



وظيفة رقم 2

دلامه ناجح عثمان 2382

ديانا عدنان سلوم 2461

لين عبد الكريم شناني 2971

1-

السيرفر (الخادم)

شرح الكود الخاص بالخادم

1. إعداد سوكت الخادم:

- يتم إنشاء سوكت باستخدام `socket.socket()` وتعيين الخيارات للسماح بإعادة استخدام العنوان `(socket.SO_REUSEADDR)`.
- يتم ربط السوكت بعنوان `localhost` ومنفذ `8000`.
- يتم تعيين الخادم للاستماع إلى 5 اتصالات كحد أقصى.

2. قاموس الحسابات:

- يتم تعريف قاموس `accounts` لتخزين معلومات الحسابات، بما في ذلك الرصيد وكلمة المرور.

3. تابع `handle_client`

- يتم استقبال اسم المستخدم وكلمة المرور من العميل والتحقق منها.
- إذا كانت بيانات الدخول غير صحيحة، يتم قطع الاتصال.
- بعد التحقق الناجح، يتم تنفيذ العمليات البنكية (التحقق من الرصيد، الإيداع، السحب، أو الخروج) بناءً على اختيار العميل.
- يتم إرسال النتائج للعميل بعد تنفيذ العمليات.
- إذا قام العميل باختيار `exit`، يتم إرسال الرصيد النهائي وقطع الاتصال.

4. الاستماع للاتصالات:

- يدخل الخادم في حلقة لا نهائية لقبول اتصالات العملاء.
- لكل اتصال جديد، يتم إنشاء ثريد جديد للتعامل مع العميل باستخدام التابع `handle_client`.

```

import socket
import threading

# Server-side code
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind(('localhost', 8000))
server_socket.listen(5)

# Dictionary to store account details
accounts = {
    'دلامه': {'balance': 1000, 'password': '2382'},
    'ديانا': {'balance': 2000, 'password': '2461'},
    'لين': {'balance': 3000, 'password': '2971'}
}

def handle_client(conn, addr):
    print(f'New connection from {addr}')

    # Authenticate the client
    try:
        conn.send('Enter your username: '.encode())
        username = conn.recv(1024).decode().strip()
        conn.send('Enter your password: '.encode())
        password = conn.recv(1024).decode().strip()

        if username not in accounts or accounts[username]['password'] != password:
            conn.send('Invalid credentials. Disconnecting...'.encode())
            conn.close()
            return

    except:
        conn.send('Error during authentication. Disconnecting...'.encode())
        conn.close()
        return

    # Perform banking operations
    while True:
        try:
            conn.send('\nEnter your choice (balance/deposit/withdraw/exit): '.encode())
            choice = conn.recv(1024).decode().strip()

            if choice == 'balance':
                conn.send(f'Your balance is: {accounts[username]["balance"]} USD'.encode())
            elif choice == 'deposit':
                conn.send('Enter the amount to deposit: '.encode())
                amount = int(conn.recv(1024).decode().strip())
                accounts[username]['balance'] += amount
                conn.send(f'\nDeposit successful. Your new balance is: {accounts[username]["balance"]} USD'.encode())
            elif choice == 'withdraw':
                conn.send('\nEnter the amount to withdraw: '.encode())
                amount = int(conn.recv(1024).decode().strip())

```

```

        if amount > accounts[username]['balance']:
            conn.send('\nInsufficient funds. Withdrawal
failed.'.encode())
        else:
            accounts[username]['balance'] -= amount
            conn.send(f'\nWithdrawal successful. Your new balance is:
{accounts[username]["balance"]} USD'.encode())
            elif choice == 'exit':
                conn.send(f'\nThank you for using our services. Your final
balance is: {accounts[username]["balance"]} USD'.encode())
                conn.close()
                print(f'Client {addr} disconnected')
                return
            else:
                conn.send('Invalid choice. Please try again.'.encode())
    except:
        conn.send('Error during banking operation.
Disconnecting...'.encode())
        conn.close()
        print(f'Client {addr} disconnected')
        return

while True:
    conn, addr = server_socket.accept()
    client_thread = threading.Thread(target=handle_client, args=(conn, addr))
    client_thread.start()

```

الكلاينت (العميل)

شرح الكود الخاص بالعميل

1. إنشاء سوكت العميل:

- يتم إنشاء سوكت باستخدام socket.socket() وتكوين الاتصال بالخادم على localhost ومنفذ 8000.

