



Name: Zainab Kanaan, Number: 2525, Submitted To GitHub: zeinab-kanaan

Second Network Programming Homework

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:

- The server should be able to handle multiple client connections concurrently.
- The quiz should consist of a set of pre-defined questions stored on the server.
- Each client should connect to the server and receive the quiz questions.
- Clients should send their answers to the server.
- The server should keep track of the scores for each client.
- At the end of the quiz, the server should send the final scores to each client.

Guidelines:

- Use Python's socket module "don't use 3rd-party packages".
- Implement multi-threading to handle multiple client connections concurrently.
- Store the quiz questions and correct answers on the server side.

Notes:

- Write brief report describing the design choices you made and any challenges faced during implementation.
- You can make a **TCP Server/Client of your choice**, such as Bank ATM, Chat application, or any other appropriate application that fulfil all requirements.

```
File Edit Format Run Options Window Help
import socket, threading

quiz_questions = {
    "x+y=? :a.9 or b.10": "a",
    "x-y=? :a.1 or b.-1": "b",
    "x+z=? :a.9 or b.10": "b",
    "x+y+z=? :a.14 or b.10": "a",
    "z+y-x=? :a.9 or b.7": "b",
    "x*y=? :a.9 or b.20": "b",
    "y/x=? :a.1 or b.10": "a",
    "x*x*y? :a.80 or b.10": "a",
    "y*y*x? :a.1 or b.100": "b",
    "z*z*x? :a.144 or b.10": "a",
    "z-y-x? :a.-3 or b.10": "a",
    "z-x? :a.2 or b.10": "a",
    "z*y/x ? :a.7.5 or b.10": "a",
    "x*z/y? :a.1 or b.3.34": "b",
    "y*y-x*x? :a.9 or b.10": "a",
    "x*x+z*y ? :a.46 or b.10": "a",
    "y*z-2x? :a.22 or b.10": "a",
    "2x+2z-y? :a.1 or b.10": "b",
    "3x-z*y? :a.30 or b.10": "a",
    "z-4y+6x? :a.1 or b.10": "b"
}

client_scores = {}

def handle_new_client(client_socket, client_address):
    try:
        client_socket.send(str(len(quiz_questions)).encode())
```

Name: Zainab Kanaan, Number: 2525, Submitted To GitHub: zeinab-kanaan

```

try:
    client_socket.send(str(len(quiz_questions)).encode())

    for question in quiz_questions:
        client_socket.send(question.encode())

        client_answer = client_socket.recv(1024).decode().strip()

        if client_answer.lower() == quiz_questions[question].lower():
            client_scores[client_address] = client_scores.get(client_address, 0) + 1

        score = client_scores.get(client_address, 0)
        client_socket.send(f"Your Score: {score}/{len(quiz_questions)}\n".encode())

except ConnectionAbortedError:
    print(f"Connection aborted by the client: {client_address}")

client_socket.close()
print(f"Disconnected client: {client_address}")

def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    server_address = ('localhost', 2525)
    server_socket.bind(server_address)

    server_socket.listen(5)
    print("Quiz Server started. Waiting for connections...")

    while True:
        client_socket, client_address = server_socket.accept()
        print(f"Connected to client: {client_address}")

        client_thread = threading.Thread(target=handle_new_client, args=(client_socket, client_address))
        client_thread.start()

if __name__ == '__main__':
    start_server()

```

يتم استيراد المكتبات اللازمة socket و threading.

يتم تعريف مجموعة من الأسئلة والإجابات في قاموس quiz_questions.

يتم تعريف قاموس client_scores لتخزين درجات العملاء.

يتم تعريف وظيفة handle_new_client للتعامل مع اتصال العميل.

يتم تعريف وظيفة start_server لبدء السيرفر والاستماع لاتصالات العملاء.

