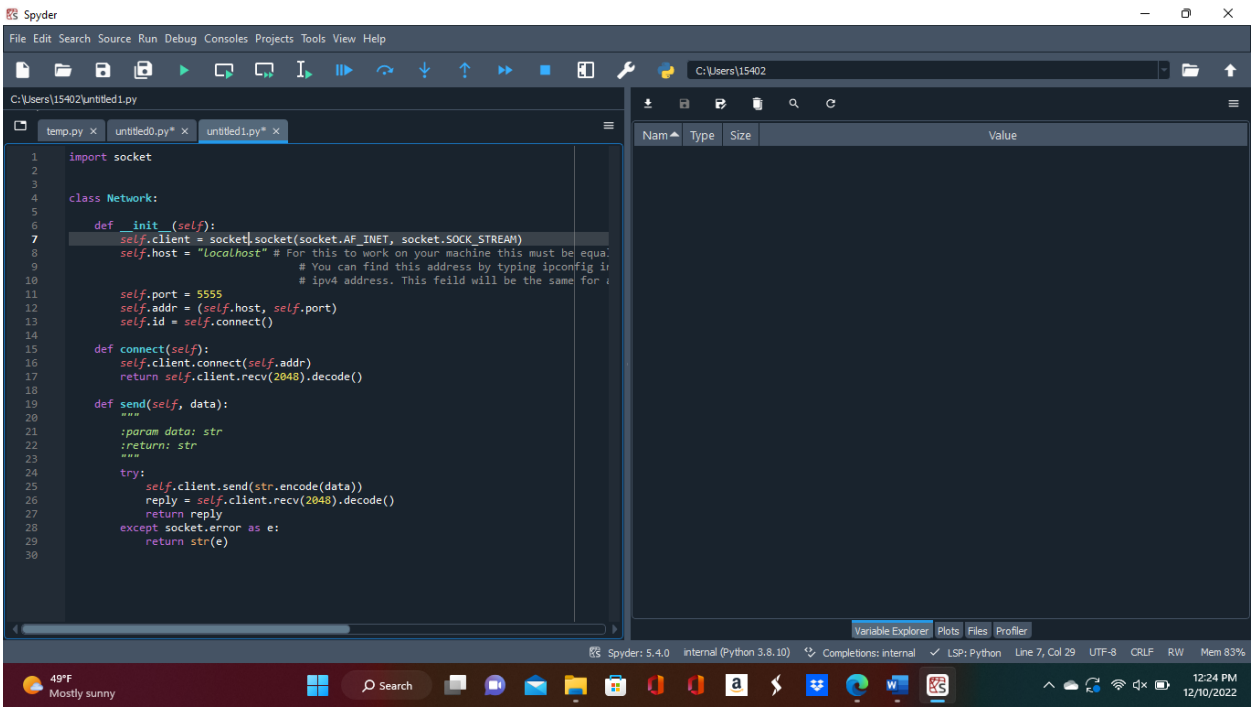
PROBLEM

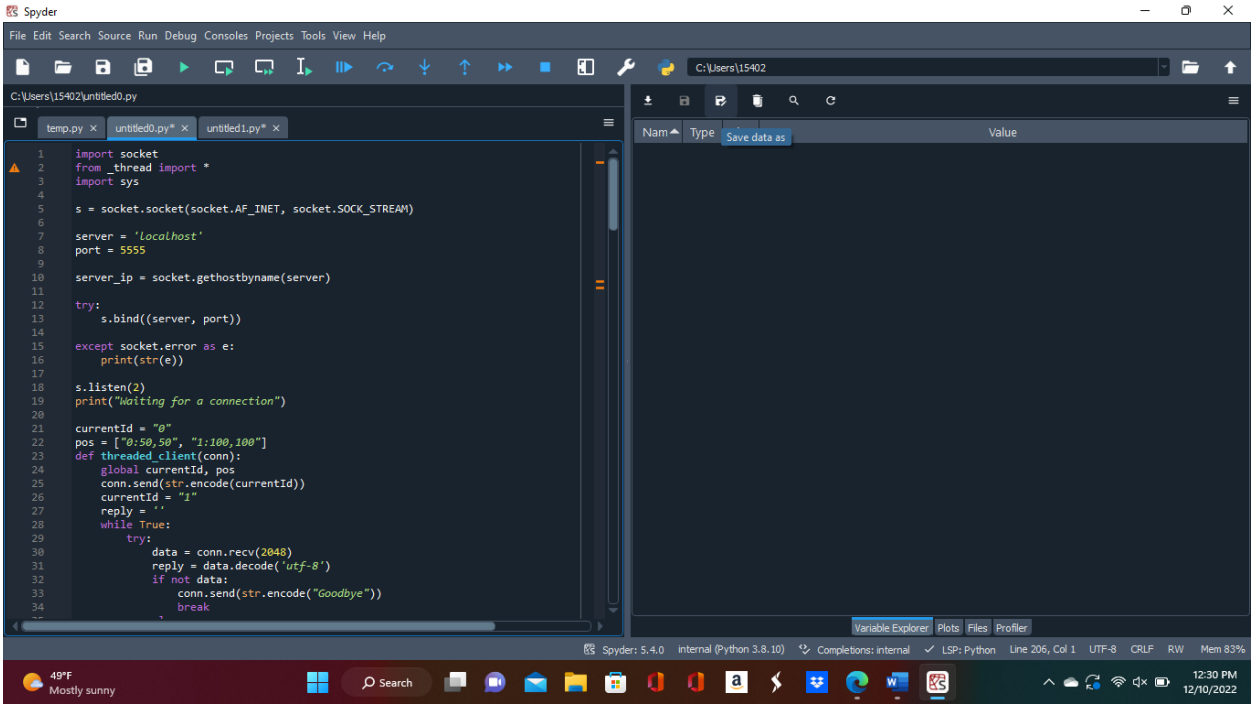
The problem is that Jake wants, to know what the top teams in the NFL every year are in rushing yards, passing yards and run defense, the reason behind this is that he makes bets to make profit from these games, so it is important to know what the best option for the team he chooses. Another reason is that jake doesn't want to lose money, from betting, so jake created hangman in which he will use to generate the team that has the best odds of winning every week from this random processor.

