated and is awaiting connections.")

    while True:
        client_socket, client_address = server_soc.accept()
        print(f"Connected to {client_address}")

        client_thread = threading.Thread(target=handle_c, args=(client_socket, client_address))
        client_thread.start()

if __name__ == '__main__':
    start_ser()


```

يتم استيراد وحدة socket للتعامل مع الاتصالات .

يتم استيراد وحدة threading لتمكين التنفيذ المتزامن لعدة اتصالات عميل.

يتم تعريف القاموس quiz لتخزين مجموعة من أسئلة الاختبار كمفاتيح وإجاباتها المقابلة كقيم.

يتم استخدام القاموس c_scores لتتبع درجات كل عميل.

يتم تعريف وظيفة handle_c لمعالجة اتصال كل عميل. تستقبل هذه الوظيفة مأخذين وهما مأخذ عميل socket وعنوانه.

داخل وظيفة handle_c، يتم إرسال طول الاختبار إلى socket العميل، مما يشير إلى عدد الأسئلة.

يتم تكرار أسئلة الاختبار، ويتم إرسال كل سؤال إلى socket العميل.

يتم استقبال إجابة العميل ومقارنتها بالإجابة الصحيحة في قاموس quiz. إذا كانت الإجابة صحيحة، يتم زيادة درجة العميل.

بعد معالجة جميع الأسئلة، يتم إرسال درجة العميل إلى socket العميل.

في حالة حدوث استثناء ConnectionAbortedError، يتم التعامل مع الاستثناء وطباعة رسالة تشير إلى أن اتصال العميل تم إلغاؤه.

يتم إغلاق socket العميل ويتم طباعة رسالة تشير إلى أن العميل قد فصل الاتصال.

يتم تعريف وظيفة start_ser لبدء السيرفر.

يتم إنشاء socket سيرفر وربطه بعنوان محدد (هنا اخذنا 'localhost' على المنفذ 2323) والاستماع إلى الاتصالات الواردة.

يتم طباعة رسالة تشير إلى أن السيرفر قد تم تشغيله وجاهز للاستقبال.

داخل حلقة تكرار لا نهاية، يقوم السيرفر بقبول اتصالات العملاء، ويقوم بطباعة رسالة تشير إلى أن العميل قد قام بالاتصال، ثم يبدأ thread جديدًا لمعالجة اتصال العميل هذا.

Name: , Number: , Submitted To GitHub:

server.py client.py

client.py > ...

```python
import socket

def start_cli():
    server_host = 'localhost'
    server_port = 2323

    client_soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    try:
        client_soc.connect((server_host, server_port))
        print(f"Connected to server {server_host}:{server_port}")

        num_questions = int(client_soc.recv(1024).decode())

        for i in range(num_questions):
            question = client_soc.recv(1024).decode()

            answer = input(f"{question}: ")

            client_soc.sendall(answer.encode())

        final_score = client_soc.recv(1024).decode()
        print(f"Final score: {final_score}")

    except ConnectionRefusedError:
        print("The connection to the server could not be established.")
    finally:
        client_soc.close()