في دالة start_server يتم تعيين عنوان المضيف (host) والمنفذ (port)، ويتم إنشاء مأخذ الخادم (server_socket) وربطه بالعنوان والمنفذ المحددين.

بعد ذلك، يتم قبول اتصال العميل وإنشاء مأخذ (client_socket) للتواصل معه.

يتم إنشاء (client_thread) لكل عميل يتصل

يتم بدء thread للتعامل مع العميل.

يتكرر هذا العملية لقبول اتصالات العملاء الجديدة.

وفي وظيفة handle_new_client، يتم التعامل مع العميل الجديد على النحو التالي:

يتم إرسال عدد الأسئلة للعميل

يتم إرسال أسئلة الواحدة تلو الأخرى إلى العميل.



Name: Zainab Kanaan , Number: 2525 , Submitted To GitHub: zeinab-kanaan
يتم استقبال إجابة العميل من المأخذ (client_socket).

يتم التحقق مما إذا كانت الإجابة الصحيحة بالمقارنة مع الإجابة المخزنة في قاموس quiz_questions باستخدام if statement.

إذا كانت الإجابة صحيحة، يتم زيادة درجة العميل في قاموس client_scores باستخدام client_scores[client_name] = client_scores.get(client_name, 0) + 1.

يتم إرسال الدرجة النهائية إلى العميل عن طريق إنشاء رسالة تحتوي على اسم العميل ودرجته.

يتم معالجة الأخطاء في حال حدوثها

يتم إغلاق المأخذ (client_socket) لإنهاء الاتصال مع العميل.

يتم طباعة رسالة توضح أن الاتصال تم إغلاقه مع عنوان العميل.

وأخيرًا، في الجزء النهائي من الكود، يتم استدعاء الدالة start_server لبدء السيرفر عند تشغيل البرنامج.

```
File Edit Format Run Options Window Help
import socket
print ("x=4","y=5","z=6")
client = input('Enter Name :')
def start_client():
    server_host = 'localhost'
    server_port = 2525
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    try:
        client_socket.connect((server_host, server_port))
        print(f"Connected to server {server_host}:{server_port}")

        num_questions = int(client_socket.recv(1024).decode())

        for i in range(num_questions):
            question = client_socket.recv(1024).decode()

            answer = input(f"{question}: ")

            client_socket.sendall(answer.encode())

        final_score = client_socket.recv(1024).decode()
        print("Dear ",client," ,Your score is :",final_score)

    except ConnectionRefusedError:
        print("Failed to connect to the server.")
    finally:
        client_socket.close()

if __name__ == '__main__':
    start_client()
```

يتم استيراد المكتبة socket.

يتم تعريف وظيفة start_client لبدء العميل والاتصال بالسيرفر.

في دالة start_client يتم تعيين عنوان المضيف (host) والمنفذ (port) المستخدمين في الاتصال.

يتم إنشاء مأخذ العميل (client_socket) باستخدام socket.socket(socket.AF_INET, socket.SOCK_STREAM).

يتم استخدام client_socket.connect((host, port)) للاتصال بالسيرفر.

يتم استقبال عدد الأسئلة من السيرفر

ندخل حلقة مجالها عدد الاسئلة



Name: Zainab Kanaan, Number: 2525, Submitted To GitHub: zeinab-kanaan

يتم طباعة السؤال ويتم طلب الإجابة من المستخدم باستخدام `input()`، ثم يتم إرسال الإجابة إلى السيرفر باستخدام `client_socket.send(answer.encode())`.

يتم استقبال الدرجة النهائية من السيرفر باستخدام `client_socket.recv(1024).decode()` ويتم طباعتها.

يتم إغلاق المآخذ (`client_socket`) لإنهاء الاتصال مع السيرفر.

