Name: Zainab Qirata , Number: 2851, Submitted To GitHub:@zainabqirata
Name: Alaa Cheikh dib, Number: 2938, Submitted To GitHub:@cheikhdibalaa
Name: Alaa Kassem, Number: 2793, Submitted To GitHub:@Alaa20kassem

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

Server.py > ...

```python
import socket
import threading
# Sample account details stored on the server
accounts = {
    'user1': {'balance': 1000, 'password': '111'},
    'user2': {'balance': 2000, 'password': '222'}
}
# Function to handle client requests
def handle_client(client_socket):
    # Authentication
    client_socket.send("Enter username: ".encode())
    username = client_socket.recv(1024).decode().strip()
    client_socket.send("Enter password: ".encode())
    password = client_socket.recv(1024).decode().strip()
    if username in accounts and accounts[username]['password'] == password:
        client_socket.send("Authentication successful.".encode())
    else:
        client_socket.send("Invalid credentials. Closing connection.".encode())
        client_socket.close()
        return
    while True:
        # Receive client requests
        request = client_socket.recv(1024).decode().strip()
        if request == "check_balance":
            balance = accounts[username]['balance']
            client_socket.send(f"Your balance is {balance}".encode())
        elif request.startswith("deposit"):
            amount = float(request.split()[1])
            accounts[username]['balance'] += amount
            client_socket.send("Deposit successful.".encode())
```

```
        elif request.startswith("withdraw"):
            amount = float(request.split()[1])
            if accounts[username]['balance'] >= amount:
                accounts[username]['balance'] -= amount
                client_socket.send("Withdrawal successful.".encode())
            else:
                client_socket.send("Insufficient funds.".encode())
        elif request == "exit":
            client_socket.send("Exiting.".encode())
            break
        else:
            client_socket.send("Invalid request.".encode())
    # Close client connection
    client_socket.close()
# Main server function
def main():
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(('127.0.0.1', 5555))
    server.listen(5)
    print("[*] Listening on 127.0.0.1:5555")
    while True:
        client_socket, addr = server.accept()
        print(f"[*] Accepted connection from {addr[0]}:{addr[1]}")
        # Create a new thread to handle client requests
        client_handler = threading.Thread(target=handle_client, args=(client_socket,))
        client_handler.start()
if __name__ == "__main__":
    main()

```

client.py > main

```python
import socket
def main():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect(('127.0.0.1', 5555))
    # Authentication
    response = client.recv(1024).decode()
    print(response)
    username = input()
    client.send(username.encode())
    response = client.recv(1024).decode()
    print(response)
    password = input()
    client.send(password.encode())
    response = client.recv(1024).decode()
    print(response)
    while True:
        print("Options:")
        print("1. Check Balance")
        print("2. Deposit")
        print("3. Withdraw")
        print("4. Exit")
        choice = input("Enter your choice: ")
        if choice == '1':
            client.send("check_balance".encode())
            balance = client.recv(1024).decode()
            print(balance)
        elif choice == '2':
            amount = float(input("Enter amount to deposit: "))
            client.send(f"deposit {amount}".encode())
            response = client.recv(1024).decode()
```

```
            print(response)
        elif choice == '3':
            amount = float(input("Enter amount to withdraw: "))
            client.send(f"withdraw {amount}".encode())
            response = client.recv(1024).decode()
            print(response)
        elif choice == '4':
            client.send("exit".encode())
            print("Exiting.")
            break
        else:
            print("Invalid choice.")
    client.close()

if __name__ == "__main__":
    main()
```

```
Enter username:
user1
Enter password:
111
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: 500
Deposit successful.
Options:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 4
Exiting.
```

```
Enter username:
user2
Enter password:
222
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: 3
Enter amount to withdraw: 500
Withdrawal successful.
Options:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 1
Your balance is 1500.0
```

```
[*] Listening on 127.0.0.1:5555
[*] Accepted connection from 127.0.0.1:52720
[*] Accepted connection from 127.0.0.1:52725
```

السؤال الأول:

كود السيرفر:

أولاً، يتم استيراد مكتبة socket لإنشاء اتصالات الشبكة ومكتبة threading لتنفيذ عمليات متعددة في وقت واحد. يتم تعريف حسابات المستخدمين في قاموس accounts، حيث يحتوي كل حساب على رصيد وكلمة مرور.

الدالة handle_client تعالج طلبات العملاء. تبدأ بالتحقق من هوية المستخدم عبر طلب اسم المستخدم وكلمة المرور. إذا كانت بيانات الدخول صحيحة، يمكن للمستخدم طلب عمليات مثل فحص الرصيد، الإيداع، السحب، والخروج. إذا كانت بيانات الدخول غير صحيحة، يتم قطع الاتصال.

