Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No2

علي سميح أسعد 2555

الجمهورية العربية السورية

اللاذقية -جامعــة تشريـــن

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة2 برمجة شبكات

Question 1: TCP Server/Client Quiz App with Multi-threading?

As an improvement to previous first homework, build a TCP server and client quiz application using Python. The server should handle multiple client connections simultaneously using multi-threading. The application should allow clients to connect, participate in a quiz, and receive their quiz scores upon completion.

Requirements:

A. The server should be able to handle multiple client connections concurrently.

B. The quiz should consist of a set of pre-defined questions stored on the server.

C. Each client should connect to the server and receive the quiz questions.

D. Clients should send their answers to the server.

E. The server should keep track of the scores for each client.

F. At the end of the quiz, the server should send the final scores to each client.

Solution:

An overview on the program

__ first we turn on the server and let it wait for a client

__ when a client connects with server the program will create a new thread to the client so that the server can serve multiple clients simultaneously

__ the client enters his name and then the server sends the questions to the client in sequence to answer them

__ finally the server shows the name of the client and his result on the screen

__ handle_client is used to handle clients

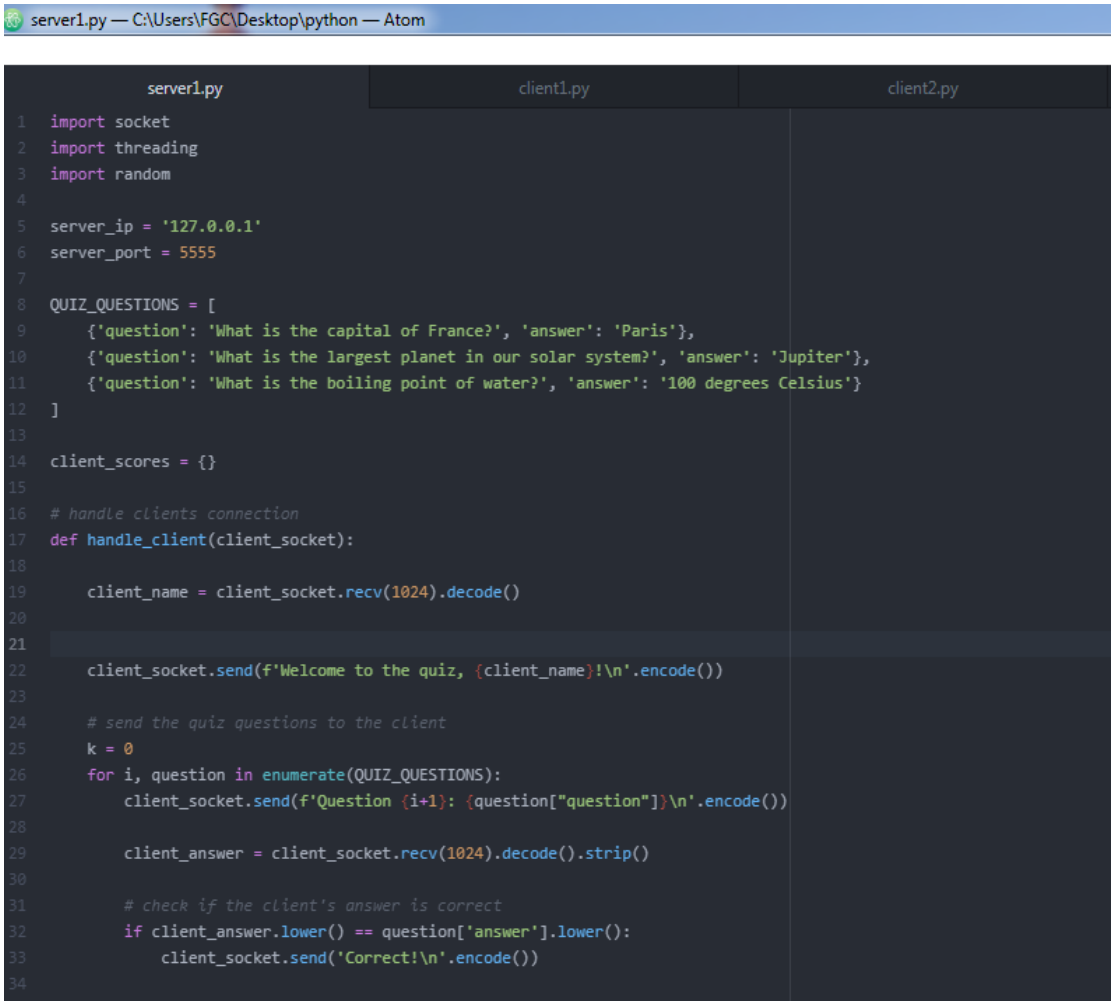
__ before the data is sent, it is encoded and the it is sent using the send() method