2. توثيق العميل:

- يتم استقبال رسالة من الخادم تطلب اسم المستخدم وكلمة المرور.
- يتم إرسال اسم المستخدم وكلمة المرور إلى الخادم للتحقق.

3. تنفيذ العمليات البنكية:

- يدخل العميل في حلقة لاستقبال الرسائل من الخادم.
- يتم استقبال الاختيار من العميل (التحقق من الرصيد، الإيداع، السحب، أو الخروج) وإرساله إلى الخادم.
- بناءً على الاختيار، يتم تنفيذ العملية المناسبة واستقبال الرد من الخادم.
- إذا قام العميل باختيار exit، يتم استقبال الرصيد النهائي وقطع الاتصال.

```
import socket
```

```
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```

client_socket.connect(('localhost', 8000))
msg=client_socket.recv(1024).decode()
# Authenticate the client
username = input(msg)
client_socket.send(username.encode())
msg=client_socket.recv(1024).decode()
password = input(msg)
client_socket.send(password.encode())

# Handle banking operations
while True:
    msg = client_socket.recv(1024).decode()
    print(msg)
    choice = input()
    client_socket.send(choice.encode())

    if choice == 'balance':
        balance = client_socket.recv(1024).decode()
        print(balance)
    elif choice == 'deposit':
        msg=client_socket.recv(1024).decode()
        amount = int(input(msg))
        client_socket.send(str(amount).encode())
        response = client_socket.recv(1024).decode()
        print(response)
    elif choice == 'withdraw':
        msg=client_socket.recv(1024).decode()
        amount = int(input(msg))
        client_socket.send(str(amount).encode())
        response = client_socket.recv(1024).decode()
        print(response)
    elif choice == 'exit':
        final_balance = client_socket.recv(1024).decode()
        print(final_balance)
        client_socket.close()
        break
    else:
        print('Invalid choice. Please try again.')

```

OUTPUT:

نعرض الان 3 عملاء متصلين بنفس الوقت

```
Run: Server x client 1 x client 2 x client 3 x
C:\pythonProject\venv\Scripts\python.exe "C:\pythonProject\السؤال الأول\client 1.py"
Enter your username: لين
Enter your password: 2971

Enter your choice (balance/deposit/withdraw/exit):
balance
Your balance is: 3000 USD

Enter your choice (balance/deposit/withdraw/exit):
|

Run: Server x client 1 x client 2 x client 3 x
C:\pythonProject\venv\Scripts\python.exe "C:\pythonProject\السؤال الأول\client 2.py"
Enter your username: ديانا
Enter your password: 2461

Enter your choice (balance/deposit/withdraw/exit):
deposit
Enter the amount to deposit: 100

Deposit successful. Your new balance is: 2100 USD

Enter your choice (balance/deposit/withdraw/exit):
|

Run: Server x client 1 x client 2 x client 3 x
C:\pythonProject\venv\Scripts\python.exe "C:\pythonProject\السؤال الأول\client 3.py"
Enter your username: دلامه
Enter your password: 2382

Enter your choice (balance/deposit/withdraw/exit):
withdraw

Enter the amount to withdraw: 400

Withdrawal successful. Your new balance is: 600 USD

Enter your choice (balance/deposit/withdraw/exit):
exit

Thank you for using our services. Your final balance is: 600 USD

Process finished with exit code 0
```

السيرفر

```
Run: Server x client 1 x client 2 x client 3 x
C:\pythonProject\venv\Scripts\python.exe "C:\pythonProject\السؤال الأول\Server.py"
New connection from ('127.0.0.1', 63194)
New connection from ('127.0.0.1', 63207)
New connection from ('127.0.0.1', 63214)
Client ('127.0.0.1', 63214) disconnected
|
```

2

نقوم بإنشاء مجلد باسم templates ومجلد آخر يسمى static نضع في المجلد templates ملفات HTML وفي المجلد static ملفات CSS ضمن مجلد .css

نقوم بكتابة كود ال Flask

```
from flask import Flask, render_template

app = Flask(__name__)

# الرئيسية الصفحة
@app.route('/')
def home():
    return render_template('home.html')

# الاتصال صفحة
@app.route('/contact')
def contact():
    return render_template('contact.html')

# نحن من صفحة
@app.route('/about')
def about():
    return render_template('about.html')

if __name__ == '__main__':
    app.run()
```

قمنا ببناء تطبيق Flask يحتوي على صفتين ويب: الصفحة الرئيسية و صفحة من نحن.

