

**Second Network Programming Homework**

**إعداد الطالبتين:**

**سدرة زين تنزكلي (2816) - رشا شادي جبيلي (2939)**

**إشراف الدكتور المهندس: مهند عيسى**

**الجمهورية العربية السورية**

**وزارة التعليم العالي والبحث العلمي**

**جامعة تشرين**

**كلية الهندسة الميكانيكية والكهربائية**

**قسم هندسة الاتصالات والالكترونيات**

**السنة الخامسة**

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

import **socket**, **threading**

accounts = {

    "Sedra": 1000,

    "Rasha": 1500

}

def **handle\_client**(client\_socket, client\_address):

**print**("Accepted connection from {}".**format**(client\_address))

    client\_socket.sendall(b"Welcome to the Bank ATM. Please enter your account name: ")

    account\_name = client\_socket.recv(1024).decode().strip()

    if account\_name **not** in accounts:

        client\_socket.sendall(b"Account not found. Connection closed.")

        client\_socket.close()

        return

    client\_socket.sendall("Hello, {}. Please enter your PIN: ".**format**(account\_name).**encode**())

    pin = client\_socket.recv(1024).decode().strip()

    while True:

        client\_socket.sendall(b"Available operations:\n1. Check Balance\n2. Deposit\n3. Withdraw\n4. Exit\nEnter your choice: ")

        choice = client\_socket.recv(1024).decode().strip()

        if choice == "1":

            client\_socket.sendall("Your current balance is: {}".**format**(accounts[account\_name]).**encode**())

        elif choice == "2":

            amount = **float**(client\_socket.recv(1024).decode().strip())

            accounts[account\_name] += amount

            client\_socket.sendall("Deposit successful. Your updated balance is: {}".**format**(accounts[account\_name]).**encode**())

        elif choice == "3":

            amount = **float**(client\_socket.recv(1024).decode().strip())

            if amount > accounts[account\_name]:

                client\_socket.sendall(b"Insufficient funds.")

            else:

                accounts[account\_name] -= amount

                client\_socket.sendall("Withdrawal successful. Your updated balance is: {}".**format**(accounts[account\_name]).**encode**())

        elif choice == "4":

            break

**print**("Closing connection with {}".**format**(client\_address))

    client\_socket.close()

server = **socket**.**socket**(**socket**.AF\_INET, **socket**.SOCK\_STREAM)

server.**bind**(("0.0.0.0", 4444))

server.**listen**(5)

**print**("Server listening on port 0.0.0.0")

while True:

    client\_socket, client\_address = server.**accept**()

    client\_thread = **threading**.**Thread**(target=**handle\_client**, args=(client\_socket, client\_address))

    client\_thread.**start**()

تم استيراد المكتبتين socket وthreading لإنشاء المقابس وجعل السيرفر يخدم العملاء بنفس الوقت. باستخدام المتغير accounts تم بناء الحسابات بحيث يكون اسم صاحب الحساب هو المفتاح Key للقاموس والمبلغ المالي في الحساب هو القيمة Value المقابلة للمفتاح.

جعلنا السيرفر يعمل على أي عنوان IP في التطبيق باستخدام server.bind(("0.0.0.0", 5555)) ومع ملاحظة أننا جعلنا رقم المنفذ للسيرفر هو: 4444.

باستخدام التابع handle\_client(client\_socket, client\_address) تم التعامل مع اتصالات العملاء جميعها كما يلي: تم تمرير البارمترين client\_socket و client\_addressاللذان يمثلان مقبس العميل وعنوان العميل وذلك عند قبول الاتصال معه، وعند قبول الاتصال نقوم بإرسال رسالة ترحيبية للمستخدم نطلب فيها إدخال اسمه (اسم صاحب الحساب) ونخزن الاسم في المتحول account\_name ثم نتحقق من كون اسم صاحب الحساب موجود فعلاً في الحسابات في البنك أم لا ونطبع في حال لم يكن موجودا الرسالة: Account not found. Connection closed..

إن كان اسم المستخدم موجود فعلاً أي لديه حساب نطلب منه إدخال رمز أمان له PIN، ومن ثم نعرض عليه جميع الخيارات الممكنة: Available operations:\n1. Check Balance\n2. Deposit\n3. Withdraw\n4. Exit

نتابع حسب رقم الخيار الذي يختاره العميل: بالعملية المرادة.

import **socket**

client = **socket**.**socket**(**socket**.AF\_INET, **socket**.SOCK\_STREAM)

client.**connect**(("127.0.0.1", 4444))

response = client.**recv**(1024).**decode**()

**print**(response)

account\_name = **input**()

client.**sendall**(account\_name.**encode**())

response = client.**recv**(1024).**decode**()

**print**(response)

pin = **input**()

client.**sendall**(pin.**encode**())

while True:

    response = client.**recv**(1024).**decode**()

