Socket Programming Project Report

Problem Statement

Take the various concepts that has been learned over the course of the semester and integrate it into one project. This includes dictionaries/lists, functions, loops, and most importantly socket programming. The goal was to create two programs, a server and a client, that sent information back and forth to one another.

Device Specifications:

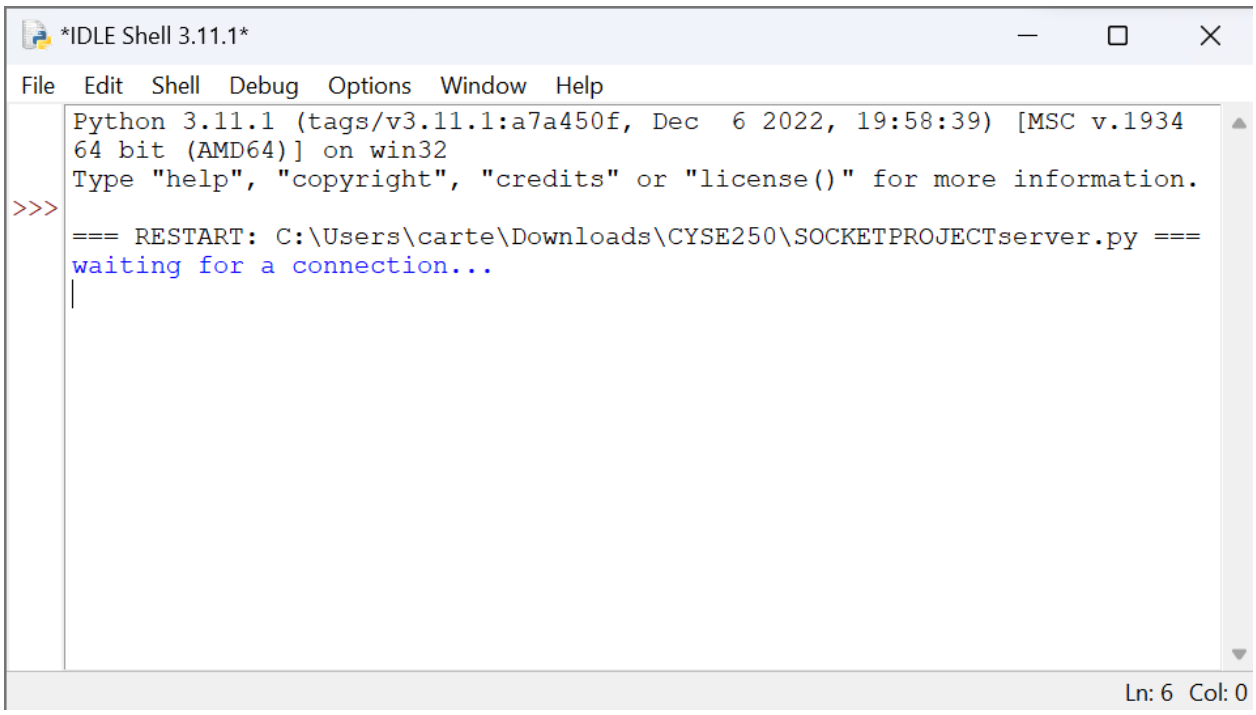
- Device name – HP
- Processor – 11th Gen Intel(R) Core (TM) i3-1125G4 @ 2.00GHz 2.00 GHz
- Installed RAM - 8.00 GB (7.77 GB usable)
- Device ID - 507159B5-9194-4F0A-B7EF-0E6B04D4A78E
- Product ID - 00356-02653-06894-AAOEM
- System type - 64-bit operating system, x64-based processor
- Pen and touch - Pen and touch support with 10 touch points

Software:

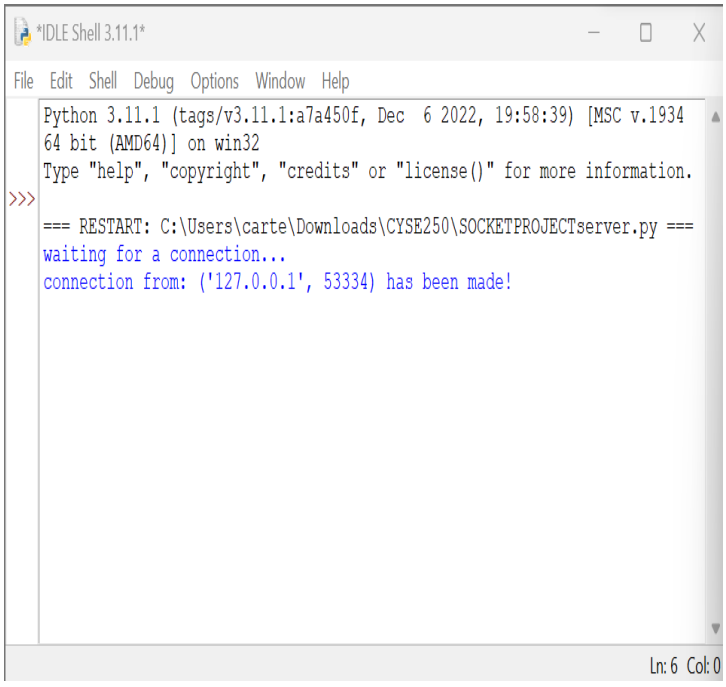
- Windows
    - Edition - Windows 11 Home
    - Version - 22H2
    - Installed on - 1/28/2023
    - OS build - 22621.1555
    - Experience - Windows Feature Experience Pack 1000.22640.1000.0
- Python
    - Python 3.11 64-bit

