Sayali Dilip Deshmukh **1001966628**

Assignment No: 02

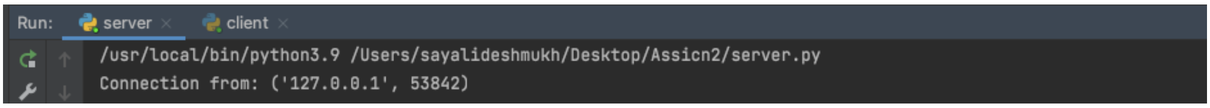
Objective: An introduction to socket programming: Client-server architecture

Description: Using client-server architecture and caching, this assignment introduces the socket programming.
Create a continuously running client-server-based application (along with proxy server) that retrieves key-value pairs using socket programming. The server facilitates service for commands like GET, PUT and DUMP, for the corresponding multiple pair (of key-value) stored. To achieve this, any programming language and any IDE can be used.

Commands:
Run server file.
Run client file.
Perform GET, PUT, DUMP commands

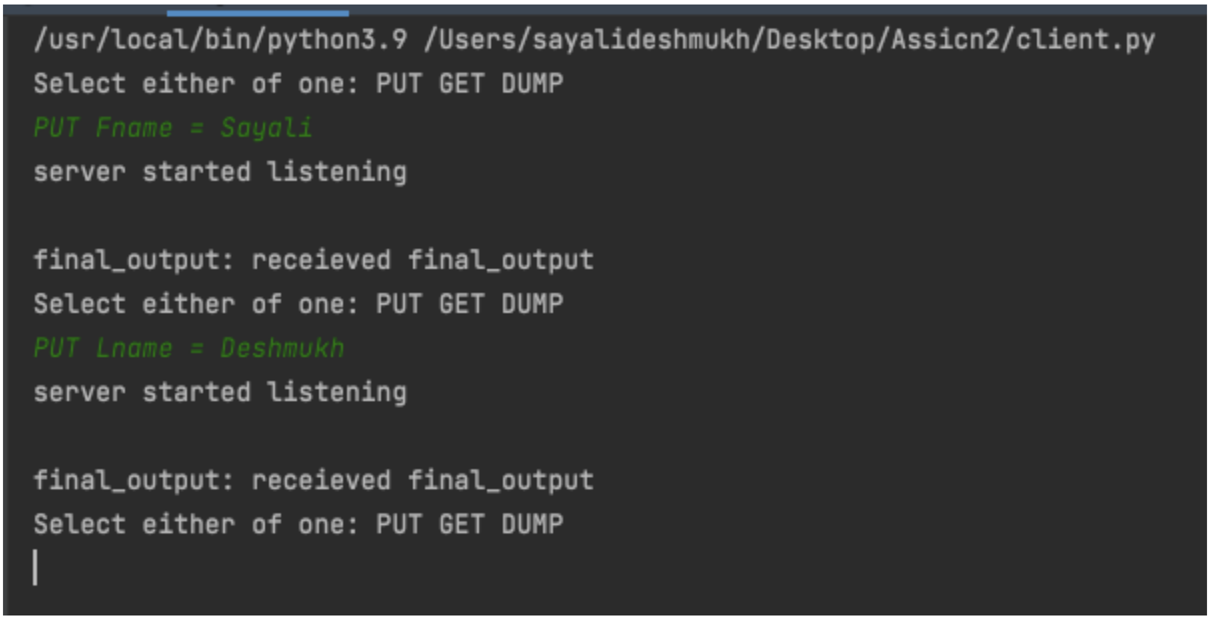
Screenshots:

Connection Establishment:



