

Name: - Meet Desai
UNI: - mpd2155

Compiling:-

The code can be directly compiled through the command line/terminal, these are the commands

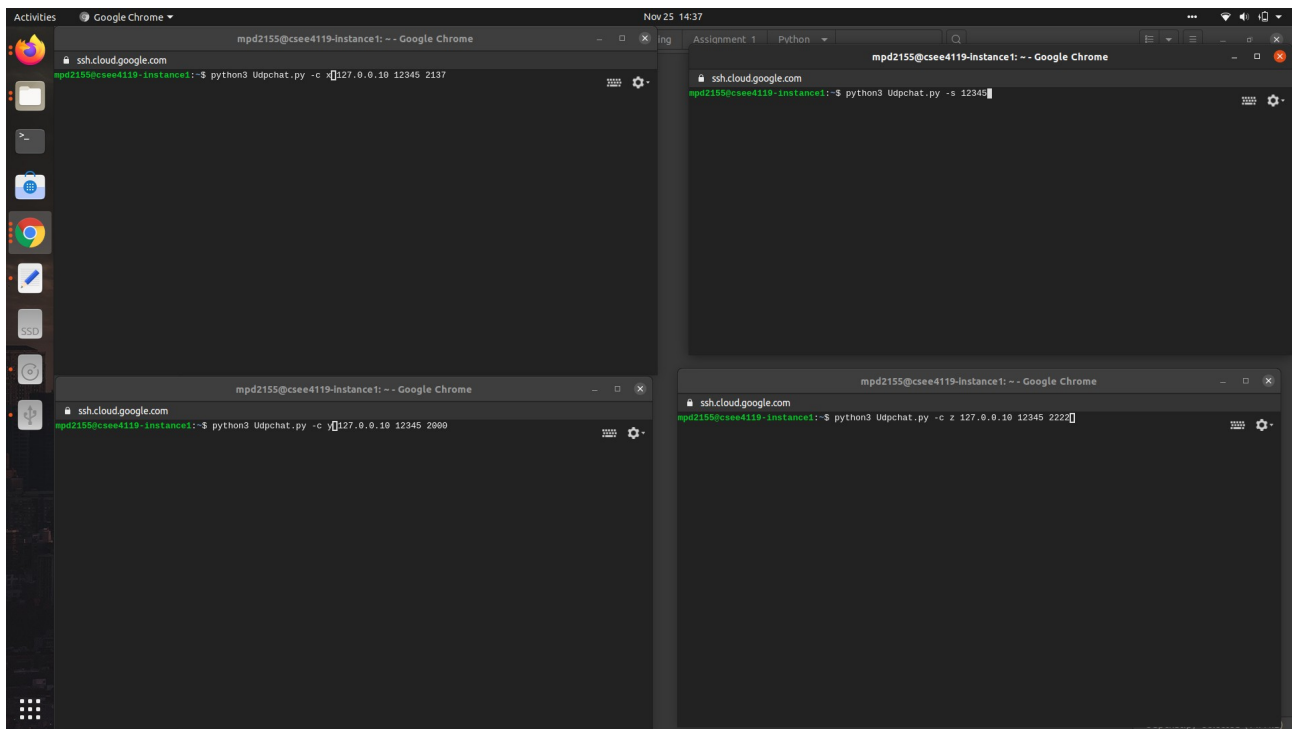
Server:-

```
python3 Udpchat.py -s 12345
```

Client:-

```
python3 Udpchat.py -c <nickname> 127.0.0.10 12345 1232
```

Below is an image showing the same



Code:-

I have written the code such that it reads the command line input for -s or -c and then calls the function server or client respectively. If there is wrong input in command line then program exits itself.

Server:-

If server is called the code checks for a port number on the command line and binds the server to that port number. Then the code checks for 3 different commands, which will be explained later. If none of those three commands are found then it takes the data sent by the client and adds it to the local client table, which is a list, and sends the table to all the active clients. The following three commands are checked: -

dereg : - if there is a dereg request, the server updates the client table and it changes the client who requested the dereg to offline in its local table and then sends the updated table to every online client.

reg : - if there is a reg request, the server updates the client table and it changes the client who requested the reg to online in its local table, then checks if it has any saved messages, which is a list, it sends the messages to the client, after which it deletes the messages from the saved messages list after which it sends the updated table to every online client.

saved_m : - It also checks if the receiver is online and sends the sender a message that the receiver is online and an updated table to the sender. If a client sends a saved message request, it saves the port number of sender client, nick name of receiver client, the message to be sent, and the time stamp in a list.

The server prints out the data table everytime it gets updates and it also prints if it receives a saved message request and it prints when saves a message.

