Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No1

الجمهورية العربية السورية

اللاذقية -جامعـــة تشريـــن

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة1 برمجة شبكات

الياس هيثم ملحم – ٢٠٨٦ - https://github.com/eliasmelhem/assignment-2

Question 1:The tcp server

```python
import socket, threading,json
sSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM )
sSocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
sSocket.bind(('127.0.0.1', 4444))
sSocket.listen(5)
print("server ON....")
count=0
store={}
with open("Question.json","r") as Question:
    o_fQ = json.load(Question)
student_answer=[]
with open("Answers.json","r") as Answer:
    o_fA = json.load(Answer)
def handle (C,cd):
    while True:
        try:
            for key,value in o_fQ.items():
                C.send(key.encode())
                print(key+str(cd))
                C.send(value.encode())
                rmsg = C.recv(1024).decode()
                student_answer.append(rmsg)
                if key=='Question20':
                    break
            print (student_answer)
            count=0
            for i in range(0,20):
                if student_answer[i] == o_fA[i]:
                    count+=1
            C.send(str(count).encode())
            break
        except socket.error as e:
            print(e)
        except KeyError as e:
            C.send("The name does not exist yet".encode())
    print("Finish the connection with ", cd)
```

```
previously_clients=[]
while True:
    C, cd = sSocket.accept()
    previously_clients.append(cd)
    thr1 = threading.Thread(target=handle, args=(C, cd))
    thr1.start()
    print("wait another client...")
```

The tcp client

```
import socket, sys

cSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
cSocket.settimeout(10)
try:
    cSocket.connect(('127.0.0.1',4444))
except socket.error as error:
    print('EXCEPTION: ', error)
    sys.exit(1)

while True:

    try:
        for i in range(20):
            data0 = cSocket.recv(1024).decode()
            print(data0)
            data00 = cSocket.recv(1024).decode()
            print(data00)
            answer=input("the answer is: ")
            cSocket.send(answer.encode())
            if i ==20 :
                break


        data = cSocket.recv(1024).decode()
        print("The Mark for your exam is :"+data)
        break
    except socket.error as error:
        print("EXCEPTION 2: ", error)
print('END..')
cSocket.close()

```

The tcp server output

```
server ON....
wait another client...Question1('127.0.0.1', 55953)

Question2('127.0.0.1', 55953)
Question3('127.0.0.1', 55953)
Question4('127.0.0.1', 55953)
Question5('127.0.0.1', 55953)
Question6('127.0.0.1', 55953)
Question7('127.0.0.1', 55953)
Question8('127.0.0.1', 55953)
Question9('127.0.0.1', 55953)
Question10('127.0.0.1', 55953)
Question11('127.0.0.1', 55953)
Question12('127.0.0.1', 55953)
Question13('127.0.0.1', 55953)
Question14('127.0.0.1', 55953)
Question15('127.0.0.1', 55953)
Question16('127.0.0.1', 55953)
Question17('127.0.0.1', 55953)
Question18('127.0.0.1', 55953)
Question19('127.0.0.1', 55953)
Question20('127.0.0.1', 55953)
['a', 'b', 'c', 'a', 'b', 'b', 'b', 'c', 'c', 'a', 'a', 'a', 'a', 'a', 'b', 'c', 'a', 'c', 'a', 'a']
Finish the connection with  ('127.0.0.1', 55953)
```

The tcp client

```
Question1
a b c
the answer is: a
Question2
a b c
the answer is: b
Question3
a b c
the answer is: c
Question4
a b c
the answer is: a
Question5
a b c
the answer is: b
Question6
a b c
the answer is: b
Question7
a b c
the answer is: b
Question8
a b c
the answer is: c
Question9
a b c
the answer is: c
Question10
a b c
the answer is: a
Question11
a b c
the answer is: a
```

```
the answer is: a
Question15
a b c
the answer is: b
Question16
a b c
the answer is: c
Question17
a b c
the answer is: a
Question18
a b c
the answer is: c
Question19
a b c
the answer is: a
Question20
a b c
the answer is: a
The Mark for your exam is :10
END..
```

يقوم السيرفر بإرسال الأسئلة الواحدة تلو الاخر الى الكلاينت وينتظر الإجابة ويخزنها ضمن مصفوفة خاصة كل سؤال على حدي

وفي نفس الوقت يقوم بالاستماع في حال وجود مستخدمين اخرين سوف يخضعون لنفس الامتحان

وفي نهاية الاختبار يقوم السيرفر بمقارنة إجابات الطالب بملف مخزن به الإجابات الصحيحة ويرسل علامة الطالب الى الكلاينت لتظهر له نتيجته مباشرة ويقوم بأنهاء الاتصال معه