وفي الجزء النهائي من الكود، يتم استدعاء الدالة `start_client` لبدء العميل عند تشغيل البرنامج.

```
Administrator: C:\Windows\System32\cmd.exe - python Server.py
new connection from ('127.0.0.1', 63069)
new connection from ('127.0.0.1', 63070)
new connection from ('127.0.0.1', 63071)
connection closed with ('127.0.0.1', 63069)
connection closed with ('127.0.0.1', 63070)
connection closed with ('127.0.0.1', 63071)

Enter your answer (a/b): a
x*x*y/z :a.80 or b.10
Enter your answer (a/b): a
y*y*x/z :a.1 or b.100
Enter your answer (a/b): a
z*z*x/z :a.144 or b.10
Enter your answer (a/b): a
z-y-x? :a.-3 or b.10
Enter your answer (a/b): a
z-x? :a.2 or b.10
Enter your answer (a/b): a
z*y/x? :a.7.5 or b.10
Enter your answer (a/b): a
x*z/y/z :a.1 or b.3.34
Enter your answer (a/b): a
y*y-x*x? :a.9 or b.10
Enter your answer (a/b): a
x*x+z*y? :a.46 or b.10
Enter your answer (a/b): a
y*z-2x? :a.22 or b.10
Enter your answer (a/b): a
2x+2z-y? :a.1 or b.10
Enter your answer (a/b): a
3x-z*y? :a.30 or b.10
Enter your answer (a/b): a
z-4y+6x? :a.1 or b.10
Enter your answer (a/b): a
Dear zainab, your score is: 12

Enter your answer (a/b): b
x*x*y/z :a.80 or b.10
Enter your answer (a/b): a
y*y*x/z :a.1 or b.100
Enter your answer (a/b): b
z*z*x/z :a.144 or b.10
Enter your answer (a/b): b
z-y-x? :a.-3 or b.10
Enter your answer (a/b): b
z-x? :a.2 or b.10
Enter your answer (a/b): b
z*y/x? :a.7.5 or b.10
Enter your answer (a/b): b
x*z/y/z :a.1 or b.3.34
Enter your answer (a/b): b
y*y-x*x? :a.9 or b.10
Enter your answer (a/b): b
x*x+z*y? :a.46 or b.10
Enter your answer (a/b): b
y*z-2x? :a.22 or b.10
Enter your answer (a/b): b
2x+2z-y? :a.1 or b.10
Enter your answer (a/b): b
3x-z*y? :a.30 or b.10
Enter your answer (a/b): b
z-4y+6x? :a.1 or b.10
Enter your answer (a/b): b
Dear dina, your score is: 8

Enter your answer (a/b): b
x*x*y/z :a.80 or b.10
Enter your answer (a/b): a
y*y*x/z :a.1 or b.100
Enter your answer (a/b): b
z*z*x/z :a.144 or b.10
Enter your answer (a/b): a
z-y-x? :a.-3 or b.10
Enter your answer (a/b): b
z-x? :a.2 or b.10
Enter your answer (a/b): a
z*y/x? :a.7.5 or b.10
Enter your answer (a/b): b
x*z/y/z :a.1 or b.3.34
Enter your answer (a/b): a
y*y-x*x? :a.9 or b.10
Enter your answer (a/b): b
x*x+z*y? :a.46 or b.10
Enter your answer (a/b): a
y*z-2x? :a.22 or b.10
Enter your answer (a/b): b
2x+2z-y? :a.1 or b.10
Enter your answer (a/b): a
3x-z*y? :a.30 or b.10
Enter your answer (a/b): b
z-4y+6x? :a.1 or b.10
Enter your answer (a/b): a
Dear ali, your score is: 8
```



Name: Zainab Kanaan, Number: 2525, Submitted To GitHub: zeinab-kanaan

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:

- Set up a local web server using XAMPP, IIS, or Python's built-in server (using Flask).
- Apply CSS and Bootstrap to style the website and make it visually appealing.
- Ensure that the website is responsive and displays correctly on different screen sizes.
- Implement basic server-side functionality using Flask to handle website features.

```

1  from flask import Flask, render_template
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def index():
7      return render_template('index.html')
8
9  @app.route('/about')
10 def about():
11     return render_template('about.html')
12
13
14 @app.route('/contact')
15 def contact():
16     return render_template('contact.html')
17
18 if __name__ == '__main__':
19     app.run(debug=True, port=9000)
20

```

يتم استيراد Flask و render_template من وحدة Flask.

