Marcus Sowers

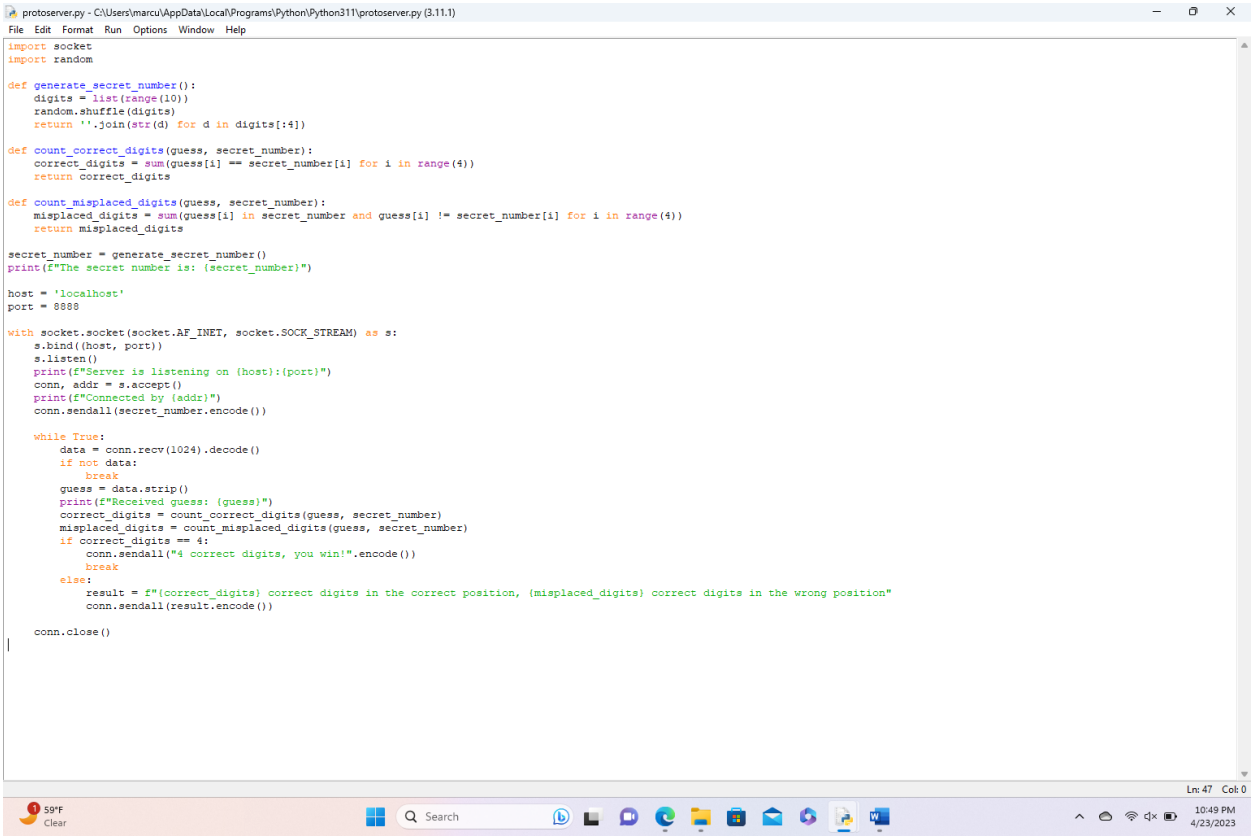
CYSE250 Cyber Programming and Network

4/19/23

Final Project Report

Server:

The program starts by defining three specific functions. The first function “generate_secret_number()” generates a random four-digit number with unique digits. The second function “count_correct_digits()” counts the number of digits in a guess that are in the correct position. The third function “count_misplaced_digits()” counts the number of digits in a guess that are in the wrong position. The program then generates a random secret number using “generate_secret_number()” and prints it to the console. The program sets the variables 'host' and 'port' to define the server's hostname or IP address and the port number the server will connect. It then creates a new socket object “s” using the AF_INET address family (IPv4) and the SOCK_STREAM socket type (TCP), and then binds the socket to the specified host and port using “s.bind((host, port))”. In then starts listening for incoming connections using “s.listen()” and prints a message to the console to indicate that the server is now listening. The program accepts a connection from a client using “conn, addr = s.accept()” and prints a message to the console to indicate that a client has connected. The program sends the secret number to the client using 'conn.sendall(secret_number.encode())'. Once connected, the program enters a loop where it receives guesses from the client using “conn.recv(1024).decode()”. It then processes the guess using “count_correct_digits()” and “count_misplaced_digits()” and sends a response back to the client indicating how many digits are correct and in the correct position, and how many are correct but in the wrong position. The loop will continue until the person successfully guesses the correct secret number, at which point the server sends a message to the client indicating that they have won, and the loop breaks.



Client:

In the client code, the first step was to import the socket module, which provides low-level access to the networking interfaces in Python. The code used sets the variables 'host' and 'port' to define the server's hostname or IP address, and the port number the server is listening to. The 'with' statement used opens a new socket using the AF_INET address family (IPv4) and the SOCK_STREAM socket type (TCP). This creates a new socket object 's' which is used to communicate with the server. The “s.connect((host, port))” method is used to establish a connection to the server at the specified host and port. It then receives a message from the server using the “s.recv()” method. It receives a secret number that will need to be guessed correctly. The client then enters a loop, where it repeatedly asks the user to enter a four-digit number with unique digits. It then sends this number to the server using “s.sendall()”. The server processes the guess and sends a response back to the client, which is printed to the console. The loop used continues until the client successfully guesses the correct secret number, then a message indicating that the game has been won will pop up.

protoclient.py - C:\Users\marcu\AppData\Local\Programs\Python\Python311\protoclient.py (3.11.1)

File Edit Format Run Options Window Help

```python
import socket

host = 'localhost'
port = 8888

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((host, port))
    print("Connected to the server")

    secret_number = s.recv(1024).decode()
    print(f"The secret number is: ****")

    while True:
        guess = input("Enter your guess (4 digits with unique digits): ")
        if len(guess) != 4 or len(set(guess)) != 4 or not guess.isdigit():
            print("Invalid guess, try again")
            continue
        s.sendall(guess.encode())
        result = s.recv(1024).decode()
        print(result)
```

Ln: 22 Col: 0