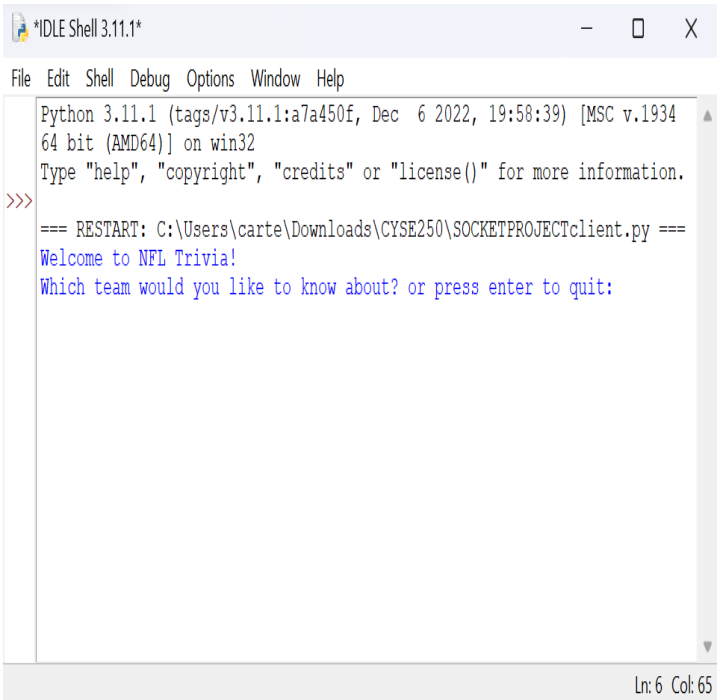
Results and Discussion:

The program I created involves a server and a client, which communicate with one another to operate a mini trivia game. This trivia is based on the NFL, allowing the user to type in the name of a team (mascot specifically), and the client will return the location of the team, as well as the last time (year) that team won the Super Bowl (if applicable). Once the client is connected to the server, the user can then begin the trivia.



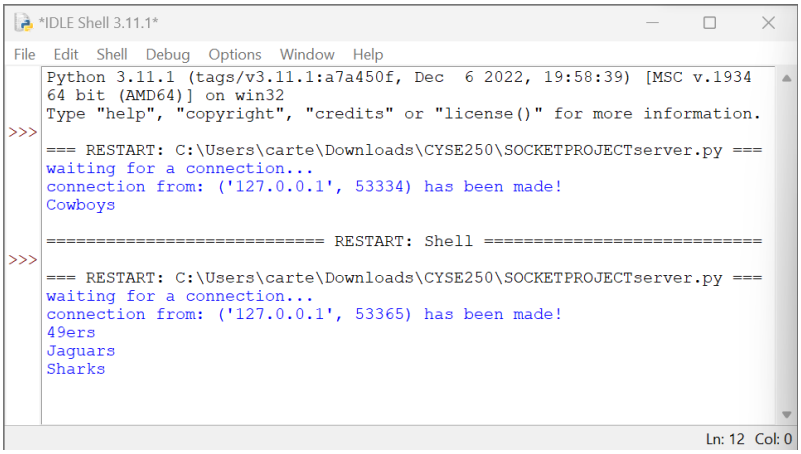
The first step is to run the server program. A message shows up saying waiting for connection, prompting the client being used to connect to the server.