`from flask import Flask, render_template`

يتم إنشاء تطبيق Flask باستخدام الكود `app = Flask(__name__)`. يتم تمرير `__name__` كمعامل لتحديد اسم التطبيق وتحديد موقع ملفات القالب.

يتم تعريف المسارات (routes) باستخدام المزخرف `@app.route()`.

المسار `template/` يعود إلى الصفحة الرئيسية ويتم تعيينه لدالة `home()`.

المسار `template/about/` يعود إلى صفحة "about" ويتم تعيينه لدالة `about()`.



Name: Zainab Kanaan, Number: 2525, Submitted To GitHub: zeinab-kanaan

المسار `template/contact/` يعود إلى صفحة `contact` " ويتم تعيينه لدالة `contact()` .

السطر `__if __name__ == '__main__':` يتحقق مما إذا كان البرنامج يتم تشغيله مباشرة عن طريق تشغيل البرنامج

الرئيسي. إذا كان الشرط صحيحًا، فإنه يشغل التطبيق بتفعيل وضع التصحيح (debug mode) بواسطة الأمر

`app.run(debug=True)`. يتم العمل على البورت 9000

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Engineering Book Library</title>
5 <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
6 <link rel="stylesheet" href="static/css/style.css">
7 </head>
8 <body>
9 <header>
10 <nav>
11 <ul>
12 <li><a href="{{ url_for('index') }}">Home</a></li>
13 <li><a href="{{ url_for('about') }}">About</a></li>
14 <li><a href="{{ url_for('contact') }}">Contact</a></li>
15 </ul>
16 </nav>
17 </header>
18 <div class="container mt-5">
19 <h1>Welcome to the Engineering Book Library!</h1>
20 
21
22 <h2>Available Books:</h2>
23 <table class="table">
24 <thead>
25 <tr>
26 <th>Book Title</th>
27 <th>Author</th>
28 <th>Specialization</th>
29 </tr>
30 </thead>
31 <tbody>
32 <tr>
33 <td>Introduction to Engineering</td>
34 <td>John Smith</td>
35 <td>General Engineering</td>
36 </tr>
37 <tr>
38 <td>Communications Engineering Principles</td>
39 <td>David Anderson</td>
40 <td>Communications Engineering</td>

```

Name: Zainab Kanaan , Number: 2525 , Submitted To GitHub: zainab-kanaan

```

EXPLORER
  OPEN EDITORS 2 unsaved
    app.py
    index.html templates
  FLASK
    __pycache__
    .vscode
    static
    css
    images
    js
    bootstrap-4.0.0-dist.zip
    templates
      about.html
      contact.html
      index.html
    app.py
  OUTLINE
  TIMELINE

templates > index.html > html > body > div.container.mt-5
  37 <table>
  38 <tr>
  39 <td>Communications Engineering Principles</td>
  40 <td>David Anderson</td>
  41 <td>Communications Engineering</td>
  42 </tr>
  43 <tr>
  44 <td>Communications Engineering</td>
  45 <td>David Anderson</td>
  46 <td>Communications Engineering</td>
  47 </tr>
  48 <tr>
  49 <td>Communications Engineering 2</td>
  50 <td>David Anderson</td>
  51 <td>Communications Engineering</td>
  52 </tr>
  53 <tr>
  54 <td>Mechanical Engineering Basics</td>
  55 <td>Emily Johnson</td>
  56 <td>Mechanical Engineering</td>
  57 </tr>
  58 <tr>
  59 <td>Electrical Engineering Principles</td>
  60 <td>Michael Davis</td>
  61 <td>Electrical Engineering</td>
  62 </tr>
  63 <tr>
  64 <td>Civil Engineering Fundamentals</td>
  65 <td>Sarah Wilson</td>
  66 <td>Civil Engineering</td>
  67 </tr>
  68 <tr>
  69 <td>Chemical Engineering Essentials</td>
  70 <td>Robert Thompson</td>
  71 <td>Chemical Engineering</td>
  72 </tr>
  73 </tbody>
  74 </table>
  75 </div>
  76 </body>
  77 </html>

```