Client : -

If the client is called it takes the nickname, server ip, server port, client port from the command line. It binds the port to the cport input. The code first sends some data to the server and to register itself. The server sends a message that says you're registered and then sends a client table. The client stores the table locally. The client then starts a select procedure so as to listen to the incoming data to the port and take input from the user. The client then listens for the following commands : -

send : - the send command sends the client the message the sender wants to send and then waits for an ack if an ack is received it prints that the message has been received otherwise it tries to send the message to the server to save. If the server doesnot respond with an ACK the client exits the program. The client also checks if the receiver is online in its local table of clients in case the receiver is offline it directly sends the message to the server to be saved.

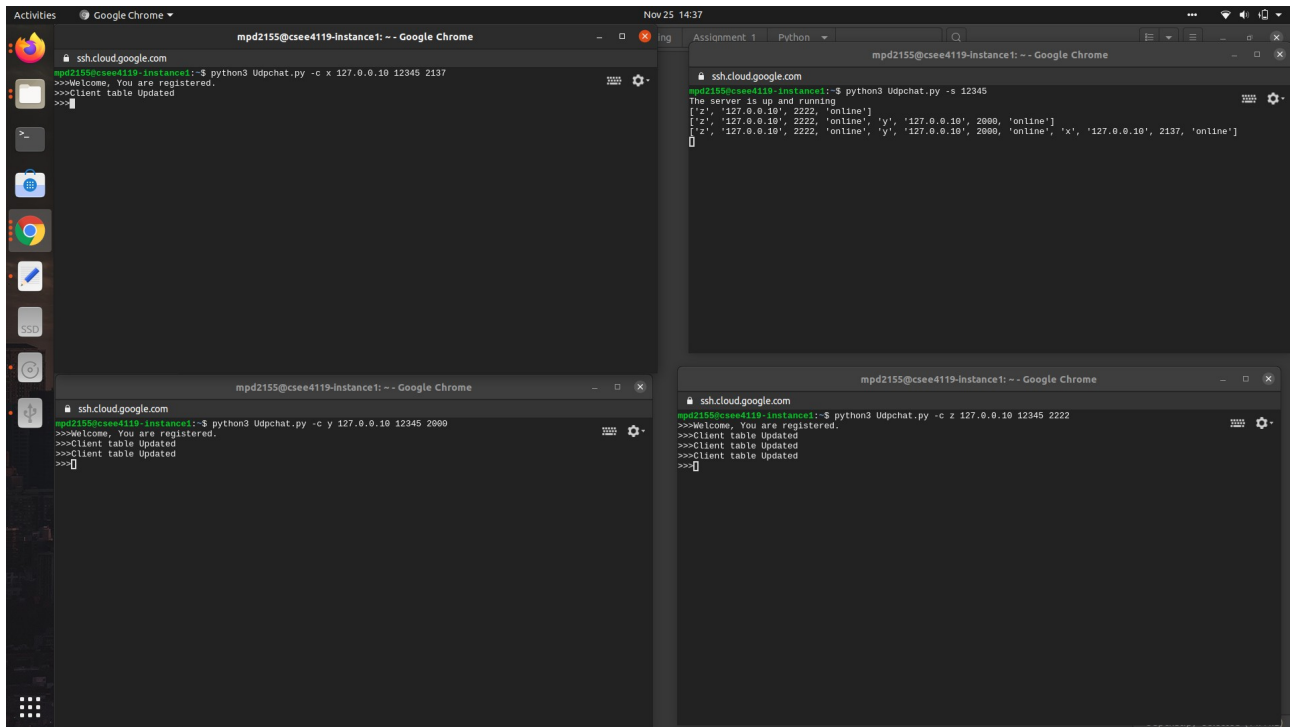
dereg : - the client sends the server dereg and then waits for the server to respond. It tries 5 times and if the server is unresponsive then it directly exits the program. In case the server sends an ack back it closes the UDP connection and frees the port number it was using. After the user can use reg command to register again to the server. The client cannot dereg or reg any other client then itself.

Any other input is treated as garbage and just prints "Wrong input" to the terminal.

I have implemented the code such that whenever there is a keyboard interrupt(ctrl +c) the client exits gracefully and also sends server a dereg request telling the server that it is going offline. The server also exits gracefully whenever there is a keyboard interrupt. I have assumed that the clients do not take nick names like reg, dereg and saved_m and my code will exit if the client tries to take any of these nicknames. My code also exits if it detects that two different client wants the same nickname. The client that registers later will not be able to take the nickname. Everything is my code is case-sensitive including commands like reg, dereg, and send. If a client is typing something and a new message comes in, the message is displayed in the same line.

