



Name: نغم حكمت جراد, Number:2260, Submitted To GitHub: _____

Name: يارا خيرات العلوني, Number:2263, 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:

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

```
import socket
import threading
accounts = {
    '1234567890': {'pin': '1234', 'balance': 1000},
    '9876543210': {'pin': '4321', 'balance': 500}
}
def handle_client(client_socket, address):
    print(f"[+] Accepted connection from {address[0]}:{address[1]}")
    account_number = client_socket.recv(1024).decode('utf-8')
    pin = client_socket.recv(1024).decode('utf-8')

    if account_number in accounts and accounts[account_number]['pin'] == pin:
        client_socket.send("Authentication successful!".encode('utf-8'))
    else:
        client_socket.send("Authentication failed!".encode('utf-8'))
        client_socket.close()
    return
```

```

while True:
    request = client_socket.recv(1024).decode('utf-8')
    print(f"[*] Received: {request} from {address[0]}:{address[1]}")

    if request == "1":
        balance = accounts[account_number]['balance']
        client_socket.send(f"Your balance: {balance}".encode('utf-8'))
    elif request == "2":
        amount = float(client_socket.recv(1024).decode('utf-8'))
        accounts[account_number]['balance'] += amount
        client_socket.send("Deposit successful!".encode('utf-8'))
    elif request == "3":
        amount = float(client_socket.recv(1024).decode('utf-8'))
        if amount <= accounts[account_number]['balance']:
            accounts[account_number]['balance'] -= amount
            client_socket.send("Withdrawal successful!".encode('utf-8'))
        else:
            client_socket.send("Insufficient funds!".encode('utf-8'))
    elif request == "4":
        client_socket.send("Session ended!".encode('utf-8'))
        break
    else:
        client_socket.send("Invalid request!".encode('utf-8'))
balance = accounts[account_number]['balance']
client_socket.send(f"Final balance: {balance}".encode('utf-8'))
client_socket.close()

HOST = '127.0.0.1'
PORT = 65432
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((HOST, PORT))
server.listen(5)
print(f"[*] Listening on {HOST}:{PORT}")
while True:
    client, address = server.accept()
    client_handler = threading.Thread(target=handle_client, args=(client, address))
    client_handler.start()

import socket
HOST = '127.0.0.1'
PORT = 65432
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((HOST, PORT))
account_number = input("Enter account number: ")
pin = input("Enter PIN: ")
client.send(account_number.encode('utf-8'))
client.send(pin.encode('utf-8'))
response = client.recv(1024).decode('utf-8')
print(f"[*] {response}")
if "successful" in response:
    while True:
        print("\nSelect an operation:")
        print("1. Check balance")
        print("2. Deposit")
        print("3. Withdraw")
        print("4. Exit")
        choice = input("Enter your choice: ")
        client.send(choice.encode('utf-8'))
        if choice == "1":
            balance = client.recv(1024).decode('utf-8')
            print(f"[*] {balance}")
        elif choice == "2":
            amount = input("Enter amount to deposit: ")
            client.send(amount.encode('utf-8'))
            response = client.recv(1024).decode('utf-8')
            print(f"[*] {response}")
        elif choice == "3":
            amount = input("Enter amount to withdraw: ")

```

```

        client.send(amount.encode('utf-8'))
        response = client.recv(1024).decode('utf-8')
        print(f"[*] {response}")
    elif choice == "4":
        response = client.recv(1024).decode('utf-8')
        print(f"[*] {response}")
        break
    else:
        print("[*] Invalid choice!")
    final_balance = client.recv(1024).decode('utf-8')
    print(f"[*] {final_balance}")

client.close()

```

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 .
from pytube import YouTube

```

def download_video():
    video_url = input("أدخل رابط فيديو يوتيوب: ")

    try:
        yt = YouTube(video_url)

        # اختيار دقة الفيديو (اختيار أول دقة متاحة)
        stream = yt.streams.filter(progressive=True).first()

        # يمكنك اختيار دقة معينة بتحديد `resolution` (مثل: `yt.streams.filter(progressive=True, resolution='720p').first()`)

        # تنزيل الفيديو
        print(f'يتم تنزيل: {yt.title}')
        stream.download()
        print("تم تنزيل الفيديو بنجاح")

    except Exception as e:
        print(f'حدث خطأ: {e}')

if __name__ == "__main__":
    download_video()

```