الدالة main تنشئ الخادم وتستمع للاتصالات الواردة. عند استقبال اتصال، يتم قبول الاتصال وإنشاء اتصال جديد لمعالجة طلبات العميل باستخدام handle_client.

إذا تم تشغيل البرنامج كملف رئيسي، يتم استدعاء الدالة main لتشغيل الخادم.

كود العميل:

أولاً، يتم استيراد مكتبة socket لإنشاء اتصالات الشبكة. ثم يتم تعريف الدالة main التي تقوم بإنشاء مقبس عميل socket للاتصال بالخادم عند العنوان 127.0.0.1والمنفذ 5555.

بعد الاتصال، يبدأ العميل بعملية المصادقة حيث يتلقى العميل طلبات لإدخال اسم المستخدم وكلمة المرور ويرسلها إلى الخادم. يستقبل العميل الردود من الخادم ويعرضها.

بعد المصادقة الناجحة، يدخل العميل في حلقة تكرارية تقدم له خيارات لفحص الرصيد، الإيداع، السحب، أو الخروج. بناءً على اختيار المستخدم، يرسل العميل الطلب المناسب إلى الخادم وينتظر الرد ليعرضه.

تتضمن الخيارات المتاحة:

١. فحص الرصيد: يرسل العميل طلبًا للتحقق من الرصيد ويتلقى الرصيد الحالي.

٢. الإيداع: يطلب العميل من المستخدم إدخال مبلغ للإيداع ويرسل الطلب إلى الخادم، ثم يعرض الرد.

٣. السحب: يطلب العميل من المستخدم إدخال مبلغ للسحب ويرسل الطلب إلى الخادم، ثم يعرض الرد.

٤. الخروج: يرسل العميل طلبًا للخروج وينهي الاتصال.

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 .

templates > index.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Comprehensive Wireless Communications Learning Platform</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="{{ url_for('index') }}">Wireless Communications</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="nav
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('index') }}">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('about') }}">About</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('contact') }}">Contact</a>
                    </li>
                </ul>
            </div>
        </nav>
    </header>

    <div class="container mt-5">
        <section>
            <h2>Lessons and Simulations</h2>
            <ul>
                <li>Lessons on wireless communication theories such as broadband and narrowband, and antenna concepts.</li>
                <li>Simulations of wireless systems to understand performance and interactions between devices.</li>
            </ul>
        </section>

        <section>
            <h2>Illustrative Videos</h2>
            <p>Explore communication processes and their real-life applications through illustrative videos.</p>
        </section>

        <section>
            <h2>Student and Engineer Community</h2>
            <p>Join the community of students and engineers to discuss modern developments and exchange experiences in the field of wireless comm
        </section>
    </div>

</body>
</html>
```

templates > about.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>About - Wireless Communications</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="{{ url_for('index') }}">Wireless Communications</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="nav
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('index') }}">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('about') }}">About</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('contact') }}">Contact</a>
                    </li>
                </ul>
            </div>
        </nav>
    </header>

    <div class="container mt-5">
        <section>
            <h2>About Us</h2>
            <p>Welcome to Wireless Communications Learning Platform, your ultimate resource for mastering the field of wire
            <p>Our team comprises seasoned professionals with extensive experience in wireless communications. We have cura
        </section>

        <section>
            <h2>Our Mission</h2>
            <p>Our mission at Wireless Communications Learning Platform is to empower learners with the knowledge and skill
        </section>

        <section>
            <h2>Why Choose Us?</h2>
            <ul>
                <li>Expertly curated curriculum covering a wide range of wireless communication topics</li>
                <li>Hands-on learning experiences with simulations and practical exercises</li>
                <li>Engaging instructional materials developed by industry professionals</li>
                <li>Interactive community forums for collaboration and knowledge sharing</li>
                <li>Flexible learning options to accommodate diverse learning styles</li>
                <li>Opportunities for professional development and career advancement</li>
            </ul>
        </section>
    </div>

</body>
</html>
```

templates > contact.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact - Wireless Communications</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="{{ url_for('index') }}">Wireless Communications</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="nav
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('index') }}">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('about') }}">About</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('contact') }}">Contact</a>
                    </li>
                </ul>
            </div>
        </nav>
    </header>
    <div class="container mt-5">
        <section>
            <h2>Contact Us</h2>
            <p>Have a question or need assistance? Feel free to reach out to our team. We are here to help!</p>

            <form>
                <div class="form-group">
                    <label for="name">Name:</label>
                    <input type="text" id="name" name="name" class="form-control" required>
                </div>
                <div class="form-group">
                    <label for="email">Email:</label>
                    <input type="email" id="email" name="email" class="form-control" required>
                </div>
                <div class="form-group">
                    <label for="message">Message:</label>
                    <textarea id="message" name="message" class="form-control" rows="5" required></textarea>
                </div>
                <button type="submit" class="btn btn-primary">Send</button>
            </form>
        </section>