__ the data is also received using the recv() method

__ the server waits for client using the accept() method

__ server works on the loop back address (127.0.0.1) and the port (5555)

Server's Code :


server1.py — C:\Users\FGC\Desktop\python — Atom
server1.py
client1.py
client2.py
import socket
import threading
import random
server_ip = '127.0.0.1'
server_port = 5555
QUIZ_QUESTIONS = [
{'question': 'What is the capital of France?', 'answer': 'Paris'},
{'question': 'What is the largest planet in our solar system?', 'answer': 'Jupiter'},
{'question': 'What is the boiling point of water?', 'answer': '100 degrees Celsius'}
]
client_scores = {}
# handle clients connection
def handle_client(client_socket):
client_name = client_socket.recv(1024).decode()
client_socket.send(f'Welcome to the quiz, {client_name}!\n'.encode())
# send the quiz questions to the client
k = 0
for i, question in enumerate(QUIZ_QUESTIONS):
client_socket.send(f'Question {i+1}: {question["question"]}\n'.encode())
client_answer = client_socket.recv(1024).decode().strip()
# check if the client's answer is correct
if client_answer.lower() == question['answer'].lower():
client_socket.send('Correct!\n'.encode())

```
            client_scores[client_name] = client_scores.get(client_name, 0) + 1
            k = k + 1
        else:
            client_socket.send(f'Incorrect! The correct answer is {question["answer"]}.\n'.encode())
    if k == 0 :
        client_scores[client_name] = client_scores.get(client_name, 0) + 0
    # send the client's score to the client
    client_socket.send(f'Thank you for playing, {client_name}! Your score is {client_scores[client_name]}.\n'.encode())

    client_socket.close()

# Create the server socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server_socket.bind((server_ip, server_port))
server_socket.listen()

print(f'Server listening on {server_ip}:{server_port}')

while True:
    client_socket, client_address = server_socket.accept()

    # create a new thread to handle the client connection
    thread = threading.Thread(target=handle_client, args=(client_socket,))
    thread.start()

```

Client1 code :

server1.py | client1.py | client2.py

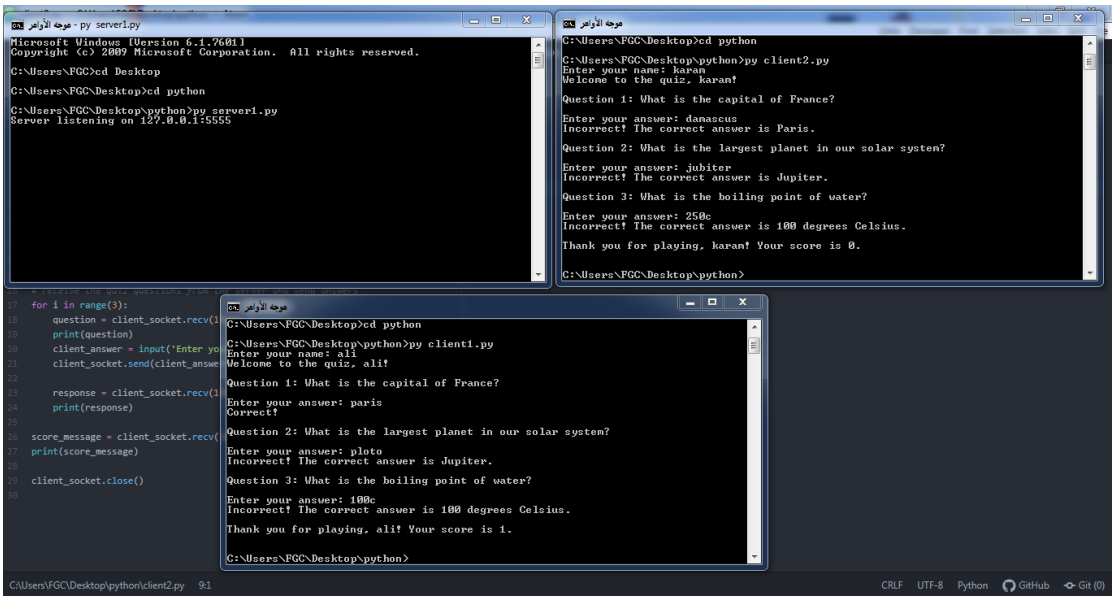
```
import socket

server_ip = '127.0.0.1'
server_port = 5555

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

client_socket.connect((server_ip, server_port))

client_name = input('Enter your name: ')
client_socket.send(client_name.encode())

welcome_message = client_socket.recv(1024).decode()
print(welcome_message)

# receive the quiz questions from the server and send answers
for i in range(3):
    question = client_socket.recv(1024).decode()
    print(question)
    client_answer = input('Enter your answer: ')
    client_socket.send(client_answer.encode())

    response = client_socket.recv(1024).decode()
    print(response)

score_message = client_socket.recv(1024).decode()
print(score_message)

client_socket.close()