if __name__ == '__main__':
    start_cli()

```

يتم استيراد المكتبة socket.

يتم تعريف وظيفة start_cli التي تقوم ببدء عملية العميل.

يتم تعيين مضيف السيرفر ومنفذه في المتغيرات server_host و server_port على التوالي.

يتم إنشاء كائن socket للعميل باستخدام socket.socket(socket.AF_INET, socket.SOCK_STREAM)، حيث يتم استخدام AF_INET لتعيين عائلة البروتوكول إلى IPv4 و SOCK_STREAM لتحديد استخدام TCP في التواصل.

يتم استدعاء دالة connect() على العميل للاتصال بالسيرفر باستخدام عنوان المضيف ومنفذ السيرفر المعينين سابقًا.

إذا تم بنجاح إنشاء الاتصال، يتم طباعة رسالة تفيد بأن العميل قد تم ربطه بالسيرفر.

يتم استقبال عدد الأسئلة من الخادم عن طريق استقبال البيانات من العميل وتحويلها إلى عدد صحيح باستخدام int().

يتم تطبيق حلقة for للحصول على الأسئلة من الخادم وطباعتها واستلام إجابات العميل.

يتم استقبال الدرجة النهائية من الخادم عن طريق استقبال البيانات من الكائن العميل وتحويلها إلى سلسلة نصية باستخدام decode().

يتم طباعة الدرجة النهائية للعميل.

إذا حدث استثناء ConnectionRefusedError، يتم طباعة رسالة تفيد بأن الاتصال بالخادم لم يتم بنجاح.

Name:_ _____, Number:_ _____, Submitted To GitHub:_ _____

في النهاية، يتم إغلاق كائن العميل باستخدام close() لإنهاء الاتصال.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

opping.                                          opping.
Enter your answer: f                             Enter your answer: f
Any cellular operation starts using BCCH.        Any cellular operation starts using BCCH.
Enter your answer: f                             Enter your answer: f
Frequency correction burst contains S bits.      Frequency correction burst contains S bits.
Enter your answer: f                             Enter your answer: f
In GSM system, the access burst has the longest g In GSM system, the access burst has the longest g
uard bits.                                       uard bits.
Enter your answer: f                             Enter your answer: f
Burst of all users must reach the BS at the same Burst of all users must reach the BS at the same
time.                                            time.
Enter your answer: f                             Enter your answer: f
Burst in GSM can have a different length.        Burst in GSM can have a different length.
Enter your answer: f                             Enter your answer: f
Any registered user must have a record in HLR.   Any registered user must have a record in HLR.
Enter your answer: f                             Enter your answer: f
Using TRAU does not affect the performance of the Using TRAU does not affect the performance of the
 air interface.                                   air interface.
Enter your answer: f                             Enter your answer: f
SRES is an essential data of GSN SIM card.       SRES is an essential data of GSN SIM card.
Enter your answer: f                             Enter your answer: f
Authentication keys are not a permanent SIM data. Authentication keys are not a permanent SIM data.

Enter your answer: f                             Enter your answer: f
GSM specifies four databases.                    GSM specifies four databases.
Enter your answer: f                             Enter your answer: f
For GSM system, TDMA is used on the Abis interfac For GSM system, TDMA is used on the Abis interfac
e.                                               e.
Enter your answer: f                             Enter your answer: f
In cellular system radio units are housed in BTSs In cellular system radio units are housed in BTSs
.                                                .
Enter your answer: f                             Enter your answer: f
Both MAHO and NCHO can be used in GSM.           Both MAHO and NCHO can be used in GSM.
Enter your answer: f                             Enter your answer: f
A HO operation can be initiated even if the MS is A HO operation can be initiated even if the MS is
 still unmoved.                                   still unmoved.
Enter your answer: f                             Enter your answer: f
Your score: 12/20                                Your score: 8/20
```

Name:_ _____, Number:_ _____, Submitted To GitHub:_ _____

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.

server.py client.py index.html

web > templates > index.html > ...

```
<!DOCTYPE html>
<html>
<head>
    <title>Cellular Systems Learning</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
    <link rel="stylesheet" href="static/css/style.css">
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="{{ url_for('index') }}">Home</a></li>
                <li><a href="{{ url_for('Cellular_Generations') }}">Cellular Generations</a></li>
                <li><a href="{{ url_for('about') }}">About</a></li>
                <li><a href="{{ url_for('contact') }}">Contact</a></li>
            </ul>
        </nav>
    </header>

    <div class="container">
        <h1 class="heading">Welcome to Cellular Systems Learning</h1>
        <p>Embark on a fascinating journey to explore the evolution of cellular systems across generations.</p>
        <p>Discover how cellular technology has transformed the way we communicate and connect with each other.</p>
        <p>From the early analog systems to the latest 5G networks, dive deep into the technological advancements and concepts that power modern
        <p>Learn about the key features, benefits, and limitations of each cellular generation, and gain insights into the future of mobile commu
        <a href="{{ url_for('Cellular_Generations') }}" class="btn btn-primary">Get Started</a>
    </div>


</body>
</html>
```

Name: , Number: , Submitted To GitHub:

server.py client.py about.html

web > templates > about.html > ...

```
<!DOCTYPE html>
<html>
<head>
    <title>About - Cellular Systems Learning</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
    <link rel="stylesheet" href="static/css/style.css">
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="{{ url_for('index') }}">Home</a></li>
                <li><a href="{{ url_for('Cellular_Generations') }}">Cellular Generations</a></li>
                <li><a href="{{ url_for('about') }}">About</a></li>
                <li><a href="{{ url_for('contact') }}">Contact</a></li>
            </ul>
        </nav>
    </header>

    <div class="container">
        <h1>Discover the Evolution of Cellular Systems</h1>
        <p>Uncover the technological advancements that shaped the way we communicate and connect.</p>
        <p>Cellular Systems Learning provides an immersive experience to explore the different generations of cellular networks, from the pioneer
        <p>Immerse yourself in the rich history, key features, and transformative impact of each generation. Gain insights into how these systems
        <p>Embark on a journey through time and technology, and deepen your understanding of cellular systems.</p>
    </div>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

server.py client.py Cellular_Generations.html

web > templates > Cellular_Generations.html > html > body > div.container