تم استيراد الكلاس Flask لاشتقاق غرض منه يمثل التطبيق و التابع render_template للتعامل مع ملفات html من مكتبة flask ومن ثم تم تعريف التطبيق باستخدام (app = Flask(__name__))

تم تعريف تابعين للصفحتين واستخدام الديكوريتور app.route لتحديد عنوان URL الذي يؤدي إلى الصفحة المطلوبة. على سبيل المثال الصفحة الرئيسية تستخدم العنوان '/' في حين أن صفحة من نحن تستخدم العنوان ./about.

تم استدعاء تابع render_template في كل تابع لإرجاع صفحة HTML معينة على سبيل المثال التابع home يستخدم render_template('index.html') لإرجاع صفحة HTML تسمى index.html والتابع about يستخدم render_template('about.html') لإرجاع صفحة HTML تسمى about.html.

يتم التحقق من أن الملف الذي يحتوي على التطبيق هو الملف الرئيسي باستخدام if __name__ == '__main__': ويتم تشغيل التطبيق باستخدام app.run في وضع التصحيح.

صفحة index:

```
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
  shrink-to-fit=no">
  <title>الثاني السؤال</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="#">الرئيسي الموقع</a>
      <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item active">
            <a class="nav-link" href="/">الرئيسية</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/about">نحن من</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/contact">بنا اتصل</a>
          </li>
        </ul>
      </div>
    </nav>
  </header>
</body>
</html>
```

```

</nav>
</header>

<main>
  <div class="container">
    <div class="row">
      <div class="col-12 text-center mt-5">
        <h1>!الرئيسي الموقع في بك مرحبًا</h1>
        <p>HTML وتصميمات Flask باستخدام إنشاؤه تم الموقع هذا</p>
      </div>
    </div>
  </div>
</main>

<footer class="bg-light py-3">
  <div class="container text-center">
    <p>&copy; 2024 - محفوظة الحقوق كل</p>
  </div>
</footer>

<script src="{ url_for('static', filename='js/jquery.min.js')
}"></script>
<script src="{ url_for('static', filename='js/bootstrap.bundle.min.js')
}"></script>
</body>
</html>

```

<DOCTYPE html!>: تعريف نوع المستند.

<html lang="ar" dir="rtl">: بداية عنصر HTML الذي يحتوي على الصفحة بأكملها وتم تحديد لغة الصفحة باللغة العربية وتوجيه النص باتجاه اليمين-إلى-اليسار.

<head>: الجزء العلوي من الصفحة ويحتوي على المعلومات الوصفية للصفحة.

<meta charset="utf-8">: تحديد ترميز الحروف المستخدم في الصفحة.

<title>السؤال الثاني</title>: تحديد عنوان الصفحة الذي يظهر في شريط عنوان المتصفح.

<link rel="stylesheet" href="{ url_for('static', filename='css/style.css') }">: رابط لملف CSS الذي يحتوي على قواعد التنسيق للصفحة.

<link rel="stylesheet" href="{ url_for('static', filename='css/bootstrap.min.css') }">: رابط لملف CSS الخاص بـ Bootstrap.

<body>: جسم الصفحة، ويحتوي على محتويات الصفحة.

<header>: العنصر الذي يحتوي على محتويات رأس الصفحة، مثل شريط القائمة.

<nav class="navbar navbar-expand-lg navbar-light bg-light">: تضمين قائمة التنقل باستخدام Bootstrap Navbar وتحديد بعض الخصائص الأساسية مثل الخلفية واللون.

الموقع الرئيسي: العنصر الذي يحتوي على عنوان الموقع.

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation"> زر التبديل الذي يستخدم لتبديل القائمة على شاشات صغيرة.

<div class="collapse navbar-collapse" id="navbarNav">: عنصر يحتوي على القائمة ويتم تعيينه باستخدام العنصر data-target في الزر.

<ul class="navbar-nav">: العنصر الذي يحتوي على العناصر النمطية للقائمة مثل الروابط.

<li class="nav-item active">: عنصر فرعي للقائمة ويتضمن رابطاً نشطاً.

الرئيسية: رابط للذهاب إلى الصفحة الرئيسية.

<li class="nav-item">: عنصر فرعي للقائمة، ويتضمن رابطاً غير نشط.

من نحن: رابط للذهاب إلى صفحة من نحن.