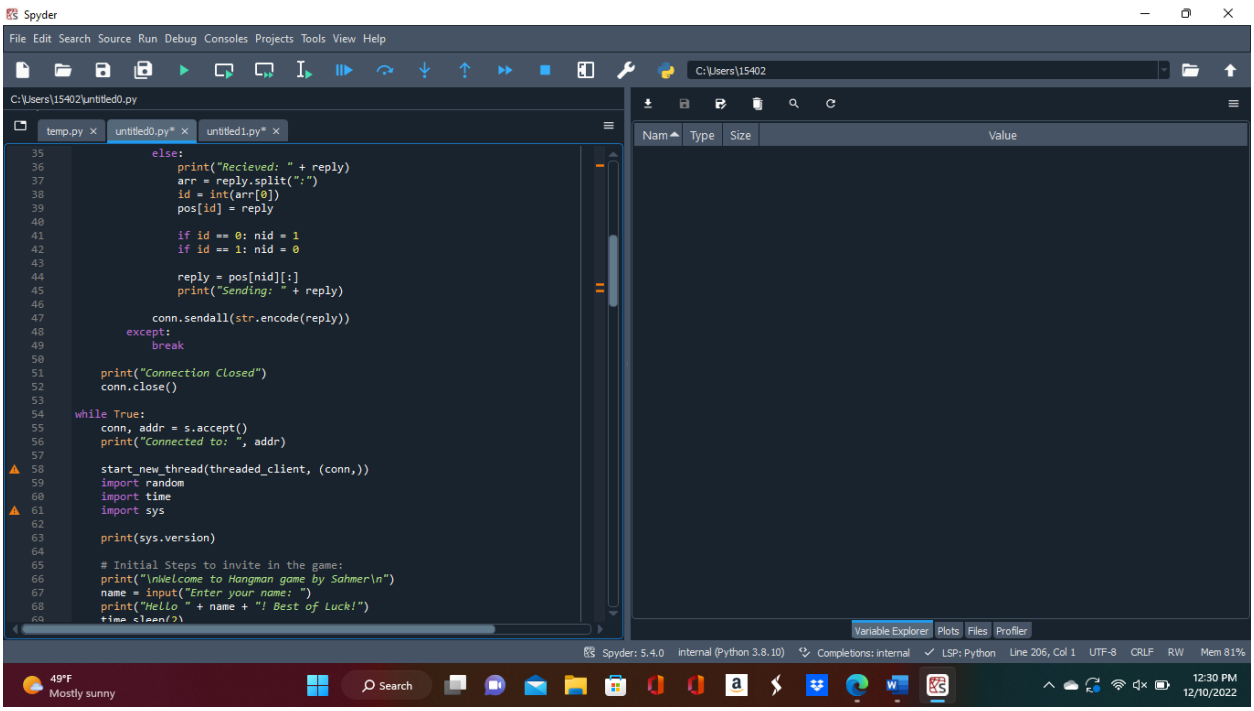
Software used to implement the project

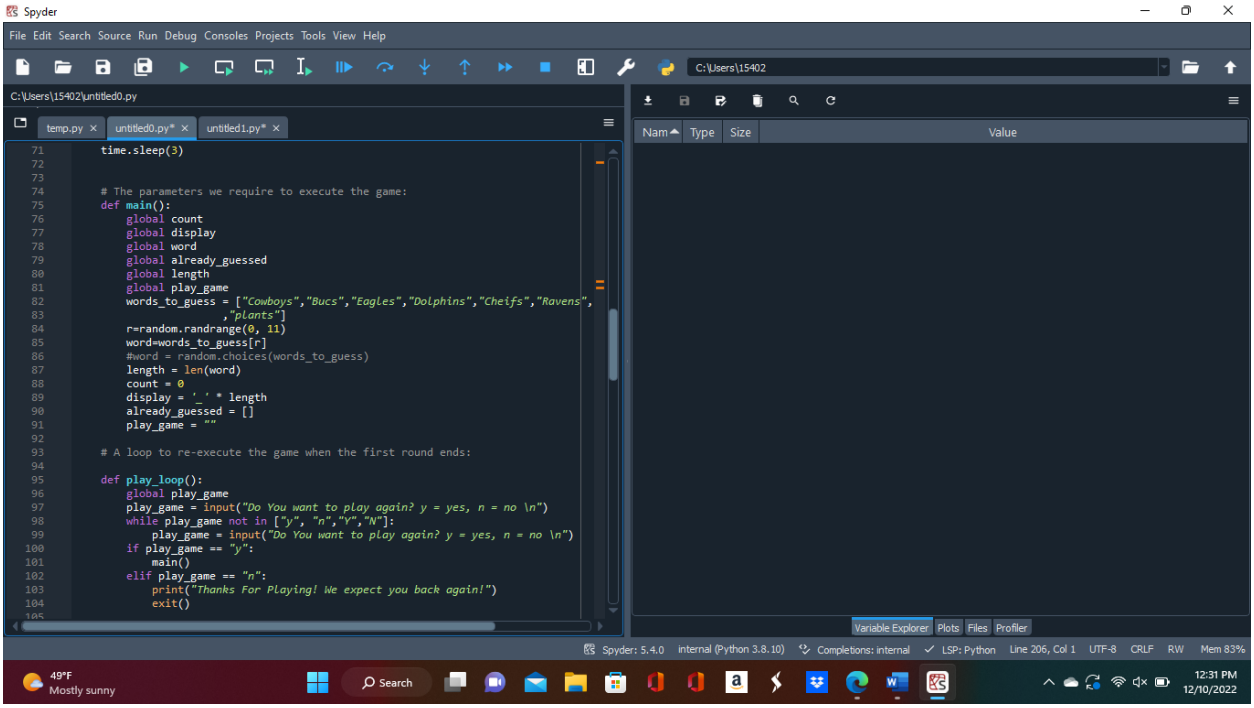
The software that will be used to implement this project will be python, we will use python codes from the client and sever in which they will exchange with one another to play hangman. The sever we will use, is a local host server in which the client will locate the network in which the server is on to access the game of hangman. The software will be used as well is called spider, in which we will supplement are codes into, to create the project.