```

Client2 code :

server1.py | client1.py | client2.py

```python
import socket

server_ip = '127.0.0.1'
server_port = 5555

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

client_socket.connect((server_ip, server_port))

client_name = input('Enter your name: ')
client_socket.send(client_name.encode())

welcome_message = client_socket.recv(1024).decode()
print(welcome_message)

# receive the quiz questions from the server and send answers
for i in range(3):
    question = client_socket.recv(1024).decode()
    print(question)
    client_answer = input('Enter your answer: ')
    client_socket.send(client_answer.encode())

    response = client_socket.recv(1024).decode()
    print(response)

score_message = client_socket.recv(1024).decode()
print(score_message)

client_socket.close()

```

Program execution :



Question 2: Simple Website with Python Flask Framework

Create a simple website with multiple pages using Flask, HTML, CSS, and Bootstrap. The website should demonstrate your understanding of web design principles

. Requirements:

A. Set up a local web server using XAMPP, IIS, or Python's built-in server (using Flask)

. B. Apply CSS and Bootstrap to style the website and make it visually appealing.

C. Ensure that the website is responsive and displays correctly on different screen sizes.

D. Implement basic server-side functionality using Flask to handle website features.

Solution :

solution's steps :

1_ to work with flask we create a virtual environment that contains the project

2_ to create a virtual environment we enter the folder that we want to work in it

Then we write this command in the command line python –m venv (name)

3_ to activate the virtual environment we copy the path of the activate.bat file in the scripts folder then we enter the path in the command line and we attach to it \activate.bat

__ the project that has been done , is a simple website that shows the offers on shoes and it consists of pages

The backend code of the website :

flask1.py

```python
from flask import Flask, redirect, url_for,  render_template
import os
app = Flask(__name__)


picture = os.path.join('static','photo')
app.config['UPLOAD_FOLDER'] = picture

@app.route("/")
def index():
    return render_template("index1.html")

@app.route("/product")
def product():
    offer = os.path.join(app.config['UPLOAD_FOLDER'], 'offer.jpg')
    return render_template("index.html", user_image = offer)

@app.route("/complain")
def complain():
    return render_template("index2.html")


if __name__ =="__main__":
    app.run(port=8888)

```

The frontend code of the website : the home page

index.html | index1.html | index2.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="static/bootstrap.css" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>shoe store</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg bg-body-tertiary">
      <div class="container-fluid">
        <button
          class="navbar-toggler"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#navbarSupportedContent"
          aria-controls="navbarSupportedContent"
          aria-expanded="false"
          aria-label="Toggle navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="#">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/product" target="_blank">Offers</a>
            </li>
          </ul>
          <form class="d-flex" role="search">
            <input
```

```
            <form class="d-flex" role="search">
              <input
                class="form-control me-2"
                type="search"
                placeholder="Search"
                aria-label="Search"
              />
              <button class="btn btn-outline-success" type="submit">
                Search
              </button>
            </form>
          </div>
        </div>
      </nav>
      <h3>About us</h3>
      <ul>
        <li>
          our main concern is to provide you with the best shoes offers that you
          can find
        </li>
        <li>please click on offers to see the avialable sales</li>
        <li>
          if you run into any problem please contact with our support team using
          the email below
        </li>
        <li><a href="/complain">shop.store@gmail.com</a></li>
      </ul>
    </body>
</html>
```

The products page code

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" href="static/style.css" />
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Nike Produckt</title>
  </head>
  <body>
    <div class="main-div">
      <h2>Nike Air Max shoes</h2>
      <div class="div">
        <img
          src="{{
          user_image
          }}"
          alt="The Shose photo"
          width="250px"
          height="250px"
        />
        <div class="div1">
          <div class="div1-sub">
            <p class="prise">$65.00</p>
            <p class="shipping">free shipping</p>
          </div>
          <p class="discription">Step into style: Exclusive shoe Deals</p>
          <a class="link" href="#" target="blank">more information </a>
          <div>
            <button class="button button1"></button>
            <button class="button button2"></button>
            <button class="button button3"></button>
            <button class="button button4"></button>
            <button class="button button5"></button>
            <button class="button button6"></button>
```

```
            <button class="button button6"></button>
          </div>
        </div>
        <div class="div2">
          <p class="produckt">produckt details</p>
          <ul class="u-list">
            <li>made with high-quality materials</li>
            <li>available with customizable options</li>
            <li>feature a visible Air cushioning</li>
          </ul>
        </div>
      </div>
      <button class="last-button">add to cart</button>
      <p class="sale">sale</p>
    </div>
  </body>
</html>
```