Below are some screenshots of the working of my code : -

Initial screenshot of 3 clients and 1 server



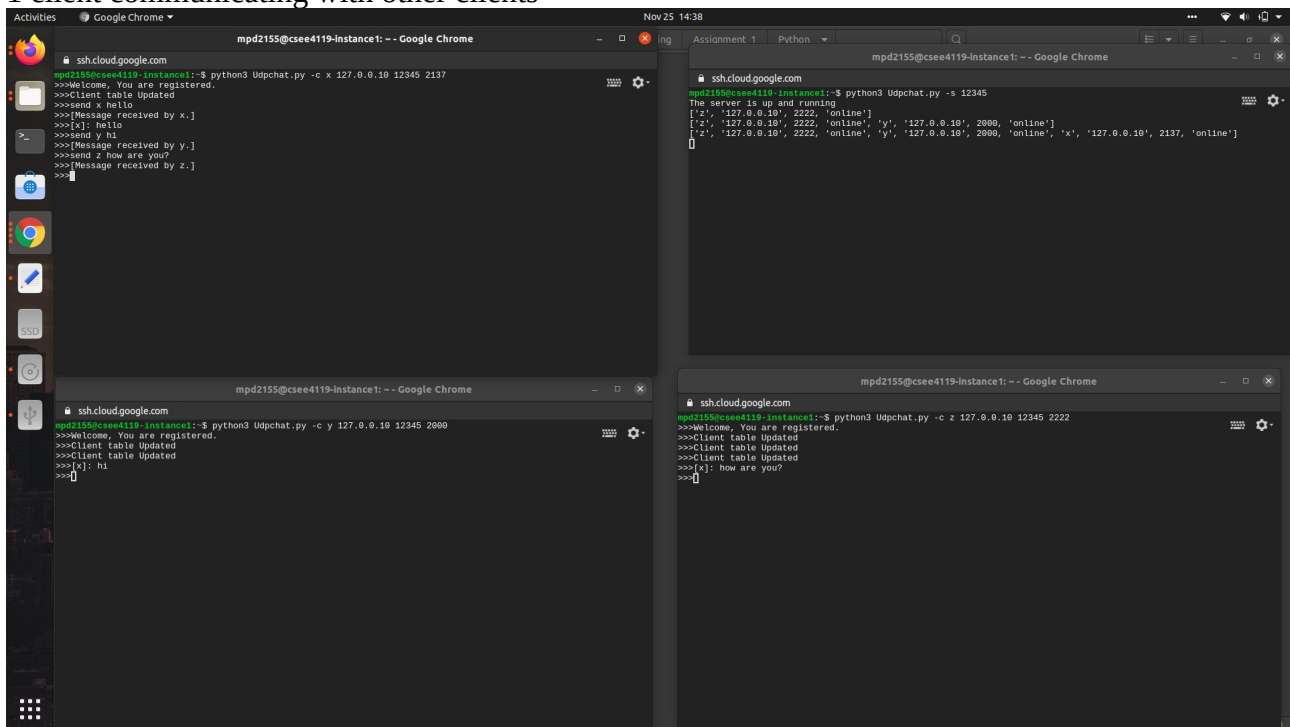
```
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c x 127.0.0.10 12345 2137
>>>Welcome, You are registered.
>>>Client table Updated
>>>
```

```
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -s 12345
The server is up and running
[2, '127.0.0.10', 2222, 'online']
[2, '127.0.0.10', 2222, 'online', 'y', '127.0.0.10', 2000, 'online']
[2, '127.0.0.10', 2222, 'online', 'y', '127.0.0.10', 2000, 'online', 'x', '127.0.0.10', 2137, 'online']
```

```
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c y 127.0.0.10 12345 2000
>>>Welcome, You are registered.
>>>Client table Updated
>>>Client table Updated
>>>
```

```
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -s 12345
The server is up and running
[2, '127.0.0.10', 2222, 'online']
[2, '127.0.0.10', 2222, 'online', 'y', '127.0.0.10', 2000, 'online']
[2, '127.0.0.10', 2222, 'online', 'y', '127.0.0.10', 2000, 'online', 'x', '127.0.0.10', 2137, 'online']
```