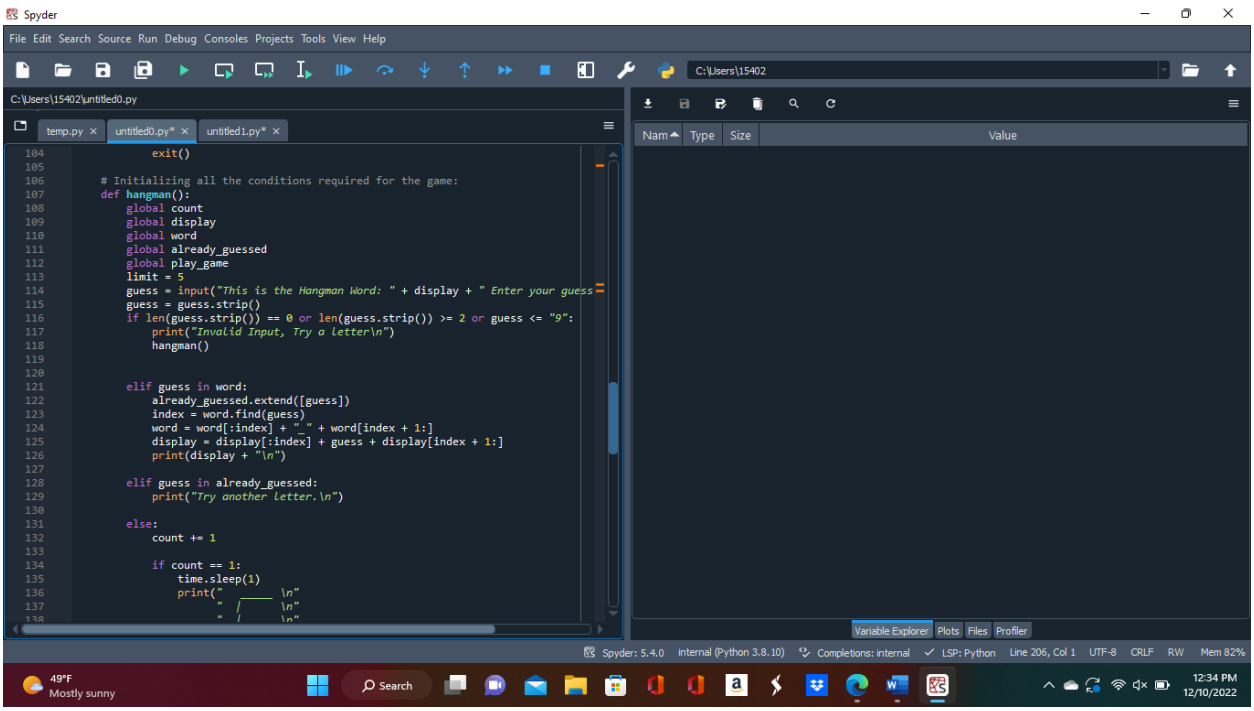
SCREENSHOTS


Spyder
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\15402\untitled1.py
temp.py
untitled0.py*
untitled1.py*
import socket
class Network:
def __init__(self):
self.client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
self.host = "localhost" # For this to work on your machine this must be equa
# You can find this address by typing ipconfig i
# ipv4 address. This feild will be the same for
self.port = 5555
self.addr = (self.host, self.port)
self.id = self.connect()
def connect(self):
self.client.connect(self.addr)
return self.client.recv(2048).decode()
def send(self, data):
"""
:param data: str
:return: str
"""
try:
self.client.send(str.encode(data))
reply = self.client.recv(2048).decode()
return reply
except socket.error as e:
return str(e)
Nam
Type
Size
Value
Variable Explorer
Plots
Files
Profiler
Spyder: 5.4.0
internal (Python 3.8.10)
Completions: internal
LSP: Python
Line 7, Col 29
UTF-8
CRLF
RW
Mem 83%
49°F
Mostly sunny
Search
12:24 PM
12/10/2022