<li class="nav-item">: عنصر فرعي للقائمة، ويتضمن رابطاً غير نشط.

اتصل بنا: رابط للذهاب إلى صفحة اتصل بنا.

<main>: العنصر الذي يحتوي على المحتوى الرئيسي للصفحة.

<div class="container">: العنصر الذي يحدد الخلفية والتباعد للمحتوى.

<h1>مرحباً بك في الموقع الرئيسي!</h1>: عنوان الصفحة الرئيسي.

<p>هذا الموقع تم إنشاؤه باستخدام Flask وتصميمات HTML و CSS و Bootstrap.</p>: نص يتضمن معلومات عن الموقع والتقنيات التي تم استخدامها لإنشائه.

<footer>: العنصر الذي يحتوي على المحتوى السفلي للصفحة.

<p>© 2023 - كل الحقوق محفوظة.</p>: نص يحتوي على حقوق النشر والملكية الفكرية للصفحة.

يتم استخدام Bootstrap لتنسيق عنصر القائمة والمحتوى وكذلك تم استخدام ملف CSS مخصص (style.css) لتنسيق المحتوى الخاص بالصفحة.

ملف الـcss:

```
header {
  background-color: #f8f9fa;
  padding: 10px;
  border-bottom: 1px solid #ddd;
}
.main {
  margin-top: 20px;
}
h1 {
  font-size: 36px;
  margin-bottom: 20px;
}
p {
  font-size: 18px;
  line-height: 1.5;
  margin-bottom: 30px;
}
```

يتم تحديد التنسيقات الخاصة بـ header في الجزء العلوي من الملف CSS وتشمل لون الخلفية وحجم الخط وتنسيق الحد السفلي.

تم تحديد التنسيقات التالية للنص:

font-size: يتم تحديد حجم الخط.

line-height: يتم تحديد ارتفاع الخط.

margin-bottom: يتم تحديد التباعد الخاص بالنص أسفل العنصر.

مرحبًا بك في الموقع الرئيسي!

هذا الموقع تم إنشاؤه باستخدام Flask وتصميمات HTML و CSS و Bootstrap.

© 2024 - كل الحقوق محفوظة.

صفحة about:

```
<!DOCTYPE html>
<html>
  <head>
    <title>نحن من</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">
    <link rel="stylesheet" href="{{ url_for('static',
filename='css/bootstrap.min.css') }}">
  </head>
  <body>
    <header>
      <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="#">الرئيسي الموقع</a>
        <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">
          <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="/">الرئيسية</a>
            </li>
            <li class="nav-item active">
              <a class="nav-link" href="/about">نحن من</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/contact">اتصل بنا</a>
            </li>
          </ul>
        </div>
      </nav>
    </header>
```

```
<main>
<div class="container">
  <h1>نحن من</h1>
  <div class="mb-4">
    <p>عثمان ناجح دلامه</p>
    <p>2382</p>
  </div>
  <div class="mb-4">
    <p>ديانا عدنان سلوم</p>
    <p>2461</p>
  </div>
  <div class="mb-4">
    <p>شناني الكريم عبد لين</p>
    <p>2971</p>
  </div>
</div>
</main>

<footer>
  <div class="container">
    <p>&copy; 2024 - محفوظة الحقوق كل</p>
  </div>
</footer>
</body>
</html>
```

اتصل بنا • نحن العربية • الموقع الرئيسي

من نحن

دلامه ناجح عثمان

2382

ديانا عدنان سلوم

2461

لين عبد الكريم شناني

2971

كل الحقوق محفوظة - © 2023

صفحة contact:

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

    <title>بنا اتصل</title>
    <link rel="stylesheet" href="{{ url_for('static',
```

```

filename='css/style.css') }}">
    <link rel="stylesheet" href="{{ url_for('static',
filename='css/bootstrap.min.css') }}">
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="#">الرئيسي الموقع</a>
            <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">
                <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="/">الرئيسية</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/about">نحن من</a>
                    </li>
                    <li class="nav-item active">
                        <a class="nav-link" href="/contact">اتصل
بنا</a>
                    </li>
                </ul>
            </div>
        </nav>
    </header>

    <main>
        <div class="container">
            <h1>نحن اتصل</h1>
            <p>promail0101@gmail.com</p>
        </div>
    </main>

    <footer>
        <div class="container">
            <p>&copy; 2024 - محفوظة الحقوق كل</p>
        </div>
    </footer>
</body>
</html>

```