The complain page :

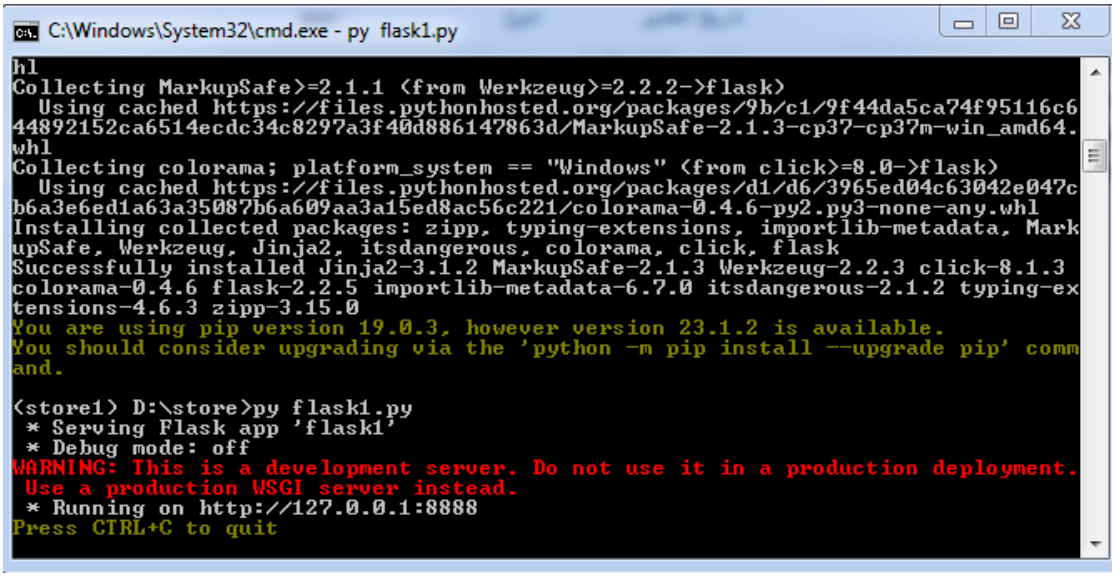
index.html | index1.html | index2.html

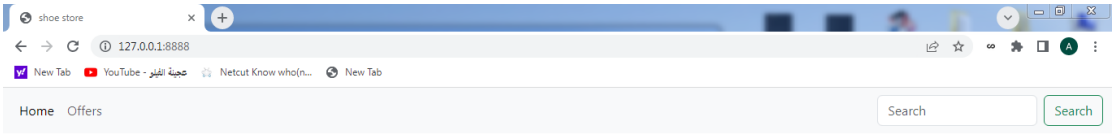
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <link rel="stylesheet" href="static/bootstrap.css" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>complains</title>
  </head>
  <body>
    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label"
        >Email address</label
      >
      <input
        type="email"
        class="form-control"
        id="exampleFormControlInput1"
        placeholder="name@example.com"
      />
    </div>
    <div class="mb-3">
      <label for="exampleFormControlTextarea1" class="form-label"
        >Please Enter your complain</label
      >
      <textarea
        class="form-control"
        id="exampleFormControlTextarea1"
        rows="3"
      ></textarea>
    </div>
  </body>
</html>

```

the website's execution :

__ in order to show the website we have first to run the server after we activate the virtual environment

__ then we open a web browser and we type in the search field the ip address of the website





About us

- our main concern is to provide you with the best shoes offers that you can find
- please click on offers to see the avialable sales
- if you run into any problem please contact with our support team using the email below
- shop.store@gmail.com