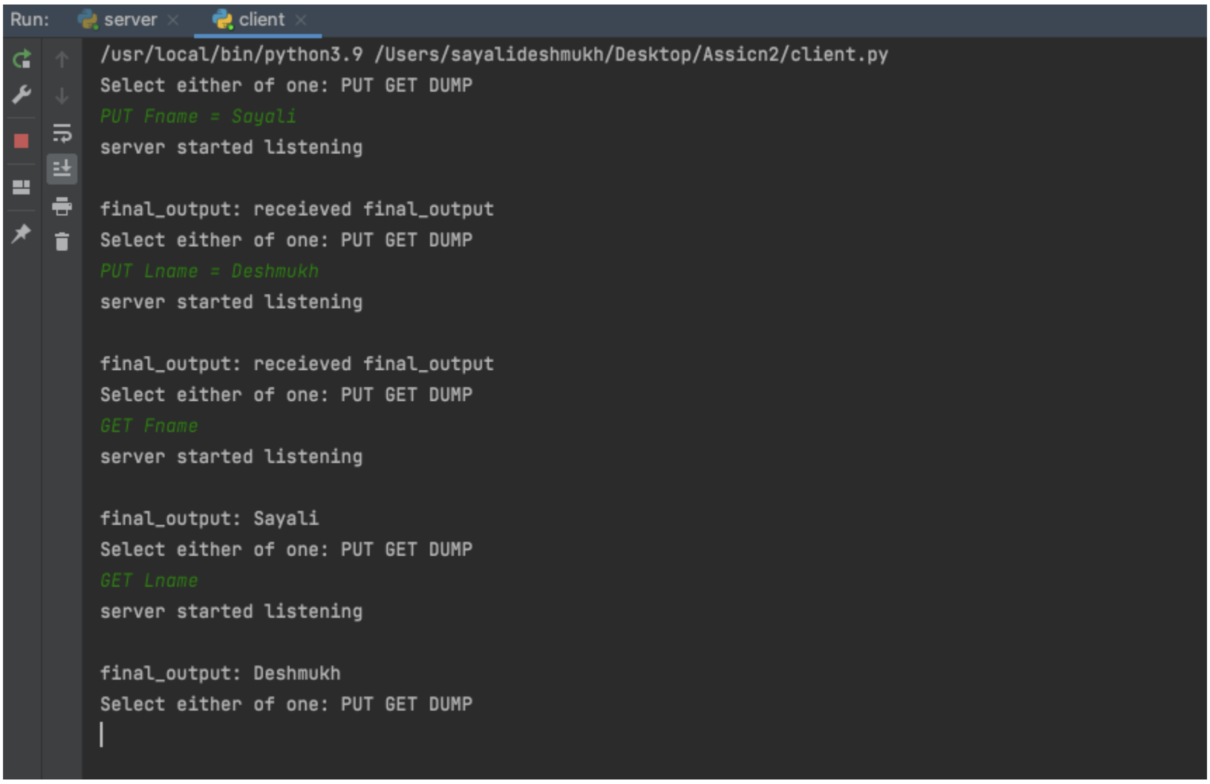
```
/usr/local/bin/python3.9 /Users/sayalideshmukh/Desktop/Assicn2/server.py
Connection from: ('127.0.0.1', 53842)
```

1.Implement PUT command on the server:



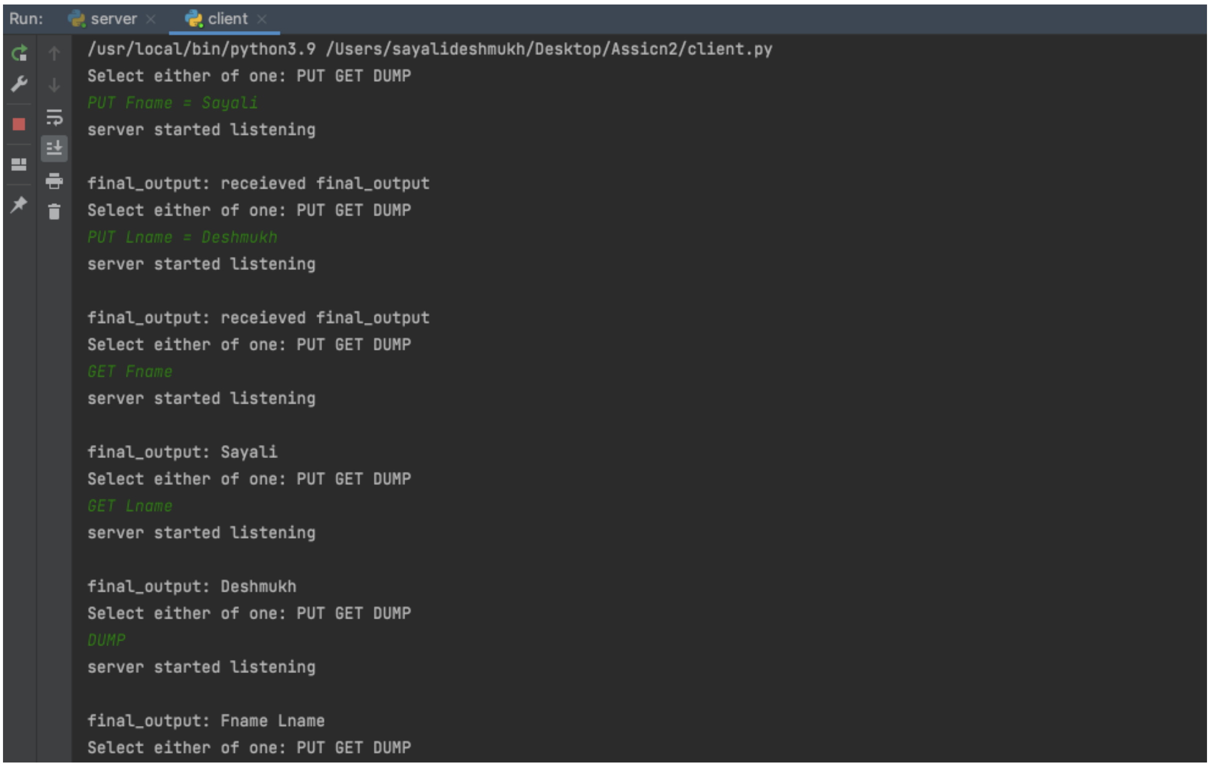
```
/usr/local/bin/python3.9 /Users/sayalideshmukh/Desktop/Assicn2/client.py
Select either of one: PUT GET DUMP
PUT Fname = Sayali
server started listening

final_output: receieved final_output
Select either of one: PUT GET DUMP
PUT Lname = Deshmukh
server started listening

final_output: receieved final_output
Select either of one: PUT GET DUMP
```

