



Name:Nour Ahmad, Number:2673, Submitted To GitHub:@NourMAhmad

Name:Odai Afif, Number:2997, Submitted To GitHub: _____

Name: Haidera Ibrahim, Number:2701, Submitted To GitHub: _____

Second Network Programming Homework

Question 1: Bank ATM Application with TCP Server/Client and Multi-threading

Project Description:

Build a TCP server and client Bank ATM application using Python. The server should handle multiple client connections simultaneously using multi-threading. The application should allow clients to connect, perform banking operations (such as check balance, deposit, and withdraw), and receive their updated account status upon completion.

Requirements:

- A. The server should be able to handle multiple client connections concurrently.
- B. The server should maintain a set of pre-defined bank accounts with balances.
- C. Each client should connect to the server and authenticate with their account details.
- D. Clients should be able to perform banking operations: check balance, deposit money, and withdraw money.
- E. The server should keep track of the account balances for each client.
- F. At the end of the session, the server should send the final account balance to each client.

Guidelines:

- Use Python's socket module without third-party packages.
- Implement multi-threading to handle multiple client connections concurrently.
- Store the account details and balances on the server side.

Notes:

- Write a brief report describing the design choices you made and any challenges faced during implementation.
- You can choose to create a TCP Server/Client Bank ATM application or any other appropriate application that fulfills all requirements.



```

1  import socket
2  import threading
3  # Sample account details stored on the server
4  accounts = {
5      'user1': {'balance': 1000, 'password': 'pass1'},
6      'user2': {'balance': 2000, 'password': 'pass2'}
7  }
8  # Function to handle client requests
9  def handle_client(client_socket):
10     # Authentication
11     client_socket.send("Enter username: ".encode())
12     username = client_socket.recv(1024).decode().strip()
13     client_socket.send("Enter password: ".encode())
14     password = client_socket.recv(1024).decode().strip()
15     if username in accounts and accounts[username]['password'] == password:
16         client_socket.send("Authentication successful.".encode())
17     else:
18         client_socket.send("Invalid credentials. Closing connection.".encode())
19         client_socket.close()
20     return
21     while True:
22         # Receive client requests
23         request = client_socket.recv(1024).decode().strip()
24         if request == "check_balance":
25             balance = accounts[username]['balance']
26             client_socket.send(f"Your balance is {balance}".encode())
27         elif request.startswith("deposit"):
28             amount = float(request.split()[1])
29             accounts[username]['balance'] += amount
30             client_socket.send("Deposit successful.".encode())
31         elif request.startswith("withdraw"):
32             amount = float(request.split()[1])
33             if accounts[username]['balance'] >= amount:
34                 accounts[username]['balance'] -= amount
35                 client_socket.send("Withdrawal successful.".encode())
36         else:
37             client_socket.send("Insufficient funds.".encode())
38         elif request == "exit":
39             client_socket.send("Exiting.".encode())
40             break
41         else:
42             client_socket.send("Invalid request.".encode())
43     # Close client connection
44     client_socket.close()
45 # Main server function
46 def main():
47     server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
48     server.bind(('127.0.0.1', 5555))
49     server.listen(5)
50     print("[*] Listening on 127.0.0.1:5555")
51     while True:
52         client_socket, addr = server.accept()
53         print(f"[*] Accepted connection from {addr[0]}:{addr[1]}")
54         # Create a new thread to handle client requests
55         client_handler = threading.Thread(target=handle_client, args=(client_socket,))
56         client_handler.start()
57 if __name__ == "__main__":
58     main()
59 
```



```
1 import socket
2 def main():
3     client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4     client.connect(('127.0.0.1', 5555))
5     # Authentication
6     response = client.recv(1024).decode()
7     print(response, end='', flush=True)
8     username = input()
9     client.send(username.encode())
10    response = client.recv(1024).decode()
11    print(response, end='', flush=True)
12    password = input()
13    client.send(password.encode())
14    response = client.recv(1024).decode()
15    print(response, flush=True)
16    while True:
17        print("Options:")
18        print("1. Check Balance 2. Deposit 3. Withdraw 4. Exit")
19        choice = input("Enter your choice: ")
20        if choice == '1':
21            client.send("check_balance".encode())
22            balance = client.recv(1024).decode()
23            print(balance, flush=True)
24        elif choice == '2':
25            amount = float(input("Enter amount to deposit: "))
26            client.send(f"deposit {amount}".encode())
27            response = client.recv(1024).decode()
28            print(response, flush=True)
29        elif choice == '3':
30            amount = float(input("Enter amount to withdraw: "))
31            client.send(f"withdraw {amount}".encode())
32            response = client.recv(1024).decode()
33            print(response, flush=True)
```

```
34         elif choice == '4':
35             client.send("exit".encode())
36             print("Exiting.", flush=True)
37             break
38         else:
39             print("Invalid choice.", flush=True)
40     client.close()
41 if __name__ == "__main__":
42     main()
43
```



```
Enter username: user1
Enter password: pass1
Authentication successful.
Options:
1. Check Balance 2. Deposit 3. Withdraw
4. Exit
Enter your choice: 1
Your balance is 1000
Options:
1. Check Balance 2. Deposit 3. Withdraw
4. Exit
Enter your choice: 2
Enter amount to deposit: 300
Deposit successful.
Options:
1. Check Balance 2. Deposit 3. Withdraw
4. Exit
Enter your choice: 4
Exiting.
```

```
Enter username: user2
Enter password: pass2
Authentication successful.
Options:
1. Check Balance 2. Deposit 3. Withdraw 4.
Exit
Enter your choice: 1
Your balance is 2000
Options:
1. Check Balance 2. Deposit 3. Withdraw 4.
Exit
Enter your choice: 2
Enter amount to deposit: 500
Deposit successful.
Options:
1. Check Balance 2. Deposit 3. Withdraw 4.
Exit
Enter your choice: 4
Exiting.
```

```
[*] Listening on 127.0.0.1:5555
[*] Accepted connection from 127.0.0.1:52605
[*] Accepted connection from 127.0.0.1:52613
```