ويستمر بالاستماع في حال وجود مستخدمين اخرين

Question 2:

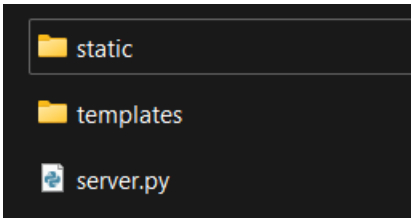
Flask Server

```python
import flask
from flask import Flask, render_template
app = flask.Flask(__name__)
@app.route('/')
def home():
    return render_template("index.html")
@app.route('/main.html')
def f1():
    return render_template("main.html")
@app.route('/page1.html')
def f2():
    return render_template("page1.html")
@app.route('/page3.html')
def f3():
    return render_template("page2.html")
if __name__ == '__main__':
    app.run(port=8888)
```

```
 * Serving Flask app 'server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:8888
Press CTRL+C to quit
```

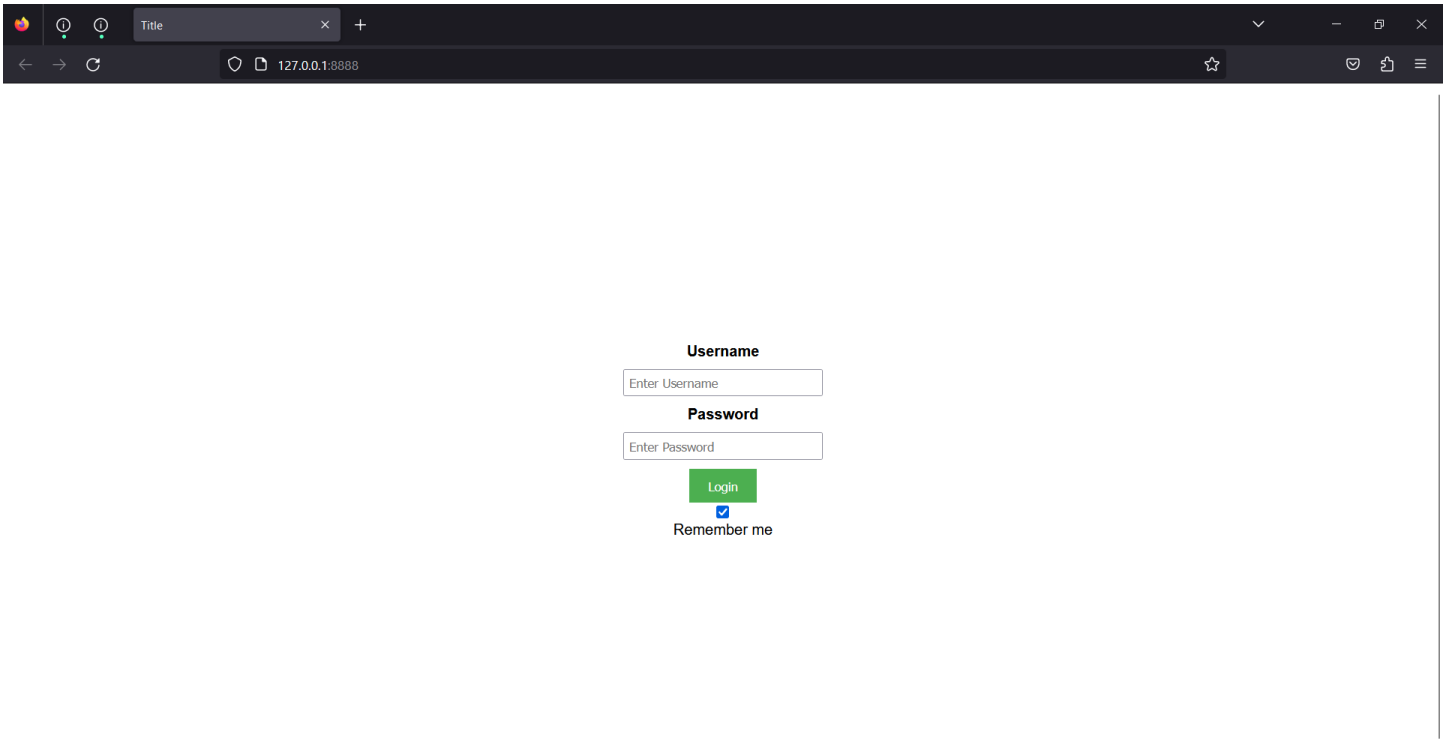
تم العمل باستخدام سيرفر فلاسك

حيث تم ترتيب الملفات كالتالي

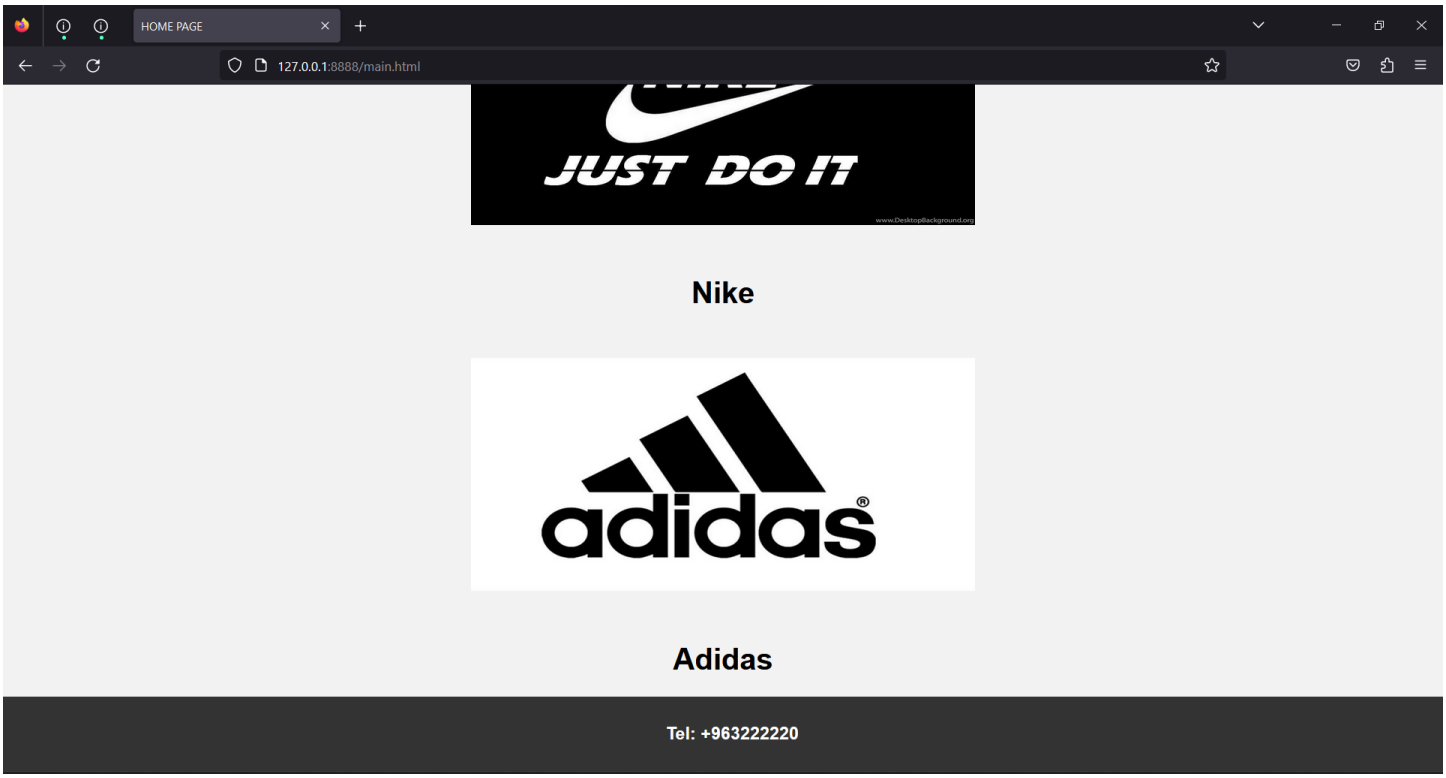