1 client communicating with other clients



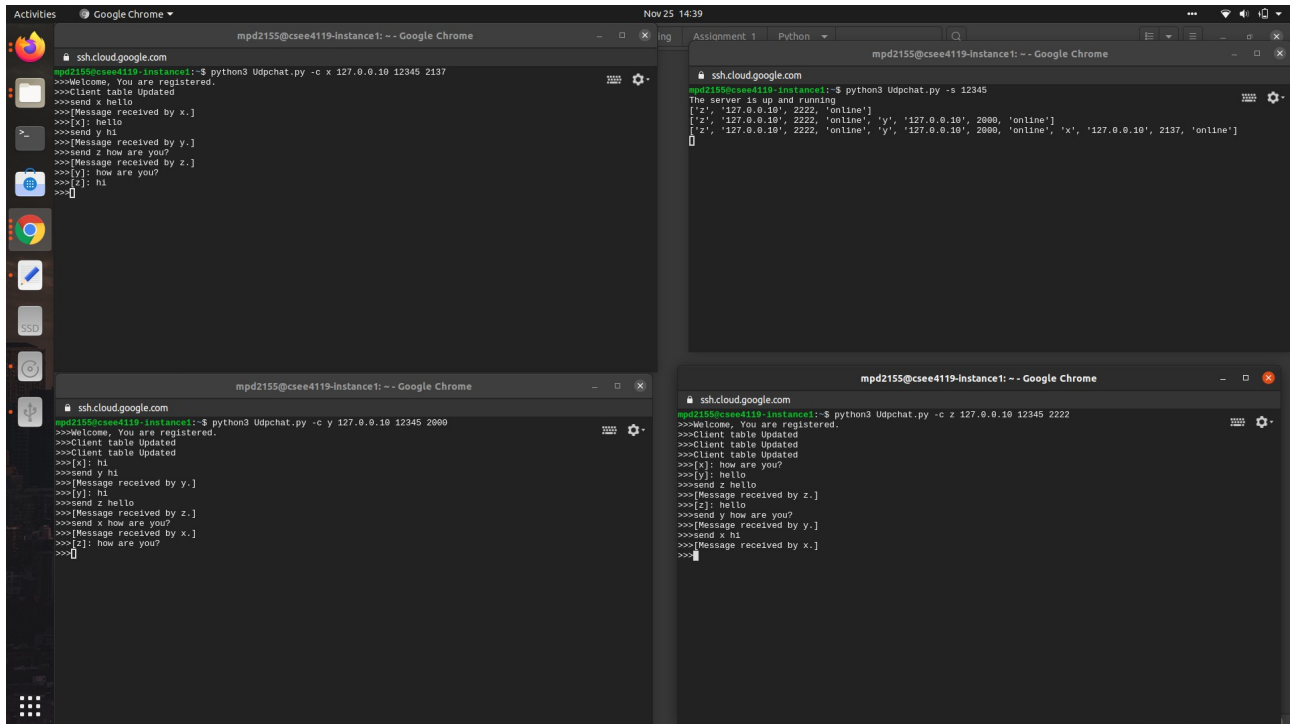
```
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c x 127.0.0.10 12345 2137
>>>Welcome, You are registered.
>>>Client table Updated
>>>send x hello
>>>[Message received by x.]
>>>[x]: hello
>>>send y hi
>>>[Message received by y.]
>>>send z how are you?
>>>[Message received by z.]
>>>
```

```
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -s 12345
The server is up and running
[2, '127.0.0.10', 2222, 'online']
[2, '127.0.0.10', 2222, 'online', 'y', '127.0.0.10', 2000, 'online']
[2, '127.0.0.10', 2222, 'online', 'y', '127.0.0.10', 2000, 'online', 'x', '127.0.0.10', 2137, 'online']
```

```
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c y 127.0.0.10 12345 2000
>>>Welcome, You are registered.
>>>Client table Updated
>>>Client table Updated
>>>[x]: hi
>>>
```

```
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -s 12345
The server is up and running
[2, '127.0.0.10', 2222, 'online']
[2, '127.0.0.10', 2222, 'online', 'y', '127.0.0.10', 2000, 'online']
[2, '127.0.0.10', 2222, 'online', 'y', '127.0.0.10', 2000, 'online', 'x', '127.0.0.10', 2137, 'online']
```