**السؤال الأول:****السيرفر:**

إنشاء خادم بنكي باستخدام مكتبة socket و threading في بايثون. يحتوي الخادم على تفاصيل حسابات مخزنة مسبقاً، حيث لكل حساب اسم مستخدم، رصيد وكلمة مرور .

الدالة الأساسية للتعامل مع طلبات العملاء تتضمن أولاً عملية المصادقة، حيث يطلب من العميل إدخال اسم المستخدم وكلمة المرور. إذا كانت البيانات صحيحة، يتم إعلام العميل بنجاح المصادقة، وإذا كانت خاطئة، يتم إغلاق الاتصال.

بمجرد المصادقة، يمكن للعميل إرسال طلبات مختلفة:

- طلب "check_balance" للتحقق من الرصيد، حيث يرسل الخادم الرصيد الحالي.
- طلب "deposit" متبوعاً بالمبلغ للإيداع، حيث يتم إضافة المبلغ إلى الرصيد وإعلام العميل بنجاح العملية.
- طلب "withdraw" متبوعاً بالمبلغ للسحب، حيث يتم التحقق من توفر الرصيد الكافي، ثم خصم المبلغ من الرصيد إذا كان كافياً، أو إعلام العميل بنقص الرصيد.
- طلب "exit" لإنهاء الاتصال، حيث يتم إغلاق الاتصال مع العميل.

الخادم يتم تشغيله في الدالة الرئيسية، حيث يتم إنشاء مقبس الشبكة وربطه بعنوان IP المحلي والمنفذ 5555. يبدأ الخادم بالاستماع لطلبات الاتصال، وعند قبول اتصال من عميل جديد، يتم إنشاء خيط جديد للتعامل مع طلبات هذا العميل، مما يسمح بالتعامل مع عدة عملاء في وقت واحد.

العميل:

إنشاء عميل للاتصال بخادم بنكي باستخدام مكتبة socket في بايثون. يبدأ البرنامج بإنشاء مقبس شبكة للعميل ثم يتصل بالخادم على العنوان المحلي والمنفذ 5555.