2. Implement GET command on the server:



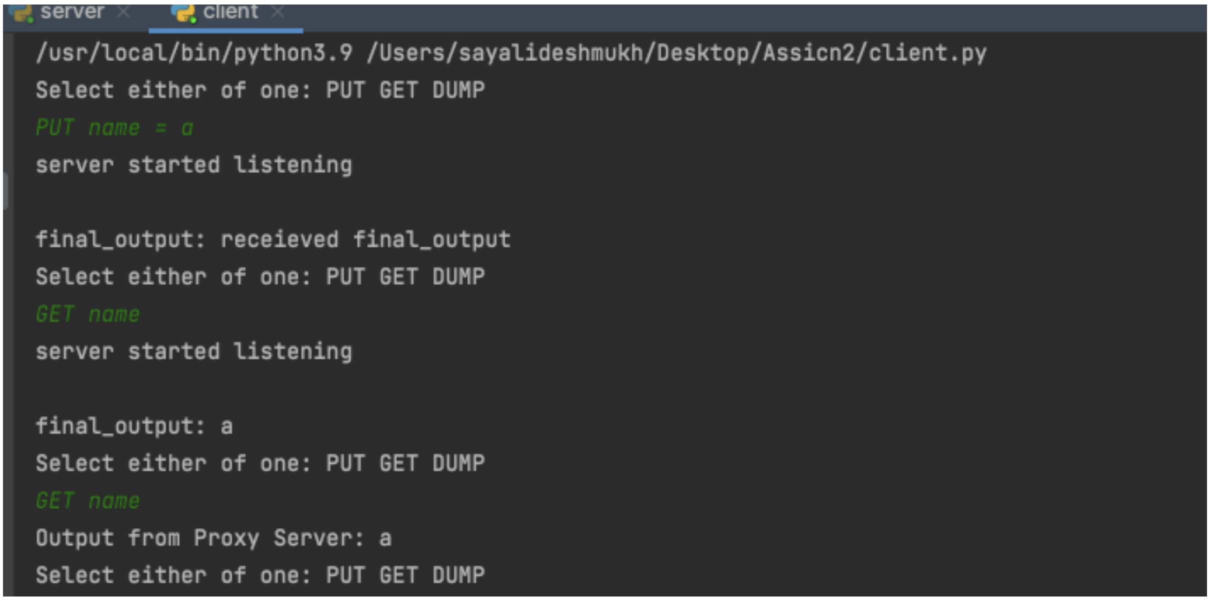
```
/usr/local/bin/python3.9 /Users/sayalideshmukh/Desktop/Assicn2/client.py
Select either of one: PUT GET DUMP
PUT Fname = Sayali
server started listening

final_output: receieved final_output
Select either of one: PUT GET DUMP
PUT Lname = Deshmukh
server started listening

final_output: receieved final_output
Select either of one: PUT GET DUMP
GET Fname
server started listening

final_output: Sayali
Select either of one: PUT GET DUMP
GET Lname
server started listening

final_output: Deshmukh
Select either of one: PUT GET DUMP
```

3. Implement DUMP command on the server:



```
/usr/local/bin/python3.9 /Users/sayalideshmukh/Desktop/Assicn2/client.py
Select either of one: PUT GET DUMP
PUT Fname = Sayali
server started listening

final_output: receieved final_output
Select either of one: PUT GET DUMP
PUT Lname = Deshmukh
server started listening

final_output: receieved final_output
Select either of one: PUT GET DUMP
GET Fname
server started listening

final_output: Sayali
Select either of one: PUT GET DUMP
GET Lname
server started listening

final_output: Deshmukh
Select either of one: PUT GET DUMP
DUMP
server started listening

final_output: Fname Lname
Select either of one: PUT GET DUMP
```

4. Implement proxy server (for caching) that returns the value of GET command if has been previously requested by the client:



```
/usr/local/bin/python3.9 /Users/sayalideshmukh/Desktop/Assicn2/client.py
Select either of one: PUT GET DUMP
PUT name = a
server started listening

final_output: receieved final_output
Select either of one: PUT GET DUMP
GET name
server started listening

final_output: a
Select either of one: PUT GET DUMP
GET name
Output from Proxy Server: a
Select either of one: PUT GET DUMP
```

Conclusion: By using Socket Programming, we have implemented client-server architecture and performed GET, PUT, DUMP commands also Implemented proxy server