```

EXPLORER
  OPEN EDITORS 2 unsaved
    app.py
    index.html templates
    about.html templates
  FLASK
    __pycache__
    .vscode
    static
    css
    images
    js
    bootstrap-4.0.0-dist.zip
    templates
      about.html
      contact.html
      index.html
    app.py
  OUTLINE
  TIMELINE

templates > about.html > ...
  1 <!DOCTYPE html>
  2 <html>
  3 <head>
  4 <title>Engineering Book Library - About</title>
  5 <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
  6 <link rel="stylesheet" href="static/css/style.css">
  7 </head>
  8 <body>
  9 <div class="container mt-5">
  10 <div class="row">
  11 <div class="col">
  12 <h1>About the Engineering Book Library</h1>
  13 
  14 <p>The Engineering Book Library is a comprehensive collection of books covering various engineering disciplines. Our mission is to provide a v
  15 </p>
  16 <h2>Our Team</h2>
  17 <p>We have a dedicated team of engineers, educators, and enthusiasts who are passionate about sharing knowledge and promoting the field of en
  18 </p>
  19 <h2>Our Vision</h2>
  20 <p>Our vision is to foster a community of lifelong learners in engineering, empower individuals with the necessary knowledge to excel in thei
  21 </p>
  22 </div>
  23 </div>
  24 </body>
  25 </html>

```



Name: Zainab Kanaan

, Number: 2525

, Submitted To GitHub: zeinab-kanaan

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Engineering Book Library - Contact</title>
5 <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
6 <link rel="stylesheet" href="static/css/style.css">
7 </head>
8 <body>
9 <header>
10 <nav>
11 <ul>
12 <li><a href="{{ url_for('index') }}">Home</a></li>
13 <li><a href="{{ url_for('about') }}">About</a></li>
14 <li><a href="{{ url_for('contact') }}">Contact</a></li>
15 </ul>
16 </nav>
17 </header>
18 <div class="container mt-5">
19 <h1>Contact the Engineering Book Library</h1>
20 
21 <p>Feel free to reach out to us for any questions, suggestions, or inquiries. We are here to assist you!</p>
22
23 <h2>Contact Information</h2>
24 <p>Email: info@engineeringbooklibrary.com</p>
25 <p>Phone: +1-123-456-7890</p>
26
27 <h2>Send us a Message</h2>
28 <form>
29 <div class="form-group">
30 <label for="name">Name</label>
31 <input type="text" class="form-control" id="name" placeholder="Enter your name">
32 </div>
33 <div class="form-group">
34 <label for="email">Email address</label>
35 <input type="email" class="form-control" id="email" placeholder="Enter your email address">
36 </div>
37 <div class="form-group">
38 <label for="message">Message</label>
39 <textarea class="form-control" id="message" rows="5" placeholder="Enter your message"></textarea>
40 </div>
41 <button type="submit" class="btn btn-primary">Submit</button>
42 </form>
43 </div>
44
45 </body>
46 </html>

```

ملف HTML يصف صفحة واجهة المستخدم لمكتبة الكتب الهندسية. يحتوي الملف على هيكل وتنسيق الصفحة وعناصر HTML مختلفة، بالإضافة إلى استخدام CSS ومكتبة Bootstrap للتصميم والتنسيق.

