ITNE352-Project-Group-SA3

Project for ITNE352 by Ahammed Ismail and Bilal Mohammad Tofeeq ID: 202201478 & 202200507 respectively.

Project Title

Multithreaded Flight Arrival Client/Server Information System

Project Description

A client-server system that exchanges information about flights at a certain airport. the clients can connect with the server and request specific data regarding flight details. such as information arrived flights, delayed flights etc. The emphasis in this project is on client/server architecture, network communication, multithreading, API, and applying good coding practices.

Semester

2nd Semester,2024-2025

Group Details

- Group Number: SA3
- Course Code: ITNE352/ITCE320
- Section Number: 01
- Students Names: Ahammed Ismail ,Bilal Mohammad Tofeeq
- Students IDs: 202201478, 202200507

Table of Contents

Requirements

To establish the project and run the project in local environment follow these steps which are given below:

1. **Install Python** Install the latest version of python from python's original website, which is: (https://www.python.org/downloads/ (https://www.python.org/downloads/))
2. Clone the repository:

```
git clone https://github.com/AhammedIsmail01478/ITNE352-Project-Group-SA3
```

3. Navigate to the closed directory:

```
cd ITNE352-Project-Group-SA3
```

4. Install required dependencies:

```
pip install -r requirements.txt
```

5. **Get a FlightAPI Key**

- Register on aviationstack.com (https://aviationstack.com/)
- Replace the API key used in the server with your own key, a comment has been provided to show the location of the key

How to Run

To run the system:

1. Open one terminal window for server and open multiple terminal windows for multiple clients.

2. In the terminal, go to the directory containing the client and server python files.

3. Start with the server side so it can prepare for incoming requests which can be done by running the script:

```
python server.py
```

4. Start the client side next and then connect to server that is already running using the script:

```
python client.py
```

5. Use the Client - side interface to:
    - Input Client username
    - Navigates Menu

- Interacts and retrieves data

The Scripts

Client Script (`client.py`)

The packages which are used in the Client Script are:

```
import socket
```

Create a Socket and connect to the Server

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as sock:


    try:
        sock.connect(('localhost', 55555))
        print("The request has been sent")
        AckMsg = sock.recv(4096).decode('utf-8')
        print(AckMsg)
```

Ask the Client to input user name

```
   server_msg = sock.recv(4096).decode('utf-8')
        CName = input(server_msg)
        sock.sendall(CName.encode('utf-8'))
```

Server sends the option menu and ask user to choose an option.

```
while True:
            print(sock.recv(4096).decode(), end='')
            choice = input("> ")
            sock.sendall(choice.encode())
```

Recieve a response from the server depending on the option they choose

```
 if choice == '3':  # request flight code
                print(sock.recv(1024).decode(), end='')
                iata = input("> ")
                sock.sendall(iata.encode())

            response = sock.recv(65536).decode()
            print("\n--- Response ---")
            print(response)

            if choice == '4':
                break
```

Error Handling

```
except ConnectionRefusedError:
        print("Error", "Cannot connect to the server.")
    except Exception as e:
        print(f"[ERROR] : {e}")
```

Server Script (`server.py`)

The packages which are used in the Server Script are:

```
import socket
import threading
import json
import requests
```

The following script display the list of active clients

```
def display_active_clients(clients_list):
```

The following script will retrieve the data from API and saving it in JSON file

```
def retrieve_flight_data(API_key, icao):
    url = "http://api.aviationstack.com/v1/flights"
    params = {
        'access_key': API_key,
        'arr_icao': icao,
        'limit': 100
    }
```

The following Script will handle the Clients

```
def handle_client(conn, sockaddr, data):
```

Creating Socket and Bindong

```
if __name__ == "__main__":
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server_sock:
        server_sock.bind(('localhost', 55555))
        print("Server has succesfully started!")
```

Acknowledgments

- Usage of aviationstack.com for API to get flight details
- Lectures with Dr. Mohammad A. Almeer was heavily used and supported us to create the scripts.
- ChatGPT was used to clarify and learnt alot of concepts. Such as the use of 'localhost' instead of writing full IP address
- A big thanks to Dr. Mohammad A. Almeer for teaching us throughtout the course as a whole.

Conclusion

In this project, We have demonstrated effective network programming such as using sockets, threading and requests modules and using them properly as required. We have successfully created a script for the server where the server is capable of retrieving flight details through API and is capable of managing connections and responding to client requests effectively through proper threading. We have also successfully created the script for client to properly establish a connection with the server and sending requests to access data requested by the client users. This project has enhanced our skills in network programming,API management and proper threat handling.