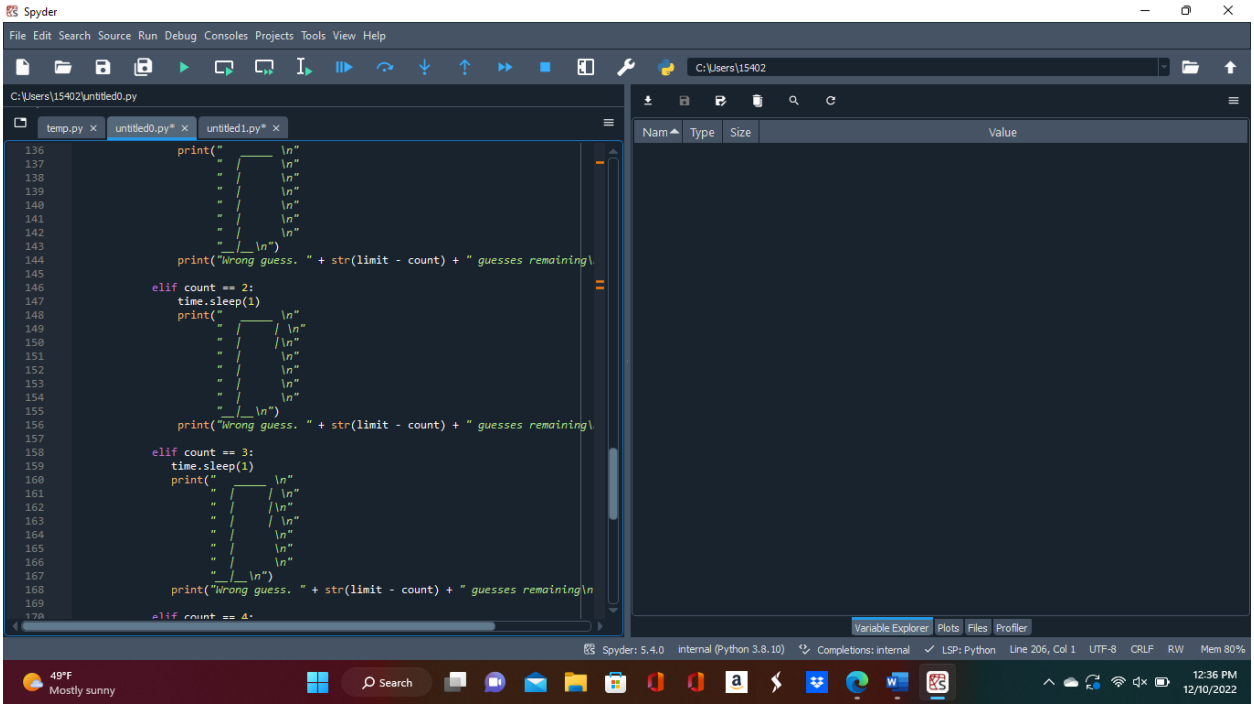

Spyder
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\15402\untitled0.py
temp.py
untitled0.py*
untitled1.py*
import socket
from _thread import *
import sys
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server = 'localhost'
port = 5555
server_ip = socket.gethostbyname(server)
try:
s.bind((server, port))
except socket.error as e:
print(str(e))
s.listen(2)
print("Waiting for a connection")
currentId = "0"
pos = ["0:50,50", "1:100,100"]
def threaded_client(conn):
global currentId, pos
conn.send(str.encode(currentId))
currentId = "1"
reply = ''
while True:
try:
data = conn.recv(2048)
reply = data.decode('utf-8')
if not data:
conn.send(str.encode("Goodbye"))
break
Nam
Type
Save data as
Value
Variable Explorer
Plots
Files
Profiler
Spyder: 5.4.0
internal (Python 3.8.10)
Completions: internal
LSP: Python
Line 206, Col 1
UTF-8
CRLF
RW
Mem 83%
49°F
Mostly sunny
Search
12:30 PM
12/10/2022