All 3 clients communicating with each other



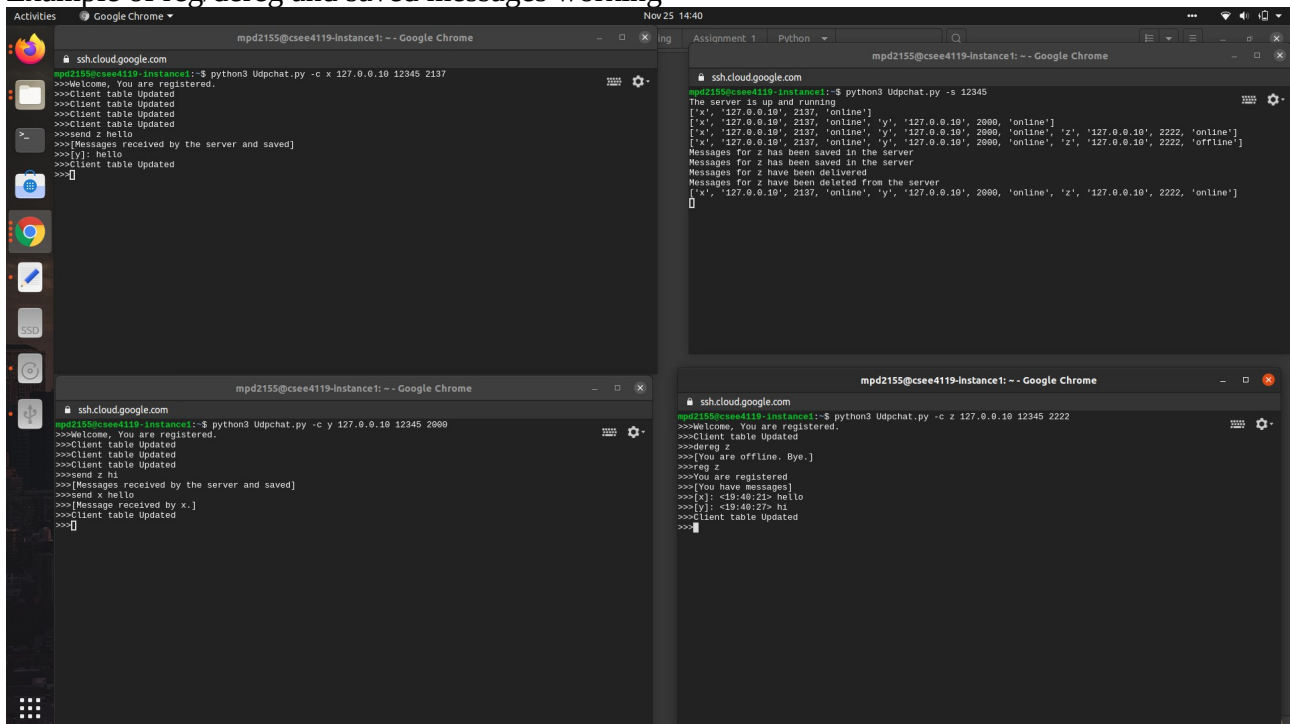
```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c x 127.0.0.10 12345 2137
>>>Welcome, You are registered.
>>>Client table Updated
>>>send x hello
>>>[Message received by x.]
>>>[x]: hello
>>>send y hi
>>>[Message received by y.]
>>>send z how are you?
>>>[Message received by z.]
>>>[y]: how are you?
>>>[x]: hi
>>>
```

```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -s 12345
The server is up and running
[{'x': '127.0.0.10', '2222', 'online'}]
[{'z': '127.0.0.10', '2222', 'online'}, {'y': '127.0.0.10', '2000', 'online'}]
[{'x': '127.0.0.10', '2222', 'online'}, {'y': '127.0.0.10', '2000', 'online'}, {'x': '127.0.0.10', '2137', 'online'}]
[]
```

```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c y 127.0.0.10 12345 2000
>>>Welcome, You are registered.
>>>Client table Updated
>>>Client table Updated
>>>[x]: hi
>>>send y hi
>>>[Message received by y.]
>>>[y]: hi
>>>send z hello
>>>[Message received by z.]
>>>send x how are you?
>>>[Message received by x.]
>>>[x]: how are you?
>>>
```

```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c z 127.0.0.10 12345 2222
>>>Welcome, You are registered.
>>>Client table Updated
>>>Client table Updated
>>>[x]: how are you?
>>>[y]: hello
>>>send z hello
>>>[Message received by z.]
>>>[z]: hello
>>>send y how are you?
>>>[Message received by y.]
>>>send x hi
>>>[Message received by x.]
>>>
```

