

APP PARALLEL AND CONCURRENT PROGRAMMING

Question 1: Create Simple Client Server Application using TCP Socket where server issue a command which will be executed at the client side as a process of remote command execution

Solution:

Client.py

```
import socket

def client_program():
    host = socket.gethostname()
    port = 100

    client_socket = socket.socket()
    client_socket.connect((host, port))

    message = input("Enter message for server: ")

    while message.lower().strip() != 'bye':
        client_socket.send(message.encode())
        data = client_socket.recv(1024).decode()

        print('Message from server: ' + data)

        message = input(" Enter message for server: ")

    client_socket.close()

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

Server.py

```
import socket
```

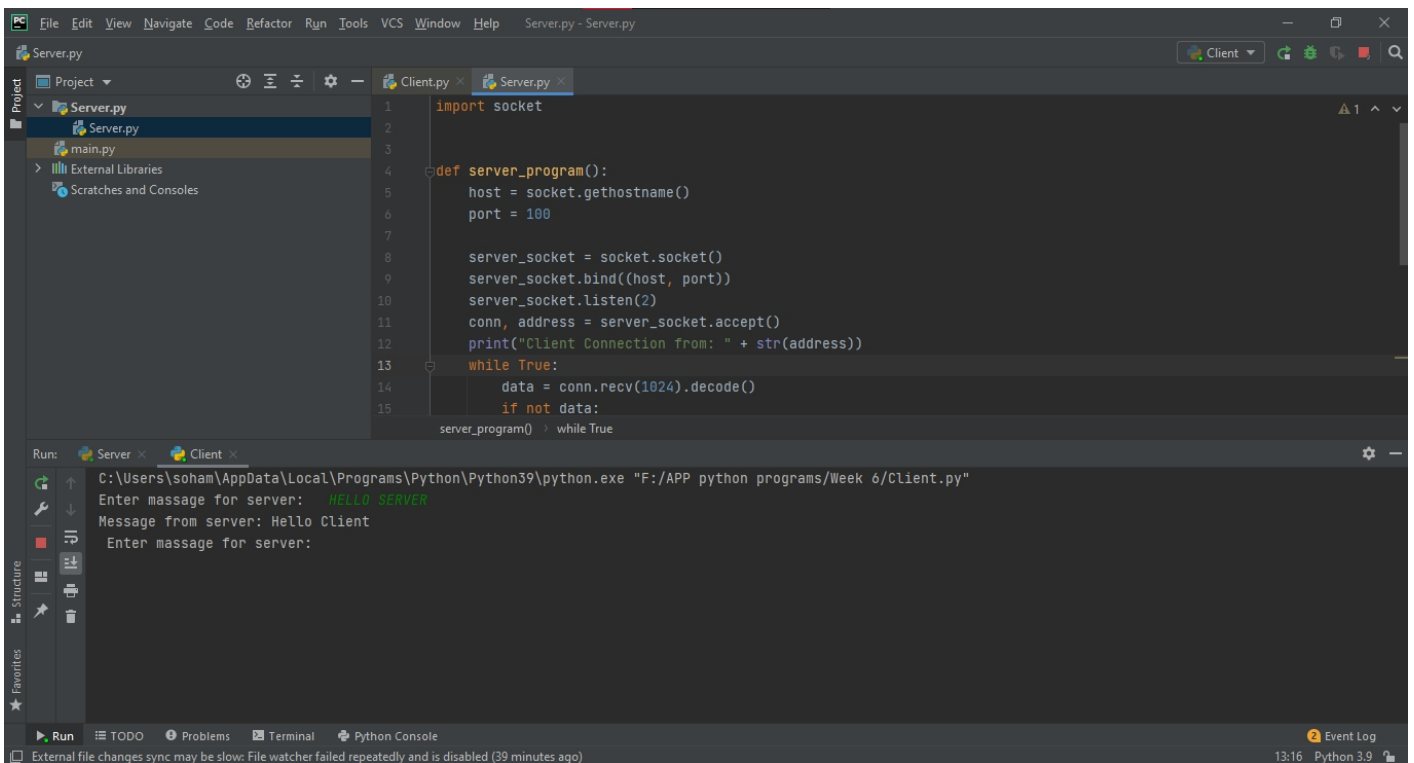
```
def server_program():
    host = socket.gethostname()
    port = 100

    server_socket = socket.socket()
    server_socket.bind((host, port))
    server_socket.listen(2)
    conn, address = server_socket.accept()
    print("Client Connection from: " + str(address))
    while True:
        data = conn.recv(1024).decode()
        if not data:
            break
        print("Message from client: " + str(data))
        data = input(' Enter message for client: ')
        conn.send(data.encode())

    conn.close()

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

SCREENSHOTS:

