



مياس رفيق ملاوخي 2531 محسن محمد علي 2585

السؤال الأول

SERVER

تعريف الحسابات:

يتم تعريف حسابات البنك في قاموس حيث يحتوي كل حساب على رقم الحساب، رمز PIN والرصيد.

تابع: handle_client

1. يستقبل بيانات الحساب ورمز PIN من العميل في صيغة: account_number:PIN.
2. يتحقق من صحة بيانات الحساب.
3. إذا كانت البيانات صحيحة، يسمح للعميل بإجراء عمليات مثل التحقق من الرصيد، الإيداع، السحب، أو إنهاء الجلسة بصيغة operation:amount تكون العملية check أو deposit أو withdraw أو exit.
4. يرسل النتائج للعميل بعد تنفيذ العمليات.
5. يغلق الاتصال في النهاية.

تابع: start_server

1. ينشئ سوكيت الخادم ويقوم بربطه على جميع العناوين المتوفرة (0.0.0.0) وعلى المنفذ 9999.
2. يستمع للاتصالات الواردة بحد أقصى 5 اتصالات في الطابور.
3. يقبل اتصالات العملاء ويخلق ثريد جديد لكل عميل باستخدام التابع handle_client.

```
import socket
import threading
```

```
# أرصدها مع البنك حسابات تعريف
accounts = {
    'mayas': ('1111', 1000.0),
    'mohsen': ('1111', 1500.0)
}
```

```
# متصل عميل كل مع للتعامل تابع
def handle_client(client_socket):
    try:
```

```
        # العميل من PIN ورمز الحساب بيانات استقبال
        client_socket.send(b'Enter account number and PIN (format:
account_number:PIN): ')
    except:
```

```

credentials = client_socket.recv(1024).decode().strip()
account_number, pin = credentials.split(':')

# الحساب بيانات صحة من التحقق
if account_number in accounts and accounts[account_number][0] == pin:
    client_socket.send(b'Authenticated\n')
    while True:
        # للعميل المتاحة الخيارات إرسال
        client_socket.send(b'Choose operation (format:
operation:amount): check, deposit, withdraw, exit\n')
        operation =
client_socket.recv(1024).decode().strip().split(':')

        if operation[0] == 'check':
            # الرصيد من تحقق
            balance = accounts[account_number][1]
            client_socket.send(f'Your balance is
{balance}\n'.encode())

        elif operation[0] == 'deposit':
            # المال إيداع
            amount = float(operation[1])
            accounts[account_number] = (pin,
accounts[account_number][1] + amount)
            client_socket.send(b'Deposit successful\n')

        elif operation[0] == 'withdraw':
            # المال سحب
            amount = float(operation[1])
            if accounts[account_number][1] >= amount:
                accounts[account_number] = (pin,
accounts[account_number][1] - amount)
                client_socket.send(b'Withdrawal successful\n')
            else:
                client_socket.send(b'Insufficient balance\n')

        elif operation[0] == 'exit':
            # الجلسة إنهاء
            balance = accounts[account_number][1]
            client_socket.send(f'Your final balance is
{balance}\n'.encode())
            break
        else:
            client_socket.send(b'Invalid operation\n')

    else:
        client_socket.send(b'Authentication failed\n')
finally:
    client_socket.close()

# الخادم بدء تابع
def start_server():
    ssock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    ssock.bind(('0.0.0.0', 5555))
    ssock.listen(5)
    print("Server started and listening on port 5555")

```

```

while True:
    client_socket, addr = ssock.accept()
    print(f'Accepted connection from {addr}')
    th = threading.Thread(target=handle_client, args=(client_socket,))
    th.start()

if __name__ == "__main__":
    start_server()

```

Client

إنشاء سوكت:

- يتم إنشاء سوكت باستخدام socket.socket() لتكوين اتصال TCP

الاتصال بالخادم:

- يتم الاتصال بالخادم على عنوان IP المحلي 127.0.0.1 والمنفذ 9999

حلقة الاستقبال والإرسال:

- يستقبل العميل البيانات من الخادم ويعرضها على المستخدم.
- إذا استلم العميل رسالة تحتوي على "Your final balance is" أو "Authentication failed" ، يتم توقيف الحلقة وإغلاق الاتصال.
- يتم استقبال إدخال المستخدم وإرساله إلى الخادم.

```

Import socket

```

```

csock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
csock.connect(('127.0.0.1', 5555))

```

```

while True:
    data = csock.recv(1024).decode()
    print(data, end='')

    if 'Your final balance is' in data or 'Authentication failed' in data:
        break

    user_input = input()
    csock.send(user_input.encode())

csock.close()

```

الخرج:
تشغيل الخادم

```
Server started and listening on port 5555
```

اتصال 3 عملاء بالخادم

```
Server started and listening on port 5555
Accepted connection from ('127.0.0.1', 49881)
Accepted connection from ('127.0.0.1', 49892)
Accepted connection from ('127.0.0.1', 49899)
```