        <section class="mt-5">
            <h2>Our Location</h2>
            <p>Our headquarters are located at:</p>
            <address>
                123 Wireless Street<br>
                City, State, ZIP<br>
                Country
            </address>
        </section>
    </div>
</body>
</html>
```

```python
app.py > ...
1   from flask import Flask, render_template
2
3   app = Flask(__name__)
4
5   @app.route('/')
6   def index():
7       return render_template('index.html')
8
9   @app.route('/about')
10  def about():
11      return render_template('about.html')
12
13  @app.route('/contact')
14  def contact():
15      return render_template('contact.html')
16
17  if __name__ == '__main__':
18      app.run(debug=True , port=25544)
19
```

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

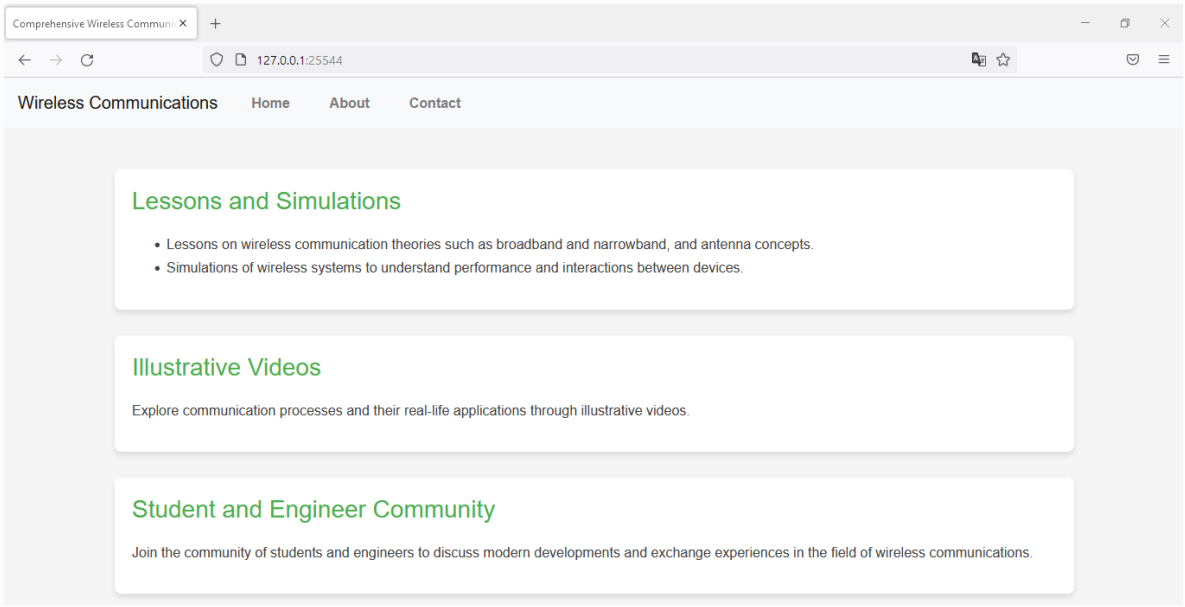
المسار /template يعود إلى الصفحة الرئيسية ويتم تعيينه لدالة home().

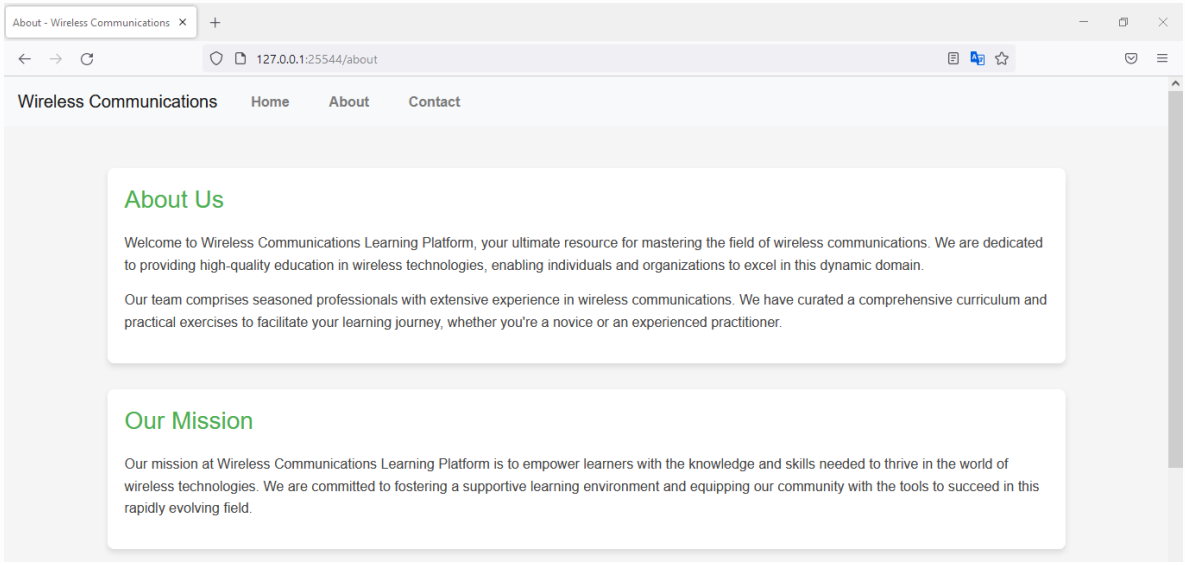
المسار /template/about يعود إلى صفحة "about" ويتم تعيينه لدالة about().

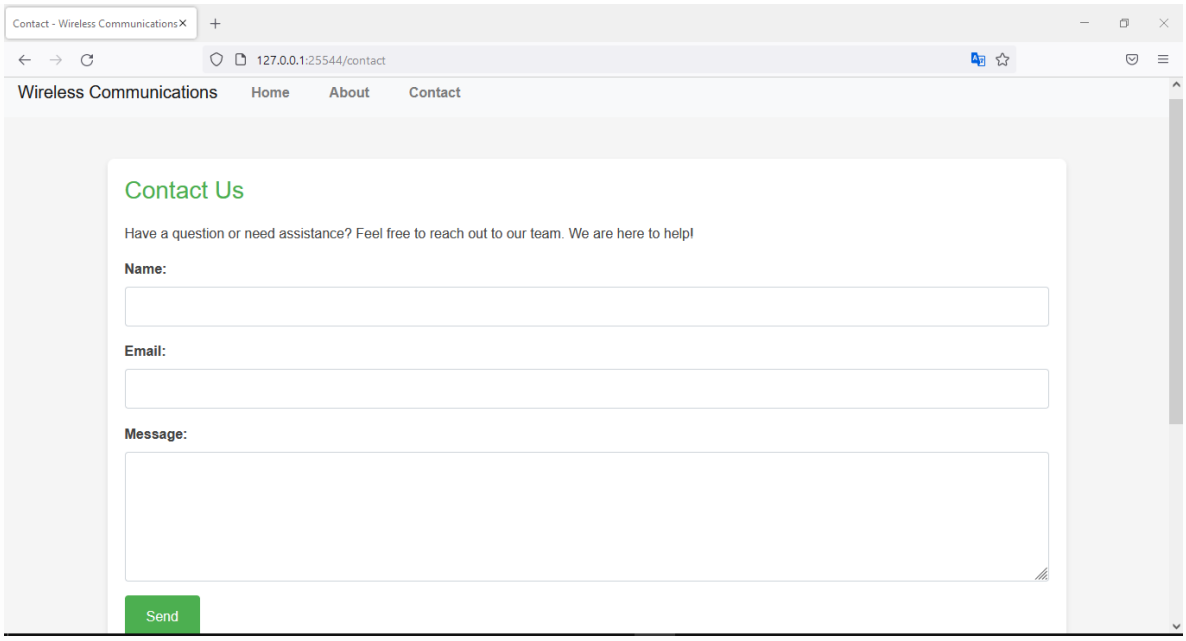
المسار /template/contact يعود إلى صفحة "contact " ويتم تعيينه لدالة contact().

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 25544


Comprehensive Wireless Communi
127.0.0.1:25544
Wireless Communications
Home
About
Contact
Lessons and Simulations
Lessons on wireless communication theories such as broadband and narrowband, and antenna concepts.
Simulations of wireless systems to understand performance and interactions between devices.
Illustrative Videos
Explore communication processes and their real-life applications through illustrative videos.
Student and Engineer Community
Join the community of students and engineers to discuss modern developments and exchange experiences in the field of wireless communications.


About - Wireless Communications
127.0.0.1:25544/about
Wireless Communications
Home
About
Contact
About Us
Welcome to Wireless Communications Learning Platform, your ultimate resource for mastering the field of wireless communications. We are dedicated to providing high-quality education in wireless technologies, enabling individuals and organizations to excel in this dynamic domain.
Our team comprises seasoned professionals with extensive experience in wireless communications. We have curated a comprehensive curriculum and practical exercises to facilitate your learning journey, whether you're a novice or an experienced practitioner.
Our Mission
Our mission at Wireless Communications Learning Platform is to empower learners with the knowledge and skills needed to thrive in the world of wireless technologies. We are committed to fostering a supportive learning environment and equipping our community with the tools to succeed in this rapidly evolving field.


Contact - Wireless Communications
127.0.0.1:25544/contact
Wireless Communications
Home
About
Contact
Contact Us
Have a question or need assistance? Feel free to reach out to our team. We are here to help!
Name:
Email:
Message:
Send