```
<!DOCTYPE html>
<html>
<head>
    <title>Cellular Systems Learning - Cellular Generations</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
    <link rel="stylesheet" href="static/css/style.css">
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="{{ url_for('index') }}">Home</a></li>
                <li><a href="{{ url_for('Cellular_Generations') }}">Cellular Generations</a></li>
                <li><a href="{{ url_for('about') }}">About</a></li>
                <li><a href="{{ url_for('contact') }}">Contact</a></li>
            </ul>
        </nav>
    </header>

    <div class="container">
        <h1>Cellular Generations</h1>

        <h2>1G - Analog Cellular</h2>
        <p>The first generation of cellular systems introduced analog technology. It enabled basic voice communication and had limited capacity ar

        <h2>2G - Digital Cellular</h2>
        <p>The second generation marked the shift from analog to digital technology. It introduced digital voice transmission and enabled SMS (Sh

        <h2>3G - Mobile Broadband</h2>
        <p>The third generation brought significant improvements, including higher data transfer rates and the ability to access the internet. It

        <h2>4G - LTE</h2>
        <p>The fourth generation, also known as LTE (Long-Term Evolution), offered faster data speeds, low latency, and improved multimedia suppor

        <h2>5G - Next-Generation Connectivity</h2>
        <p>The fifth generation is the latest advancement in cellular technology. 5G networks provide ultra-fast speeds, extremely low latency, ar
    </div>
</body>
</html>
```

Name: , Number: , Submitted To GitHub:

server.py client.py Cellular_Generations.html contact.html

web > templates > contact.html > ...

```
<!DOCTYPE html>
<html>
<head>
    <title>Contact Us</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
    <link rel="stylesheet" href="static/css/style.css">
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="{{ url_for('index') }}">Home</a></li>
                <li><a href="{{ url_for('Cellular_Generations') }}">Cellular Generations</a></li>
                <li><a href="{{ url_for('about') }}">About</a></li>
                <li><a href="{{ url_for('contact') }}">Contact</a></li>
            </ul>
        </nav>
    </header>

    <main>
        <section>
            <div class="container">
                <h1>Contact Information</h1>
                <p>If you have any questions or would like to learn more about cellular systems, please contact us through the following channels:
                <ul>
                    <li>Email: info@cellularsystemslearning.com</li>
                    <li>Phone: +963 999-555-666</li>
                    <li>Address: 123 Main Street, City, Country</li>
                </ul>
                <footer>
                    <p>&copy; 2023 Cellular Systems Learning. All rights reserved.</p>
                </footer>
            </div>
        </section>
    </main>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

Name: , Number: , Submitted To GitHub:

server.py client.py Cellular_Generations.html contact.html style.css

web > static > css > # style.css > body

```css
body {
    font-family: Arial, sans-serif;
    background-color: #f5f5f5;
}



header {
    background-color: #f8f8f8;
    padding: 10px;
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

nav ul li {
    display: inline;
    margin-right: 10px;
}

nav ul li a {
    text-decoration: none;
    color: #333333;
    font-weight: bold;
}

h1 {
    color: #3366cc;
    margin-top: 0;
}

h2 {
    color: #cc3366;
}

```

Name:_ _____ , Number: _____ , Submitted To GitHub: _____

app.py > index

```python
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('/Cellular_Generations')
def Cellular_Generations():
    return render_template('Cellular_Generations.html')
@app.route('/contact')
def contact():
    return render_template('contact.html')

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

```

يتم إنشاء تطبيق Flask باستخدام الكود app = Flask(__name__).

يتم تمرير __name__ كمعامل لتحديد اسم التطبيق وتحديد موقع ملفات html.

يتم تعريف المسارات (routes) باستخدام المزخرف @app.route

المسار /template يعود إلى الصفحة الرئيسية ويتم تعيينه لدالة home().