ملخص العناصر الرئيسية في الملف:

<DOCTYPE html!> تعليمية (doctype) تحدد نوع المستند ك HTML5.

<html> عنصر جذر المستند.

<head> يحتوي على المعلومات العامة للصفحة والروابط المستخدمة.

<title> يحدد عنوان الصفحة.

<link> يستخدم لتضمين ملفات CSS الخارجية.



Name: Zainab Kanaan , Number: 2525 , Submitted To GitHub: zeinab-kanaan

<body> يحتوي على محتوى الصفحة.

<header> يحتوي على شريط التنقل (navbar) وعنوان الصفحة.

<main> يحتوي على المحتوى الرئيسي للصفحة.

```

1  body {
2    font-family: Arial, sans-serif;
3    margin: 0;
4    padding: 0;
5    background-image: url("../static/images/1.jfif");
6    background-repeat: no-repeat;
7    background-position: center center;
8    background-attachment: fixed;
9    background-size: cover;
10   background-color: rgba(255, 255, 255, 0.5);
11 }
12
13 .container {
14   max-width: 960px;
15   margin: 0 auto;
16   padding: 20px;
17   background-color: rgba(255, 255, 255, 0.8);
18 }
19
20 .book-image {
21   position: absolute;
22   top: 0;
23   left: 0;
24   width: 100%;
25   height: 100%;
26   opacity: 0.2;
27   z-index: -1;
28 }
29
30 header {
31   background-color: #4287f5;
32   color: #ffffff;
33   padding: 20px;
34 }
35
36 nav ul li a {
37   text-decoration: none;
38   color: #333333;
39   font-weight: bold;
40 }
  
```

Name: **Zainab Kanaan** , Number: 2525 , Submitted To GitHub: **zeinab-kanaan**

```
static > css > # style.css > .book-image
24 width: 100%;
25 height: 100%;
26 opacity: 0.2;
27 z-index: -1;
28 }
29 header {
30   background-color: #4287f5;
31   color: #ffffff;
32   padding: 20px;
33 }
34
35 nav ul li a {
36   text-decoration: none;
37   color: #333333;
38   font-weight: bold;
39 }
40
41 nav ul li {
42   display: inline;
43   margin-right: 10px;
44 }
45 h1 {
46   color: #3366cc;
47   margin-top: 0;
48 }
49
50 h2 {
51   color: #3366cc;
52 }
53
54
55
56 .navbar-brand {
57   font-size: 24px;
58   font-weight: bold;
59   color: #fff;
60 }
61
62
```

Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical
engineering

5th , Network Programming : Homework No2



الجمهورية العربية السورية

اللاذقية - جامعة تشرين

كلية الهندسة الكهربائية والميكانيكية

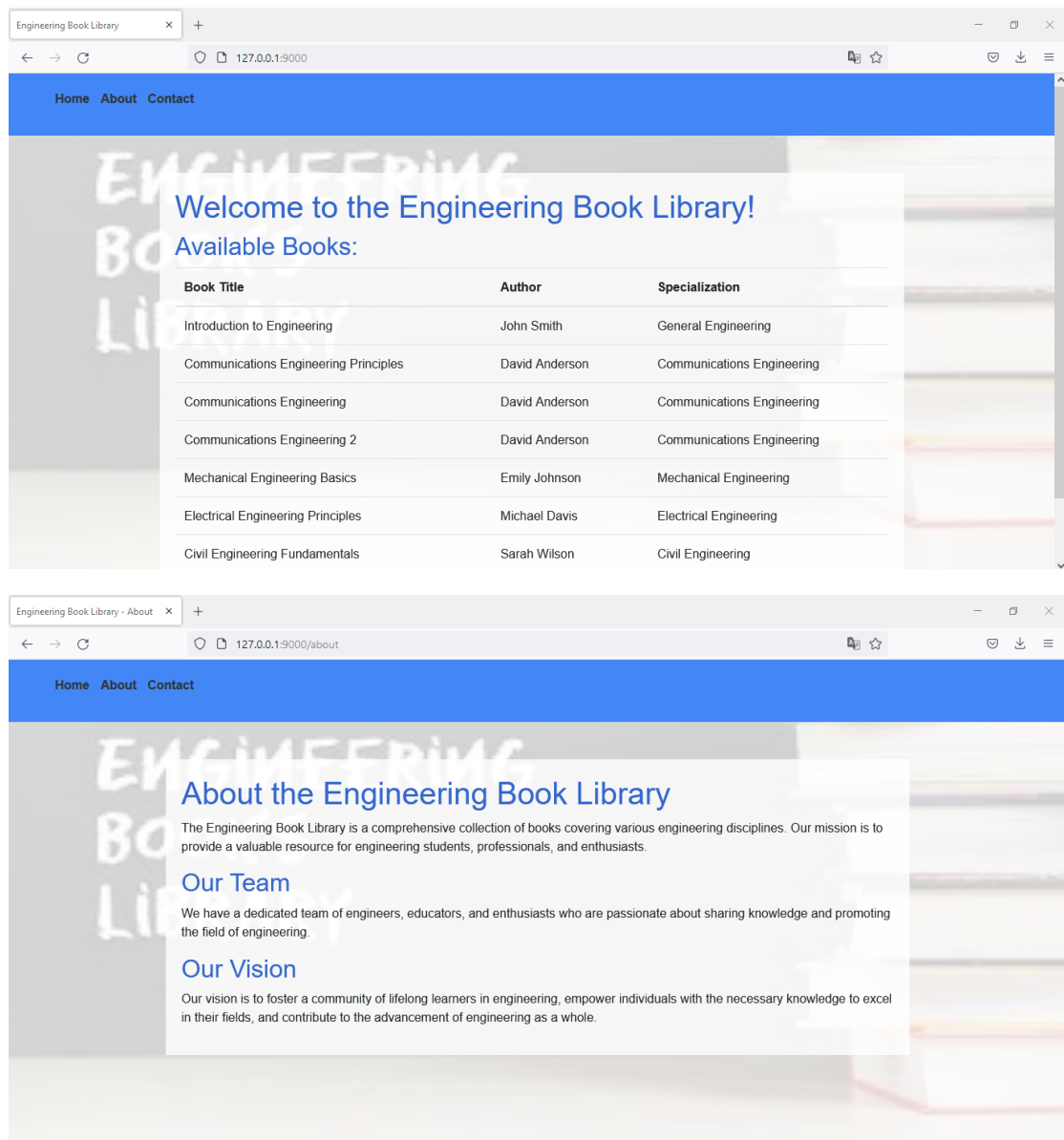
قسم هندسة الاتصالات والإلكترونيات

السنة الخامسة: وظيفة 2 برمجة شبكات

Name: Zainab Kanaan , Number: 2525 , Submitted To GitHub: zeinab-kanaan

نقوم بتشغيل كود app.py

بعد ذلك نضع العنوان التالي في المتصفح <http://127.0.0.1:9000>





Name: Zainab Kanaan , Number: 2525 , Submitted To GitHub: zeinab-kanaan

Engineering Book Library - Contact X +

127.0.0.1:9000/contact

Home About Contact

Contact the Engineering Book Library

Feel free to reach out to us for any questions, suggestions, or inquiries. We are here to assist you!

Contact Information

Email: info@engineeringbooklibrary.com

Phone: +963-888-000-111

Send us a Message

Name

Email address

Engineering Book Library - Contact X +

127.0.0.1:9000/contact

Home About Contact

Contact the Engineering Book Library

Feel free to reach out to us for any questions, suggestions, or inquiries. We are here to assist you!

Contact Information

Email: info@engineeringbooklibrary.com

Phone: +963-888-000-111

Send us a Message

Name

Email address

Message

Submit