Example of reg/dereg and saved messages working



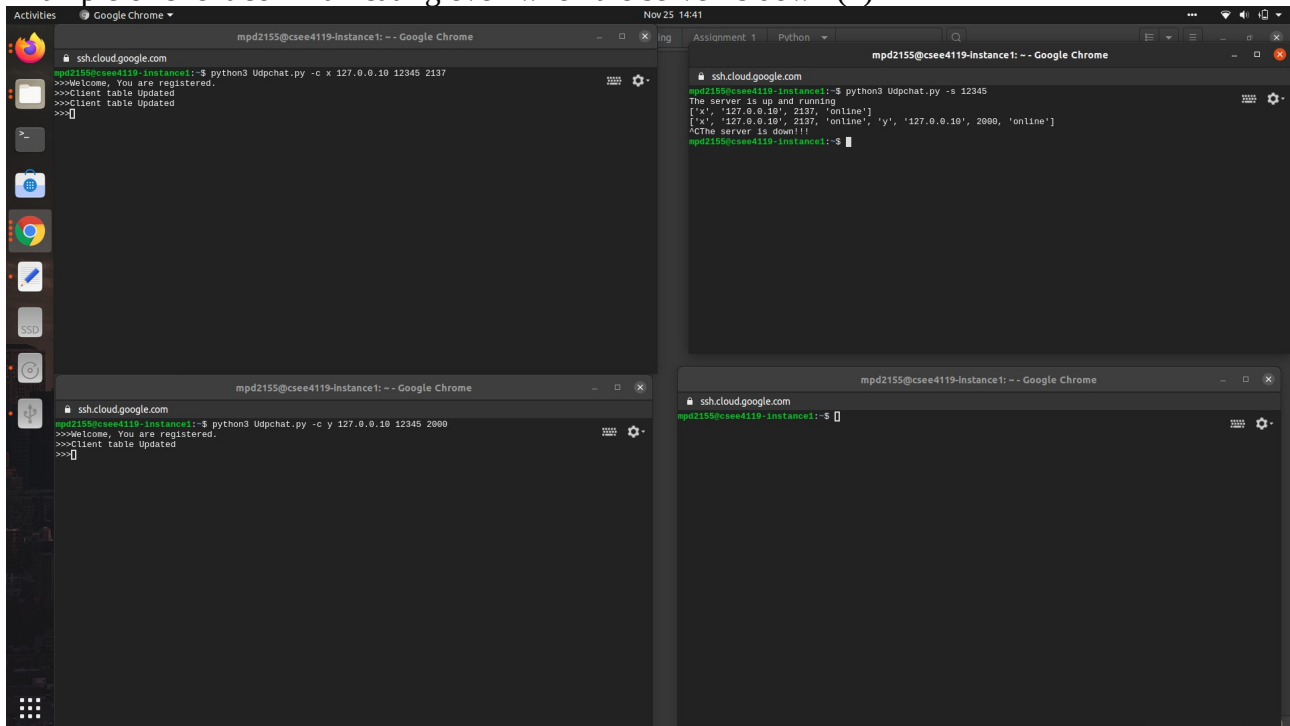
```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c x 127.0.0.10 12345 2137
>>>Welcome, You are registered.
>>>Client table Updated
>>>Client table Updated
>>>Client table Updated
>>>send z hello
>>>[Messages received by the server and saved]
>>>[y]: hello
>>>Client table Updated
>>>
```

```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -s 12345
The server is up and running
[{'x': '127.0.0.10', '2137', 'online'}]
[{'x': '127.0.0.10', '2137', 'online'}, {'y': '127.0.0.10', '2000', 'online'}]
[{'x': '127.0.0.10', '2137', 'online'}, {'y': '127.0.0.10', '2000', 'online'}, {'z': '127.0.0.10', '2222', 'online'}]
[{'x': '127.0.0.10', '2137', 'online'}, {'y': '127.0.0.10', '2000', 'online'}, {'z': '127.0.0.10', '2222', 'offline'}]
Messages for z has been saved in the server
Messages for z has been delivered
Messages for z have been deleted from the server
[{'x': '127.0.0.10', '2137', 'online'}, {'y': '127.0.0.10', '2000', 'online'}, {'z': '127.0.0.10', '2222', 'online'}]
[]
```

```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c y 127.0.0.10 12345 2000
>>>Welcome, You are registered.
>>>Client table Updated
>>>Client table Updated
>>>Client table Updated
>>>send z hi
>>>[Messages received by the server and saved]
>>>send x hello
>>>[Message received by x.]
>>>Client table Updated
>>>
```

```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c z 127.0.0.10 12345 2222
>>>Welcome, You are registered.
>>>Client table Updated
>>>dereg z
>>>[You are offline. Bye.]
>>>dereg z
>>>[You are registered]
>>>[You have message:]
>>>[x]: <19:40:21> hello
>>>[y]: <19:40:21> hi
>>>Client table Updated
>>>
```

Example of client communicating even when the server is down (1)