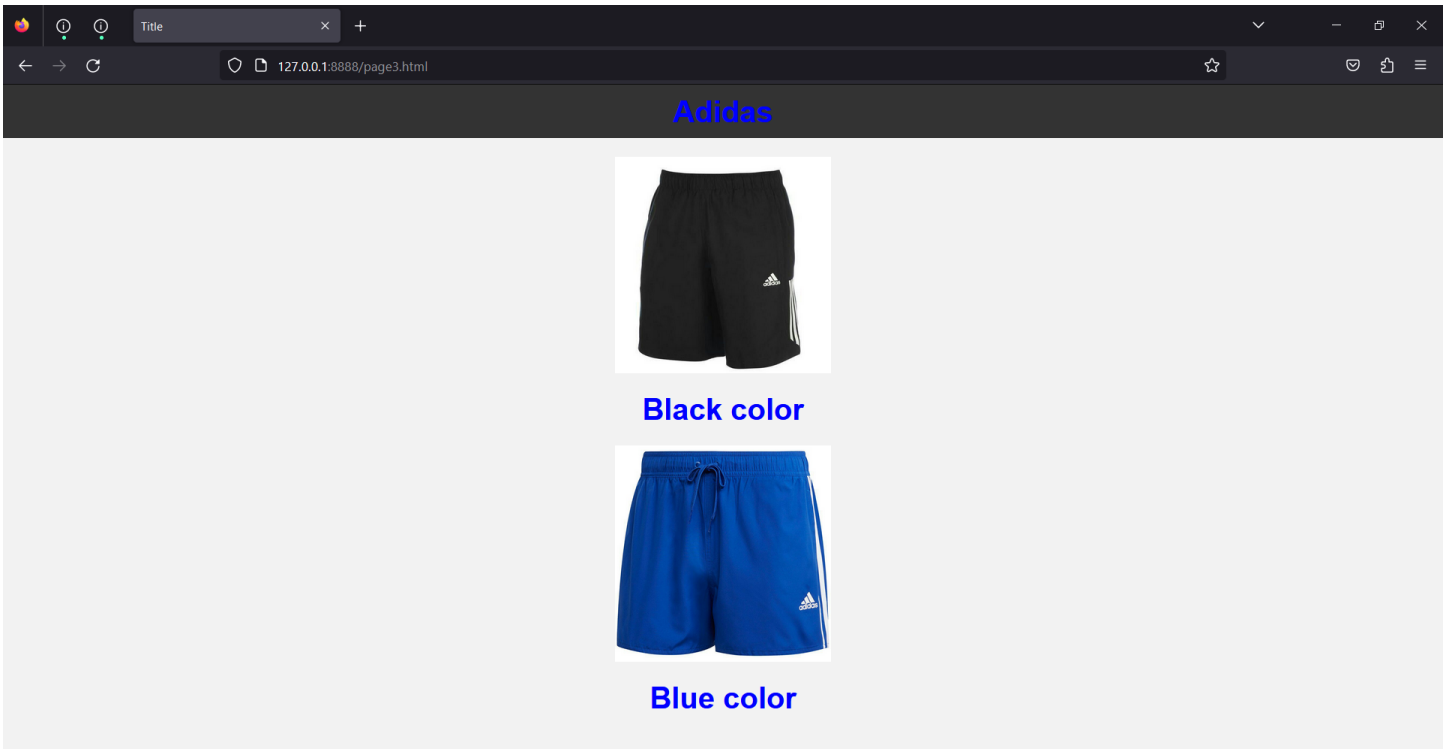
يحوي المجلد static الصور المستخدمة ضمن صفحة الويب

بينما يحوي المجلد templates الصفحات المكتوبة بلغة HTML

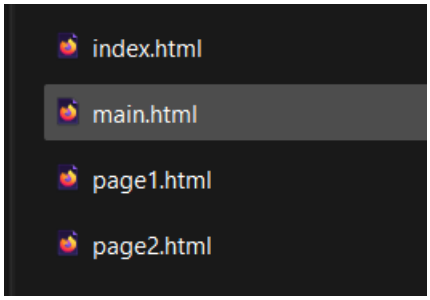


عند الدخول الى العنوان http://127.0.0.1:8888/ تظهر صفحة البداية و منها نستطيع الدخول الى المتجر و التنقل بين الصفحات الثانوية





صور الملفات



كود ال html لصفحة الا index البداية

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <style>
        body {
```

```
            font-family: Arial, sans-serif;
        }

        .container {
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }

        div {
            margin-bottom: 10px;
            margin-top: 10px;
        }

        input[type="text"],
        input[type="password"] {
            padding: 5px;
            width: 200px;
        }

        button[type="submit"] {
            padding: 10px 20px;
            background-color: #4CAF50;
            color: white;
            border: none;
            cursor: pointer;
            margin-top: 10px;
        }

        button[type="submit"]:hover {
            background-color: #45a049;
        }

        input[type="checkbox"] {
            margin-right: 5px;
        }
    </style>
</head>
<body>
    <div class="container">
        <div>
            <b>Username</b>
        </div>
        <input type="text" placeholder="Enter Username">

        <div>
            <b>Password</b>
        </div>
```

```
        <input type="password" placeholder="Enter Password">

        <a href="main.html"><button type="submit">Login</button></a>

        <input type="checkbox" checked="checked" name="remember"> Remember me
    </div>
</body>
</html>
```