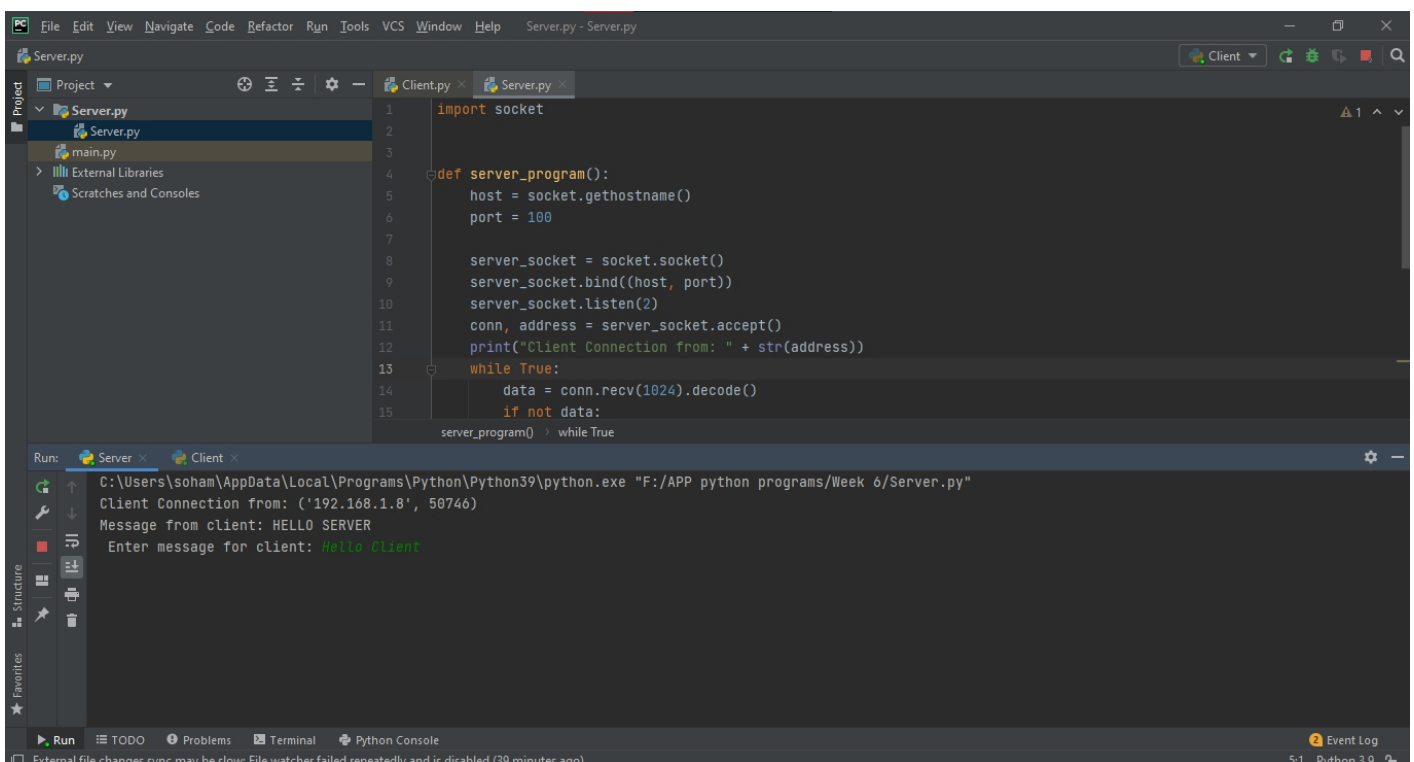


The screenshot shows an IDE window titled "Server.py - Server.py". The left sidebar displays a project structure with "Server.py" and "main.py" under a "Server.py" folder. The main editor area shows the code for "Server.py":

```
1 import socket
2
3
4 def server_program():
5     host = socket.gethostname()
6     port = 100
7
8     server_socket = socket.socket()
9     server_socket.bind((host, port))
10    server_socket.listen(2)
11    conn, address = server_socket.accept()
12    print("Client Connection from: " + str(address))
13    while True:
14        data = conn.recv(1024).decode()
15        if not data:
16            break
17    server_socket.close()
18
19 if __name__ == '__main__':
20    server_program()
```

The bottom panel shows the "Run" output for "Server.py":

```
C:\Users\soham\AppData\Local\Programs\Python\Python39\python.exe "F:/APP python programs/Week 6/Server.py"
Enter message for server: HELLO SERVER
Message from server: Hello Client
Enter message for server:
```



The screenshot shows an IDE window titled "Server.py - Server.py". The left sidebar displays a project structure with "Server.py" and "main.py" under a "Server.py" folder. The main editor area shows the code for "Client.py":

```
1 import socket
2
3
4 def client_program():
5     host = socket.gethostname()
6     port = 100
7
8     client_socket = socket.socket()
9     client_socket.connect((host, port))
10    data = client_socket.recv(1024).decode()
11    print(data)
12
13 if __name__ == '__main__':
14    client_program()
```

The bottom panel shows the "Run" output for "Client.py":

```
C:\Users\soham\AppData\Local\Programs\Python\Python39\python.exe "F:/APP python programs/Week 6/Client.py"
Client Connection from: ('192.168.1.8', 50746)
Message from client: HELLO SERVER
Enter message for client: Hello Client
```

Done by: Soham Sakaria
RA1911003010246