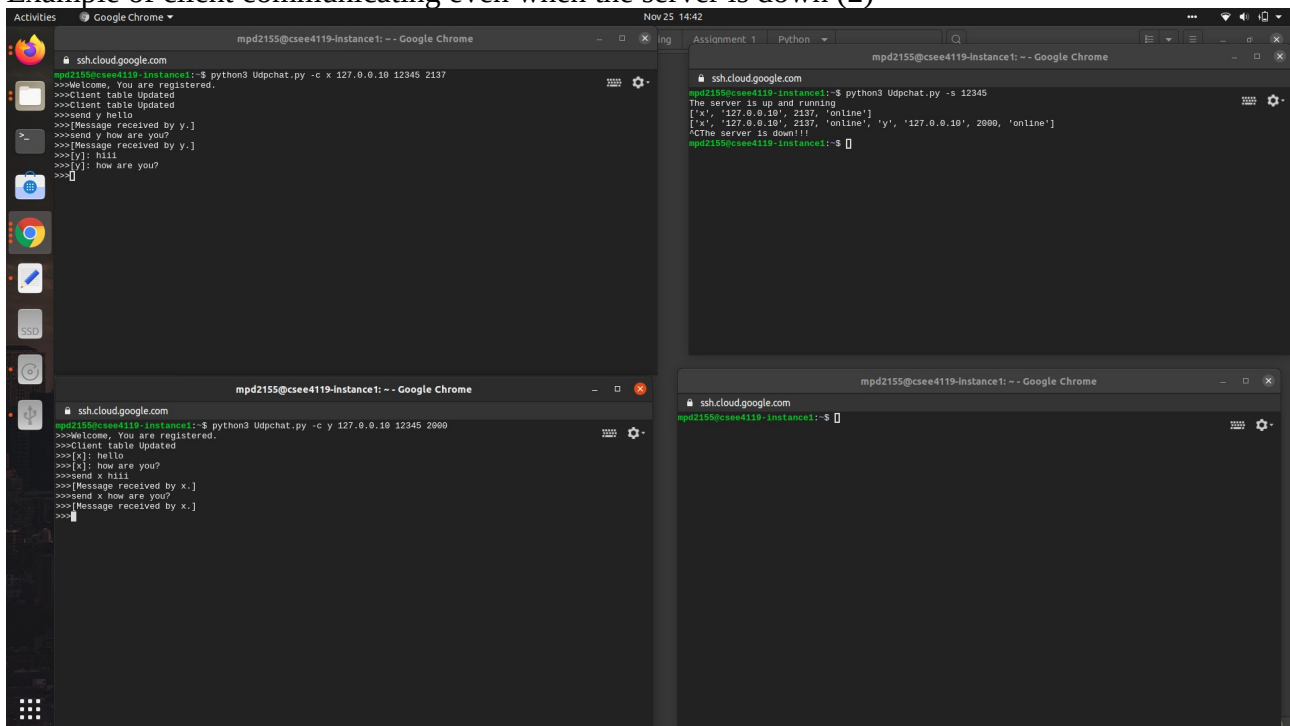


The screenshot shows a terminal window with the following content:

```
mpd2155@csee4119-instance1: ~ - Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c x 127.0.0.10 12345 2137
>>>Welcome, You are registered.
>>>Client table Updated
>>>Client table Updated
>>>
```

The terminal window is titled "mpd2155@csee4119-instance1: ~ - Google Chrome". The prompt is "mpd2155@csee4119-instance1:~\$". The user has run the command "python3 Udpchat.py -c x 127.0.0.10 12345 2137". The output shows a welcome message, a client table update, and a client table update. The terminal window is also titled "ssh.cloud.google.com".

Example of client communicating even when the server is down (2)

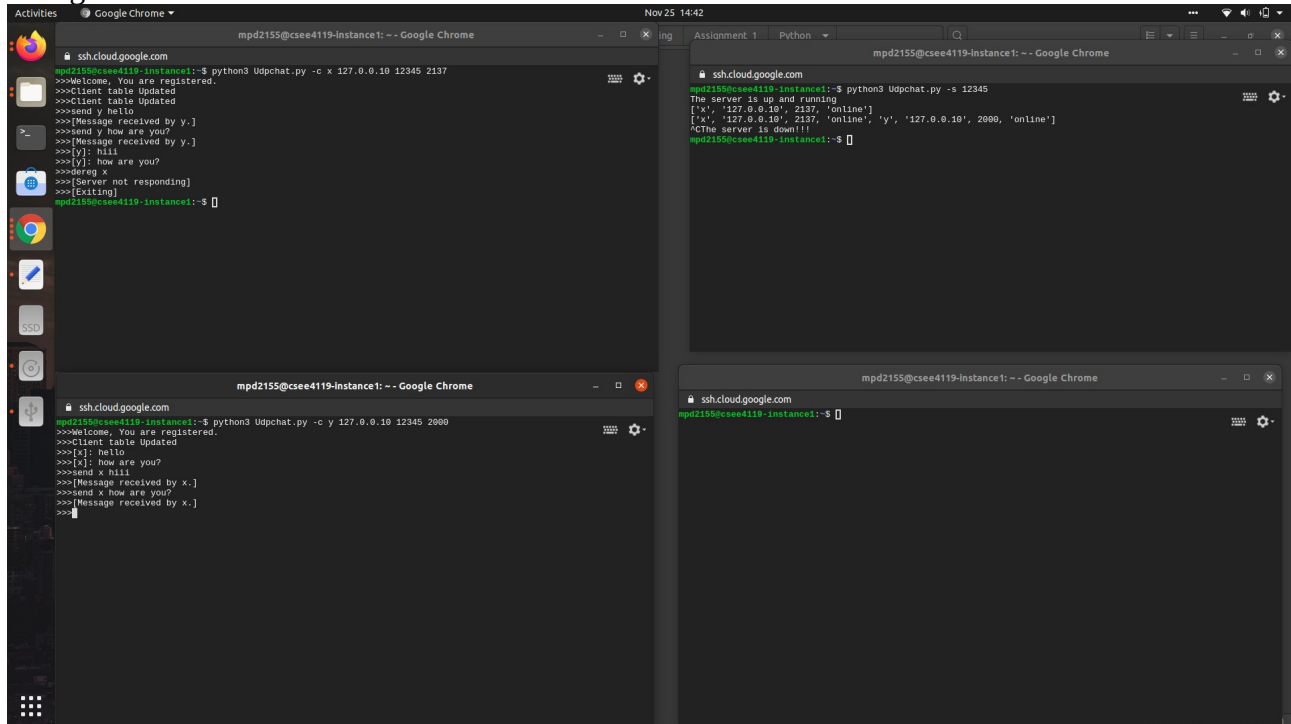


The screenshot shows a terminal window with the following content:

```
mpd2155@csee4119-instance1: ~ - Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c x 127.0.0.10 12345 2137
>>>Welcome, You are registered.
>>>Client table Updated
>>>Client table Updated
>>>send y hello
>>>[Message received by y.]
>>>send y how are you?
>>>[Message received by y.]
>>>[y]: hi!!
>>>[y]: how are you?
>>>
```