**print**(response)

    choice = **input**()

    client.**sendall**(choice.**encode**())

    if choice == "4":

        break

    if choice in ["2", "3"]:

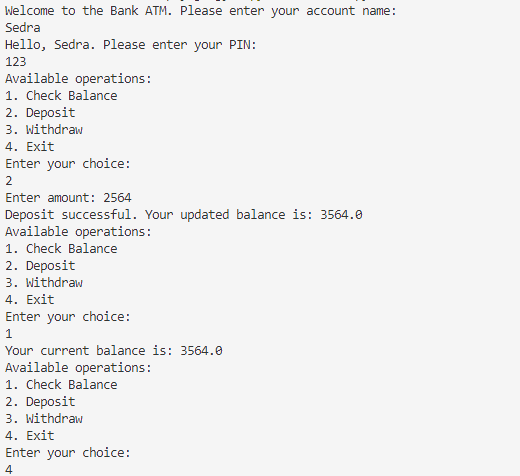
        amount = **input**("Enter amount: ")

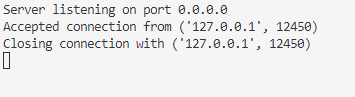
        client.**sendall**(amount.**encode**())

    operation\_result = client.**recv**(1024).**decode**()

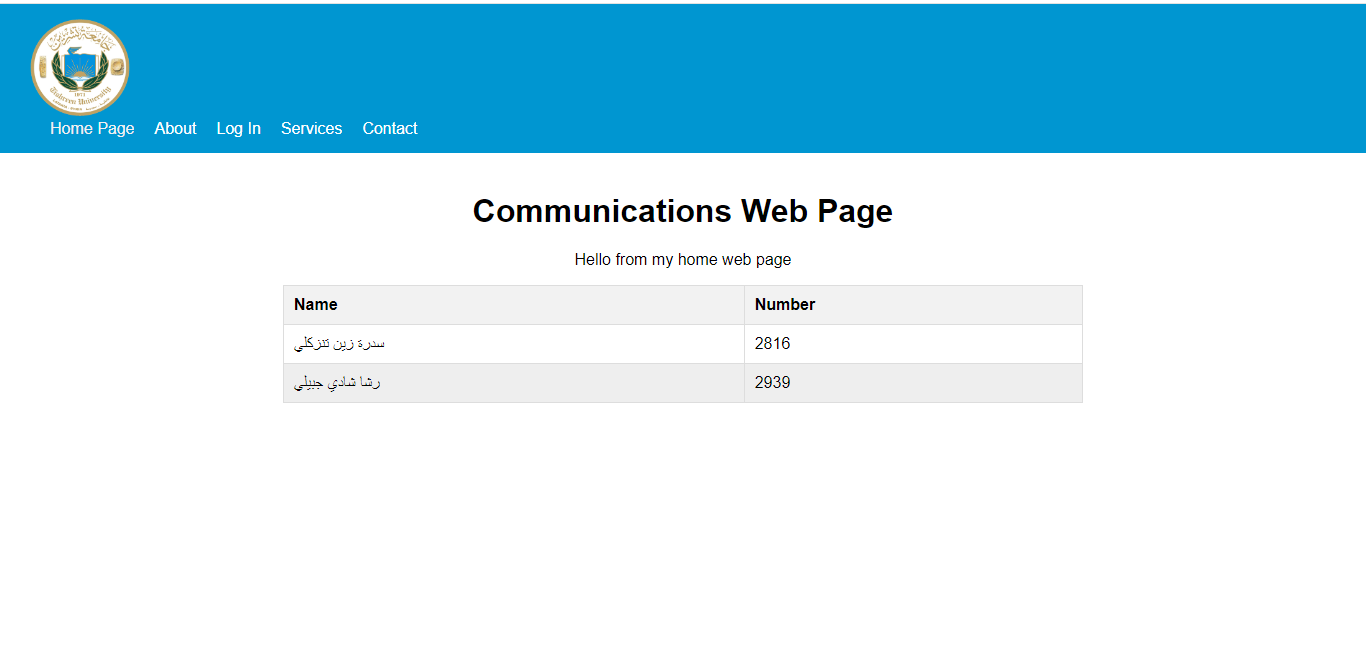
**print**(operation\_result)

client.**close**()





**Question 2:** Website:



برنامج سيرفر Flask:

from **flask** import **Flask**, **render\_template**

app = **Flask**(\_\_name\_\_)

**@app.route**('/')

def **home**():

    return **render\_template**('index.html')

**@app.route**('/info')

def **about**():

    return **render\_template**('about.html')

**@app.route**('/login')

def **login**():

    return **render\_template**('login.html')

if \_\_name\_\_ == '\_\_main\_\_':

    app.**run**(debug=True)