Spyder
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\15402\untitled0.py
temp.py
untitled0.py*
untitled1.py*
else:
print("Recieved: " + reply)
arr = reply.split(":")
id = int(arr[0])
pos[id] = reply
if id == 0: nid = 1
if id == 1: nid = 0
reply = pos[nid][:]
print("Sending: " + reply)
conn.sendall(str.encode(reply))
except:
break
print("Connection Closed")
conn.close()
while True:
conn, addr = s.accept()
print("Connected to: ", addr)
start_new_thread(threaded_client, (conn,))
import random
import time
import sys
print(sys.version)
# Initial Steps to invite in the game:
print("\nWelcome to Hangman game by Sahmer\n")
name = input("Enter your name: ")
print("Hello " + name + "! Best of Luck!")
time.sleep(3)
Nam
Type
Size
Value
Variable Explorer
Plots
Files
Profiler
Spyder: 5.4.0
internal (Python 3.8.10)
Completions: internal
LSP: Python
Line 206, Col 1
UTF-8
CRLF
RW
Mem 81%
49°F
Mostly sunny
Search
12:30 PM
12/10/2022

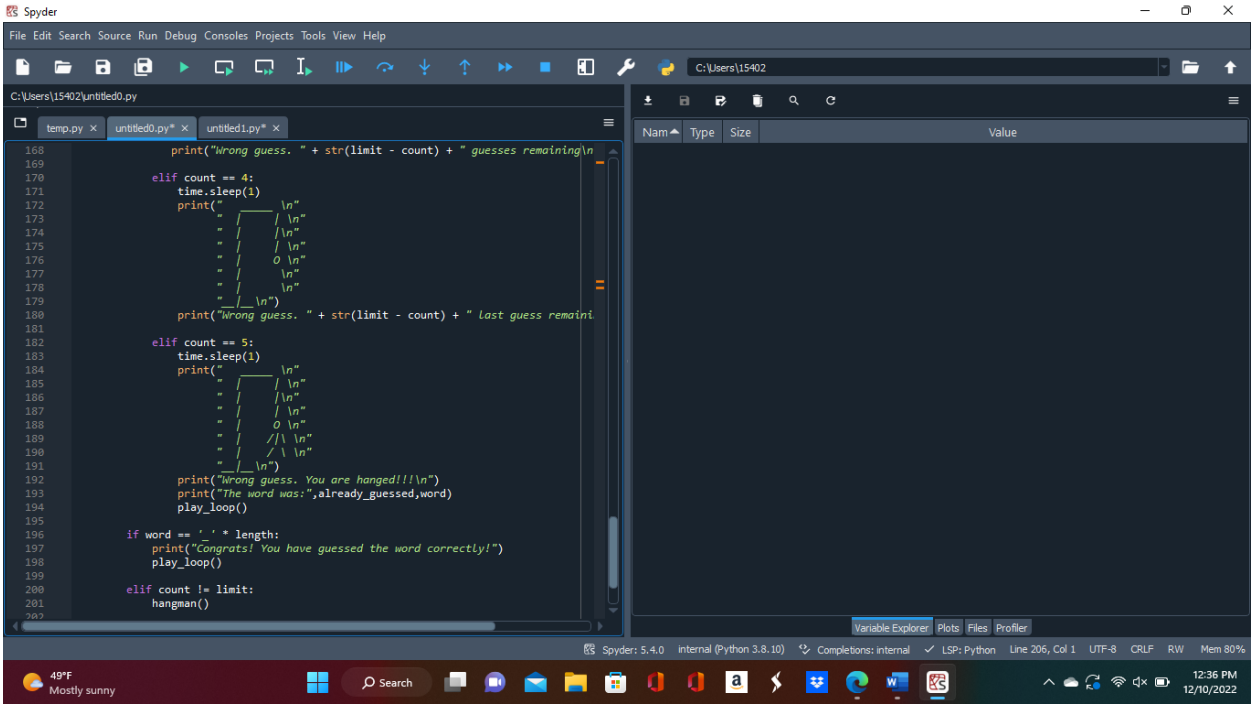

Spyder
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\15402\untitled0.py
temp.py
untitled0.py*
untitled1.py*
time.sleep(3)
# The parameters we require to execute the game:
def main():
global count
global display
global word
global already_guessed
global length
global play_game
words_to_guess = ["Cowboys","Bucs","Eagles","Dolphins","Cheifs","Ravens",
,"plants"]
r=random.randrange(0, 11)
word=words_to_guess[r]
#word = random.choices(words_to_guess)
length = len(word)
count = 0
display = '_' * length
already_guessed = []
play_game = ""
# A loop to re-execute the game when the first round ends:
def play_loop():
global play_game
play_game = input("Do You want to play again? y = yes, n = no \n")
while play_game not in ["y", "n","Y","N"]:
play_game = input("Do You want to play again? y = yes, n = no \n")
if play_game == "y":
main()
elif play_game == "n":
print("Thanks For Playing! We expect you back again!")
exit()
Nam
Type
Size
Value
Variable Explorer
Plots
Files
Profiler
Spyder: 5.4.0
internal (Python 3.8.10)
Completions: internal
LSP: Python
Line 206, Col 1
UTF-8
CRLF
RW
Mem 83%
49°F
Mostly sunny
Search
12:31 PM
12/10/2022




```
            exit()

    # Initializing all the conditions required for the game:
    def hangman():
        global count
        global display
        global word
        global already_guessed
        global play_game
        limit = 5
        guess = input("This is the Hangman Word: " + display + " Enter your guess
        guess = guess.strip()
        if len(guess.strip()) == 0 or len(guess.strip()) >= 2 or guess <= "9":