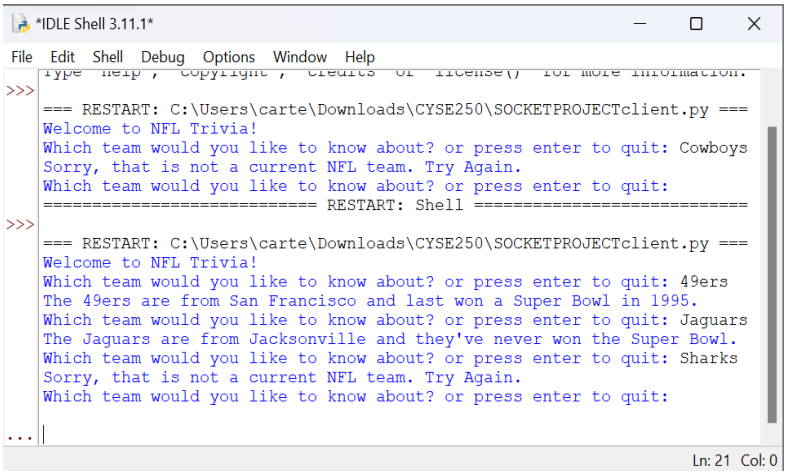




Next, the client server must run in order to make a connection. Once connected, a message on the server side will say “connection from: (address) has been made!”

clarifying that there has been a connection from a client. The client that has connected to the server will see an introduction message saying, “Welcome to NFL Trivia!” following the introductory message, a question will be asked requesting a team name to learn about. This is the input for the loop, and the server will return the facts back to the client.





The user can ask about all 32 NFL teams and receive an informative sentence that is specific to each team and their data. If the user would like to stop at any time, all they have to do is press the enter key when prompted and the connection will end. There are three different messages that the user can receive in total. The location of a team and the year of their last super bowl, the location of a team and a message saying they have never won, and if the user inputs the wrong team it’ll respond with “Sorry, that is not a current NFL team. Try Again.”

Appendix

```
#Defining the host and port
import socket
HOST='localhost'
PORT=5000
BUFSIZE=1024
ADDRESS=(HOST,PORT)
server=socket.socket() #FIRST STEP creating a socket
server.bind(ADDRESS)   #SECOND STEP binding the socket to a certain adress
server.listen(1)

#dictionary/list of all NFL teams and their location, as well as last super bowl
teams = {"Falcons":["Atlanta",0],
         "Cardinals":["Arizona",0],
         "Ravens":["Baltimore",2013],
         "Bills":["Buffalo",0],
         "Panthers":["Carolina",0],
         "Bears":["Chicago",1986],
         "Bengals":["Cincinnati",0],
         "Browns":["Cleveland",0],
         "Cowboys":["Dallas",1996],
         "Broncos":["Denver",2016],
         "Lions":["Detriot",0],
         "Packers":["Green Bay",2011],
         "Texans":["Houston",0],
         "Colts":["Indianapolis",2007],
         "Jaguars":["Jacksonville",0],
         "Chiefs":["Kansas City",2023],
         "Raiders":["Las Vegas",1984],
         "Chargers":["Los Angeles",0],
```

```
        "Rams":["Los Angeles",2022],
        "Dolphins":["Miami",1974],
        "Vikings":["Minnesota",0],
        "Patriots":["New England",2019],
        "Saints":["New Orleans",2010],
        "Giants":["New York",2012],
        "Jets":["New York",1969],
        "Eagles":["Philadelphia",2018],
        "Steelers":["Pittsburgh",2009],
        "49ers":["San Francisco",1995],
        "Seahawks":["Seattle",2014],
        "Buccaneers":["Tampa Bay",2021],
        "Titans":["Tennessee",0],
        "Commanders":["Washington",1992]
        }


def get_team_info(teamName): #function to get the data of the teams
    if teamName in teams: #looking for the information (location/year) of each team
        location = teams[teamName][0]
        year = teams[teamName][1]
        if year == 0:
            return f"The {teamName} are from {location} and they've never won the Super
Bowl."
        else:
            return f"The {teamName} are from {location} and last won a Super Bowl in
{year}."
    else:
        return "Sorry, that is not a current NFL team. Try Again."
```

```
while True: #outer loop to accept connection from client
    print("waiting for a connection...")
    (client,address)=server.accept() #accepting the connection
    print("connection from:",address,"has been made!")
    client.send("Welcome to NFL Trivia!".encode()) #"string".encode, sending starting message to the client



    while True: #inner loop to send the information to the client
        teamName = client.recv(BUFSIZE).decode()
        if not teamName:
            break
        else:
            print(teamName) #ensures that the server received the team name
            team_info = get_team_info(teamName)
            client.send(team_info.encode()) #sends the information of the team to the client
server_socket.close() #close both the socket and the client


#Defining host and port
import socket
HOST='localhost'
PORT=5000
BUFSIZE=1024
ADDRESS=(HOST,PORT)
client=socket.socket()
client.connect(ADDRESS)
print(client.recv(1024).decode()) #Receiving the "Welcome to NFL Trivia!" message

while True: #question for the user to begin the trivia, input
```

```
    teamName = input("Which team would you like to know about? or press enter to quit: ")
    #send the name of the team to the server, then receive the information back and print it
    client.send(teamName.encode()) #sends the input (team name) to the server
    team_info = client.recv(BUFSIZE).decode() #receives the information
    print(team_info) #displays the information sentence


client_socket.close()
```