عمليات احد العملاء

```
Enter account number and PIN (format: account_number:PIN): mayas:1111
Authenticated

Choose operation (format: operation:amount): check, deposit, withdraw, exit
check
Your balance is 1000.0

Choose operation (format: operation:amount): check, deposit, withdraw, exit
deposit:1000
Deposit successful

Choose operation (format: operation:amount): check, deposit, withdraw, exit
withdraw:500
Withdrawal successful

Choose operation (format: operation:amount): check, deposit, withdraw, exit
4
Invalid operation

Choose operation (format: operation:amount): check, deposit, withdraw, exit
exit
Your final balance is 1500.0

Process finished with exit code 0
|
```

```
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=5000)
```

- من مكتبة flask يتم استدعاء Flask و render_template.
- يتم إنشاء كائن Flask باستخدام التعليمة `app = Flask(name)`، حيث يتم تعريف المتغير `app` الذي يمثل التطبيق الخاص بنا.
- يتم تعريف مسارات الصفحات باستخدام التابع `route` الذي يأخذ عنوان URL ويتعامل معه حيث تم تعريف ثلاث مسارات لصفحات الويب الأساسية الرئيسية (الصفحة الرئيسية، من نحن، اتصل بنا).
- يتم استخدام تابع `render_template` لتحميل صفحة HTML على المتصفح
- يتم تشغيل السيرفر على منفذ 5000 وفي وضع التصحيح `debug=True`.

صفحة index:

```
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
  <meta charset="utf-8">
  <style>
```

```

body {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}

h1 {
    font-size: 36px;
    color: #229944;
}

p {
    font-size: 18px;
    line-height: 2;
    color: #999;
}

</style>
<title>الرئيسية الصفحة</title>
</head>
<body >

<ul>
    <li><a href="{ { url_for('about') } }">الموقع عن</a> </li>
    <li><a href="{ { url_for('contact') } }">بنا اتصل</a></li>
</ul>
    <h1>ملاوخي رفيق مياس</h1>
    <p>2531 الجامعي الرقم</p>
<h1>علي محمد محسن</h1>
<p>2585 الجامعي الرقم</p>
</body>
</html>

```

تبدأ الصفحة بعلامة DOCTYPE التي تحدد نوعية المستند كـ HTML5.

تظهر علامة HTML بعدها تأتي العناصر الأكثر أهمية في الصفحة مثل العنوان، النص، والصورة.

تحتوي عناصر HEAD على العناصر التي توفر المعلومات عن الصفحة مثل عنوان الصفحة والتعريفات الأخرى.

العناصر الأساسية في هذه الصفحة هي عناصر BODY، والتي تحتوي على العناصر التي يمكن للمستخدم رؤيتها والتفاعل معها مثل النص والصورة الموجودة في هذه الصفحة.

تحتوي أيضاً الصفحة على عناصر STYLE التي تسمح بتعيين الأنماط والتنسيقات للعناصر المختلفة في الصفحة مثل تحديد الحجم واللون والخط للنص وغيرها من الخصائص.

وتحديداً يقوم الكود بعرض صورة باسم "image.png" الموجودة في مجلد "static" باستخدام التابع url_for وذلك في عنصر img. كما يتم عرض عنوان رئيسي بحجم 36 بكسل باللون الأخضر الداكن، وكذلك عرض فقرة بحجم 18 بكسل ولون رمادي فاتح.



- [عن الموقع](#)
- [اتصل بنا](#)

مياس رفيق ملاوخي

الرقم الجامعي 2531

محسن محمد علي

الرقم الجامعي 2585

صفحة about:

```
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
  <meta charset="utf-8">
  <title>عن الموقع</title>
  <style>
  body {
    display: flex;
```

```
        justify-content: center;
        align-items: center;
        flex-direction: column;
    }
    h1 {
        font-size: 36px;
        color: #229944;
    }

p {
    font-size: 18px;
    line-height: 2;
    color: #999;
}

</style>
</head>
<body>

<ul>
    <li><a href="{{ url_for('contact') }}">بنا اتصل</a> </li>
    <li><a href="{{ url_for('index') }}">الرئيسية</a></li>
</ul>

    <h1>الموقع عن</h1>
    <p>الشبكات برمجة الثانية للوظيفة كحل مصمم الموقع هذا</p>
    <p>الثاني السؤال</p>

</body>
</html>
```




- [اتصل بنا](#)
- [الرئيسية](#)

عن الموقع

هذا الموقع مصمم كحل للوظيفة الثانية برمجة الشبكات السؤال الثاني

صفحة :contact

```
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
  <style>
    body {
      display: flex;
      justify-content: center;
      align-items: center;
      flex-direction: column;
    }
    h1 {
      font-size: 36px;
      color: #229944;
    }
  }
  p {
    font-size: 18px;
```

```
    line-height: 2;
    color: #999;
}
</style>
<meta charset="utf-8">
<title>بنا اتصل</title>
</head>
<body>

<ul>
    <li><a href="{{ url_for('about') }}">الموقع عن</a> </li>
    <li><a href="{{ url_for('index') }}">الرئيسية</a></li>
</ul>
<h1>بنا اتصل</h1>
<p>الموقع حول المعلومات من لمزيد معنا التواصل يمكنك</p>
</body>
</html>
```