عند الاتصال بالخادم، يتم استقبال رسالة تطلب إدخال اسم المستخدم وكلمة المرور، حيث يقوم العميل بإدخالها وإرسالها إلى الخادم للتحقق من صحة بياناته. إذا كانت المصادقة ناجحة، يمكن للعميل متابعة العمليات البنكية.



يتم عرض قائمة بالخيارات المتاحة للعميل :

- للتحقق من الرصيد، يقوم العميل بإرسال طلب "check_balance" للخادم، ويستقبل الرصيد الحالي.
 - للإيداع، يدخل العميل المبلغ المراد إيداعه، ويتم إرسال طلب "deposit" متبوعًا بالمبلغ، ويستقبل تأكيدًا بنجاح العملية.
 - للسحب، يدخل العميل المبلغ المراد سحبه، ويتم إرسال طلب "withdraw" متبوعًا بالمبلغ، ويتم التحقق من توفر الرصيد الكافي، ثم يستقبل تأكيدًا بنجاح العملية أو إشعارًا بنقص الرصيد.
 - لإنهاء الاتصال، يرسل العميل طلب "exit"، ويقوم بإغلاق الاتصال.
- يستمر البرنامج في تكرار هذه العملية حتى يختار العميل إنهاء الاتصال، وعندها يتم إغلاق مقبس الشبكة وإنهاء البرنامج.



Question 2: Simple Website Project with Python Flask Framework (you have choice to use Django or any Other Deferent Useful Python Project “from provide Project Links”)

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

Requirements :

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

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Comprehensive 5G Networking Design and Operation Learning Platform</title>
7      <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
8      <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
9  </head>
10 <body>
11     <header>
12         <nav class="navbar navbar-expand-lg navbar-light bg-light">
13             <a class="navbar-brand" href="{{ url_for('index') }}">5G Networking Design and Operation</a>
14             <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
15                 aria-label="Toggle navigation">
16                 <span class="navbar-toggler-icon"></span>
17             <div class="collapse navbar-collapse" id="navbarNav">
18                 <ul class="navbar-nav">
19                     <li class="nav-item">
20                         <a class="nav-link" href="{{ url_for('index') }}">Home</a>
21                     </li>
22                     <li class="nav-item">
23                         <a class="nav-link" href="{{ url_for('about') }}">About</a>
24                     </li>
25                     <li class="nav-item">
26                         <a class="nav-link" href="{{ url_for('contact') }}">Contact</a>
27                     </li>
28                 </ul>
29             </div>
30         </nav>
31     </header>
32
33     <div class="container mt-5">
34         <section>
35             <h2>Comprehensive Introduction to 5G Technologies and Design Principles</h2>
36             <ul>
37                 <li>Explore comprehensive introduction to 5G technologies and design principles.</li>
38             </ul>
39         </section>
40
41         <section>
42             <h2>Lessons on Structures, Frequencies, and Technologies Used in 5G Networks</h2>
43             <p>Explore lessons on structures, frequencies, and technologies used in 5G networks.</p>
44         </section>
45
46         <section>
47             <h2>Simulations for Designing and Operating Fifth Generation Wireless Networks</h2>
48             <p>Engage in simulations to experience designing and operating fifth generation wireless networks.</p>
49         </section>
50
51         <section>
52             <h2>Forum for Sharing Ideas, Experiments, and Methods Used in Designing 5G Networks</h2>
53             <p>Join the forum to share ideas, experiments, and methods used in designing 5G networks.</p>
54         </section>
55     </div>
56 </body>
57 </html>
58
59

```