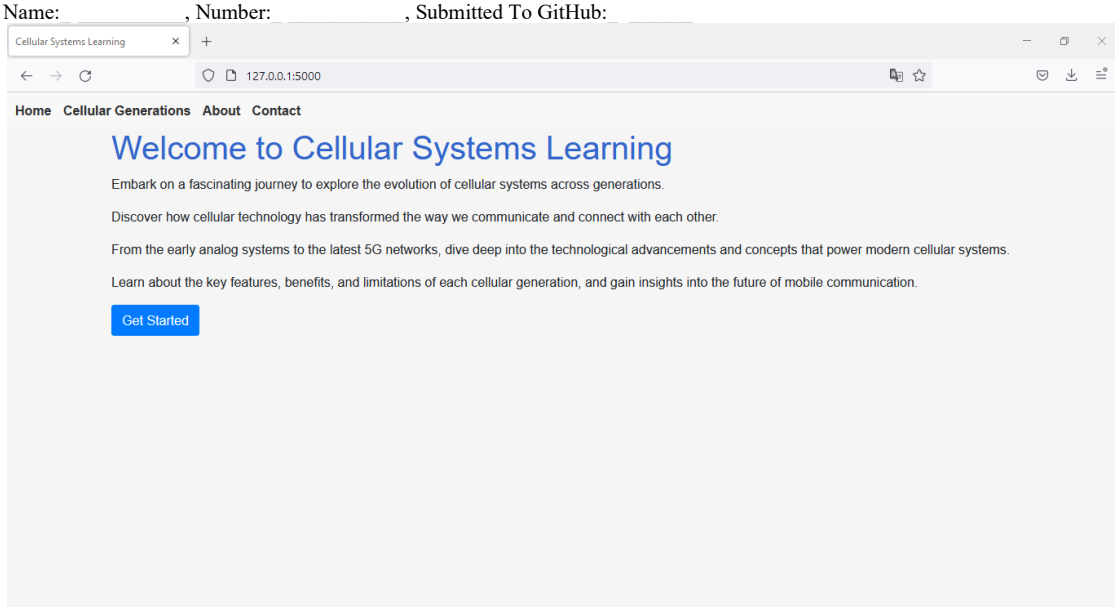
المسار /template/about يعود إلى صفحة "about" ويتم تعيينه لدالة about().

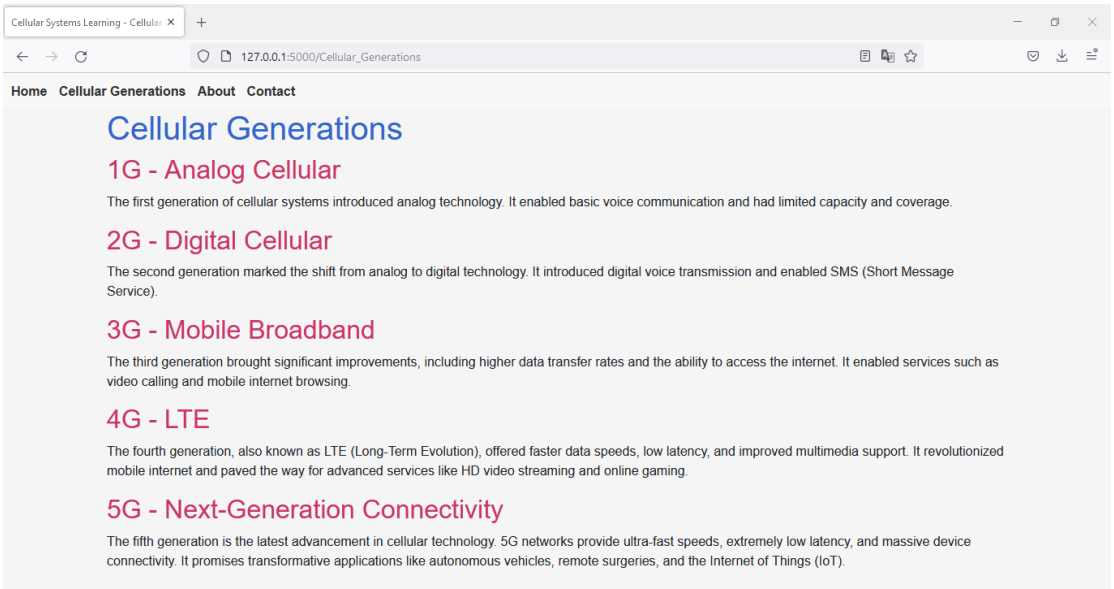
المسار /template/contact يعود إلى صفحة " contact" ويتم تعيينه لدالة contact().

إذا كان البرنامج يتم تشغيله مباشرة عن طريق تشغيل البرنامج الرئيسي ، فإنه يشغل التطبيق بتفعيل وضع التصحيح (debug mode) بواسطة الأمر app.run(debug=True).

يتم التشغيل على port 5000

Name: _____, Number: _____, Submitted To GitHub: _____





Name: _____, Number: _____, Submitted To GitHub: _____