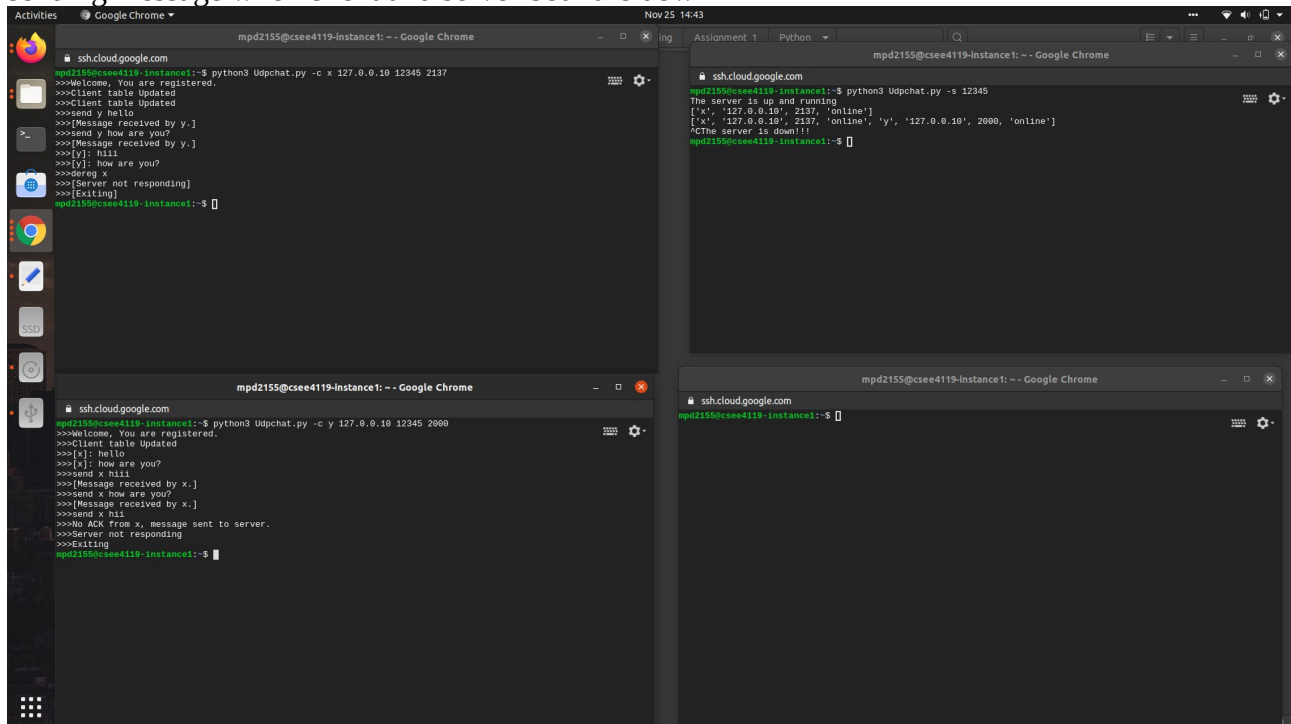
The terminal window is titled "mpd2155@csee4119-instance1: ~ - Google Chrome". The prompt is "mpd2155@csee4119-instance1:~\$". The user has run the command "python3 Udpchat.py -c x 127.0.0.10 12345 2137". The output shows a welcome message, a client table update, a client table update, and a client table update. The user has also sent a message "y hello" and received a response "[Message received by y.]". The user has also sent a message "y how are you?" and received a response "[Message received by y.]". The user has also sent a message "y: hi!!" and received a response "[y]: hi!!". The user has also sent a message "y: how are you?" and received a response "[y]: how are you?".

Dereg when server is down



```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c x 127.0.0.10 12345 2137
>>>Welcome, You are registered.
>>>Client table Updated
>>>send y hello
>>>[Message received by y.]
>>>send y how are you?
>>>[Message received by y.]
>>>[y]: hiii
>>>[y]: how are you?
>>>dereg x
>>>[Server not responding]
>>>[Exiting]
mpd2155@csee4119-instance1:~$
```

sending message when client and server both are down



```
mpd2155@csee4119-instance1: ~ -- Google Chrome
ssh.cloud.google.com
mpd2155@csee4119-instance1:~$ python3 Udpchat.py -c x 127.0.0.10 12345 2137
>>>Welcome, You are registered.
>>>Client table Updated
>>>send y hello
>>>[Message received by y.]
>>>send y how are you?
>>>[Message received by y.]
>>>[y]: hiii
>>>[y]: how are you?
>>>dereg x
>>>[Server not responding]
>>>[Exiting]
mpd2155@csee4119-instance1:~$
```