```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>About - 5G Networking Design and Operation Learning Platform</title>
7 <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
8 <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
9 </head>
10 <body>
11 <header>
12 <nav class="navbar navbar-expand-lg navbar-light bg-light">
13 <div class="container">
14 <div class="float-left">
15 <a class="navbar-brand" href="{{ url_for('index') }}">5G Networking Design and Operation</a>
16 </div>
17 <div class="float-right">
18 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
19 <aria-label="Toggle navigation">
20 <span class="navbar-toggler-icon"></span>
21 </button>
22 <div class="collapse navbar-collapse" id="navbarNav">
23 <ul class="navbar-nav">
24 <li class="nav-item">
25 <a class="nav-link" href="{{ url_for('index') }}">Home</a>
26 </li>
27 <li class="nav-item">
28 <a class="nav-link" href="{{ url_for('about') }}">About</a>
29 </li>
30 <li class="nav-item">
31 <a class="nav-link" href="{{ url_for('contact') }}">Contact</a>
32 </li>
33 </ul>
34 </div>
35 </div>
36 </nav>
37 </header>
38 <div class="container mt-5">
39 <section>
40 <h2>About Us</h2>
41 <p>Welcome to the 5G Networking Design and Operation Learning Platform, your ultimate resource for mastering 5G network design and operation. We
42 <are committed to providing high-quality education in 5G networking technologies, enabling individuals and organizations to excel in this
43 <cutting-edge domain.</p>
44 <p>Our team consists of seasoned professionals with extensive experience in 5G networking and operation. We have curated a comprehensive
45 <curriculum and practical exercises to facilitate your learning journey, whether you're a novice or an experienced practitioner.</p>
46 </section>
47 <section>
48 <h2>Our Mission</h2>
49 <p>Our mission at 5G Networking Design and Operation Learning Platform is to empower learners with the knowledge and skills needed to thrive in
50 <the world of 5G networking design and operation. We are committed to fostering a supportive learning environment and equipping our community with
51 <the tools to succeed in this rapidly evolving field.</p>
52 </section>
53 <section>
54 <h2>Why Choose Us?</h2>
55 <ul>
56 <li>Expertly curated curriculum covering a wide range of 5G networking design and operation topics</li>
57 <li>Hands-on learning experiences with simulations and practical exercises</li>
58 <li>Engaging instructional materials developed by industry professionals</li>
59 <li>Interactive community forums for collaboration and knowledge sharing</li>
60 <li>Flexible learning options to accommodate diverse learning styles</li>
61 <li>Opportunities for professional development and career advancement</li>
62 </ul>
63 </section>
64 </div>
65 </body>
66 </html>
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Contact - 5G Networking Design and Operation Learning Platform</title>
7 <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
8 <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
9 </head>
10 <body>
11 <header>
12 <nav class="navbar navbar-expand-lg navbar-light bg-light">
13 <div class="container">
14 <div class="float-left">
15 <a class="navbar-brand" href="{{ url_for('index') }}">5G Networking Design and Operation</a>
16 </div>
17 <div class="float-right">
18 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
19 <aria-label="Toggle navigation">
20 <span class="navbar-toggler-icon"></span>
21 </button>
22 <div class="collapse navbar-collapse" id="navbarNav">
23 <ul class="navbar-nav">
24 <li class="nav-item">
25 <a class="nav-link" href="{{ url_for('index') }}">Home</a>
26 </li>
27 <li class="nav-item">
28 <a class="nav-link" href="{{ url_for('about') }}">About</a>
29 </li>
30 <li class="nav-item">
31 <a class="nav-link" href="{{ url_for('contact') }}">Contact</a>
32 </li>
33 </ul>
34 </div>
35 </div>
36 </nav>
37 </header>
```




```

32 .....<div class="container mt-5">
33 .....<section>
34 .....<h2>Contact Us</h2>
35 .....<p>Have a question or need assistance? Feel free to reach out to our team. We are here to help!</p>
36 .....<form action="mailto:info@example.com" method="post" enctype="text/plain">
37 .....<div class="form-group">
38 .....<label for="name">Name:</label>
39 .....<input type="text" id="name" name="name" class="form-control" required>
40 .....</div>
41 .....<div class="form-group">
42 .....<label for="email">Email:</label>
43 .....<input type="email" id="email" name="email" class="form-control" required>
44 .....</div>
45 .....<div class="form-group">
46 .....<label for="message">Message:</label>
47 .....<textarea id="message" name="message" class="form-control" rows="5" required></textarea>
48 .....</div>
49 .....<button type="submit" class="btn btn-primary">Send</button>
50 .....</form>
51 .....</section>
52 .....<section class="mt-5">
53 .....<h2>Our Location</h2>
54 .....<p>Our headquarters are located at:</p>
55 .....<div class="address">
56 .....123 Network Street<br>
57 .....City, State, ZIP<br>
58 .....Country
59 .....</div>
60 .....</section>
61 .....</div>
62 .....</body>
63 .....</html>
64
65

```