            print("Invalid Input, Try a Letter\n")
            hangman()


        elif guess in word:
            already_guessed.extend([guess])
            index = word.find(guess)
            word = word[:index] + "_" + word[index + 1:]
            display = display[:index] + guess + display[index + 1:]
            print(display + "\n")

        elif guess in already_guessed:
            print("Try another letter.\n")

        else:
            count += 1

            if count == 1:
                time.sleep(1)
                print("    _____  \n"
```




```
                print("    _____  \n"
                      "   |     \n"
                      "   |     \n"
                      "   |     \n"
                      "   |     \n"
                      "   |     \n"
                      "   |     \n"
                      "  _|__\n")
                print("Wrong guess. " + str(limit - count) + " guesses remaining\

            elif count == 2:
                time.sleep(1)
                print("    _____  \n"
                      "   |     | \n"
                      "   |     |\n"
                      "   |     \n"
                      "   |     \n"
                      "   |     \n"
                      "   |     \n"
                      "  _|__\n")
                print("Wrong guess. " + str(limit - count) + " guesses remaining\

            elif count == 3:
                time.sleep(1)
                print("    _____  \n"
                      "   |     | \n"
                      "   |     |\n"
                      "   |     | \n"
                      "   |      \n"
                      "   |     \n"
                      "   |     \n"
                      "  _|__\n")
                print("Wrong guess. " + str(limit - count) + " guesses remaining\n

            elif count == 4:
```



Meaning of screenshots

The process that occurred in the screen shots is the local host, connected with the server by located the specified IP address of the server. After they connected the server greeted the client and made sure it was the right client in which after they played the hangman game. This game contains for loops in which it will scramble the words in the list and choose A random word from

the list. This game also contains many functions, and it imports many such as random, in which helps the process of this game run. This game has the list, so the computer has A variety of different options to choose from and makes the game fun.