```

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('/contact')
def contact():
    return render_template('contact.html')

if __name__ == '__main__':
    app.run(debug=True, port=25546)

```

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()`.

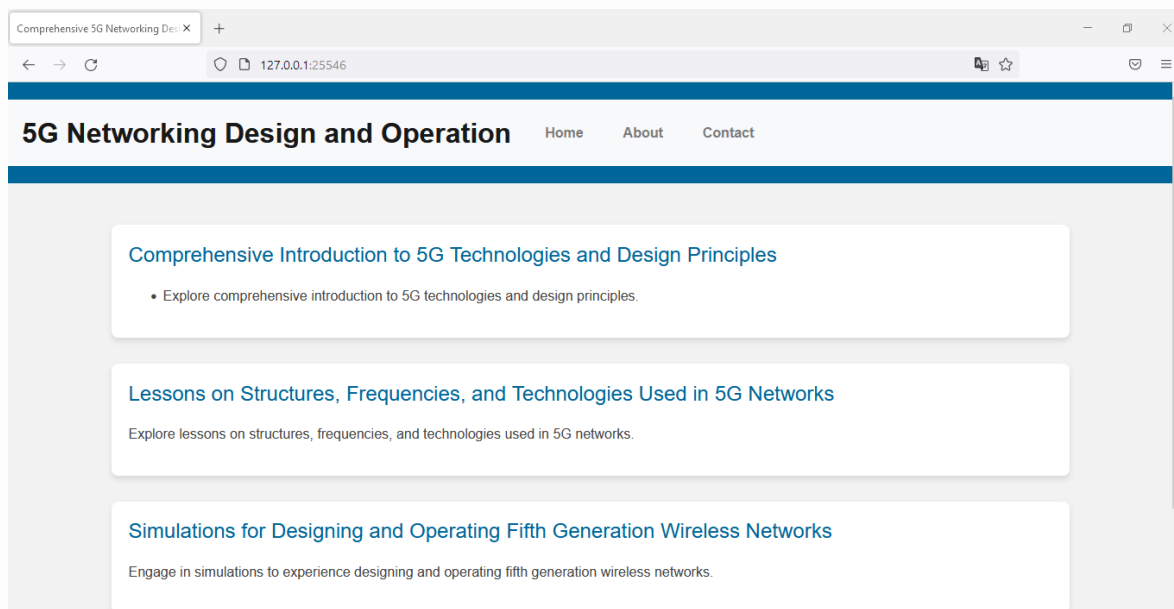


المسار template/about/ يعود إلى صفحة "about" ويتم تعيينه لدالة (about).

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 25546



5th, Network Programming : Homework No2

السنة الخامسة: وظيفة 2 برمجة شبكات

The image displays two screenshots of a web browser showing a website for "5G Networking Design and Operation".

Top Screenshot (About Us page):

- Browser Tab:** "About - 5G Networking Design and Operation"
- Address Bar:** "127.0.0.1:25546/about"
- Navigation Bar:** "Home", "About", "Contact"
- Section Header:** "About Us"
- Text:**

Welcome to the 5G Networking Design and Operation Learning Platform, your ultimate resource for mastering 5G network design and operation. We are committed to providing high-quality education in 5G networking technologies, enabling individuals and organizations to excel in this cutting-edge domain.

Our team consists of seasoned professionals with extensive experience in 5G networking and operation. We have curated a comprehensive curriculum and practical exercises to facilitate your learning journey, whether you're a novice or an experienced practitioner.

Bottom Screenshot (Contact Us page):

- Browser Tab:** "Contact - 5G Networking Design and Operation"
- Address Bar:** "127.0.0.1:25546/contact"
- Navigation Bar:** "Home", "About", "Contact"
- Section Header:** "Contact Us"
- Text:**

Have a question or need assistance? Feel free to reach out to our team. We are here to help!
- Form Fields:**
 - Name:** [Text input field]
 - Email:** [Text input field]
 - Message:** [Large text area]
- Submit Button:** "Send" (partially visible)