**Project Report**

**On**

“Simple Chat With Python”

Submitted in the Partial fulfillment of the requirement for the Award of Degree of

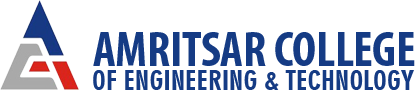
**Bachelor of Technology**

in

**COMPUTER SCIENCE & ENGINEERING**

Batch

(2017-2021)



|  |  |
| --- | --- |
| **Submitted to**  Er.Tejinder Sharma | **Submitted by**  Shekhar Kashyap (1701367)  Shrish Saini (1701370)  Sanchay Khandelwal (1701355) |
|  |  |

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Amritsar College of Engineering & Technology, Amritsar**

**(Autonomous college under UGC Act – 1956[2(f) and 12(B)])**

###### ACKNOWLEDGEMENT

This is a humble effort to express our sincere gratitude towards those who have guided and helped me to complete this project.

A project is major milestone during the study period of a student. As such this project was a challenge to us and was an opportunity to prove our caliber. We are highly grateful and obliged to each and every one making me help out of problems being faced by us.

It would not have been possible to see through the undertaken project without the guidance of **Er. Tejinder Sharma.** It was purely on the basis of their experience and knowledge that we able to clear all the theoretical and technical hurdles during the development phases of this project work.

Last but not the least we are very thankful to our Head of Department **Mr. Vinod Sharma** and all Members of Computer Science Deptt. who gave us an opportunity to face real time problems while fulfilling need of an organization by making projects for them.

**DECLARATION**

We hereby declare that the project work entitled **“Simple Chat With Python”** is an authentic record of my own work carried out as requirements of Institutional project for the award of degree of B.Tech(CSE), **Amritsar College of Engg. And Technology, Amritsar,** under the guidance of **Er. Tejinder Sharma**

(Signature of Students)

Shekhar Kashyap (1701367)

Shrish Saini (1701370)

Sanchay Khandelwal (1701355)

Certified that the above statement made by the student is correct to the best of our knowledge

and belief.

**Faculty Coordinator**

Er. Tejinder Sharma(Assistant Professor – CSE Department)

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| S.NO | CONTENTS | PAGE NO. |
| 1. | INTRODUCTION TO SUBJECT | 5 |
| 2. | INTRODUCTION TO FLAVOURS OF PYTHON | 7 |
| 3. | INTRODUCTION TO SIMPLE CHAT WITH PYTHON | 8 |
| 4.  5. | PROJECT CODE  REFERENCES | 10  19 |
|  |  |  |
|  |  |  |

## Introduction to Python programming

**What is Python…?**

* Python is a general-purpose programming language that is often applied in scripting roles. So, Python is programming language as well as scripting language.
* Python is also called as Interpreted language.

**Differences between program and scripting language**

|  |  |
| --- | --- |
| Program | Script |
| * A program is executed (i.e. the source is first compiled, and the result of that compilation is expected) | * A script is interpreted. |
| * A "program" in general, is a sequence of instructions written so that a computer can perform certain task. | * A "script" is code written in a scripting language. A scripting language is nothing but a type of programming language in which we can write code to control another software application. |

**History of Python: -**

* Invented in the Netherlands, early 90s by Guido van Rossum.
* Python was conceived in the late 1980s and its implementation was started in December 1989. Guido Van Rossum is fan of ‘Monty Python’s Flying Circus’, this is a famous TV show in Netherlands.
* Named after Monty Python.
* Open sourced from the beginning.

**Why was python created?**

"My original motivation for creating Python was the perceived need for a higher-level language in the Amoeba [Operating Systems] project. I realized that the development of system administration utilities in C was taking too long. Moreover, doing these things in the Bourne shell wouldn't work for a variety of reasons. ... So, there was a need for a language that would bridge the gap between C and the shell” - Guido Van Rossum

**Scope of Python Science: -**

* Bioinformatics System Administration
* Unix
* Web logic
* Web sphere Web Application Development
* CGI -Jython
* Servlets Testing scripts

Why do people use Python…?

The following primary factors cited by Python users seem to be these:

1. Python is object-oriented

* Structure supports such concepts as polymorphism, operation overloading, and multiple inheritance.

1. Indentation

* Indentation is one of the greatest futures in Python.

1. It's free (open source)

* Downloading and installing Python is free and easy Source code is easily accessible

1. It's powerful

* Dynamic typing
* Built-in types and tools
* Library utilities –
* Third party utilities (e.g. Numeric, NumPy, SciPy)
* Automatic memory management

1. It's portable

* Python runs virtually every major platform used today
* As long as you have a compatible Python interpreter installed, Python programs will run in exactly the same manner, irrespective of platform.

**INTRODUCTION TO**

**FLAVOURS OF PYTHON**

Here’s a list of Python implementations:

[Jython](http://www.jython.org/): A Python implementation written in Java. Its main advantage is allowing a good interaction with Java libraries and legacy software. Think about being a programmer in a big, old, company where they’ve all their systems written in Java. You can start a new project by writing Python and still using the legacy libraries and services from your company. We haven’t tried it extensively, but people usually say they note Jython as being slower than CPython. The JVM consumes a good amount of memory and it takes some time to load, so that might be the case. The bad news are that much of CPython libraries out there won’t work with Jython (specially the ones written in C). Summary:

* + **Good**: Compatibility with Java libraries and services
  + **Bad**: Arguably slow, lack of compatibility with CPython libraries.

[IronPython](http://ironpython.net/): A Python implementation targeting Microsoft’s .NET framework. Similar to Jython, the main advantage of IronPython, is the ability to work close to the .NET framework. “*IronPython can use the .NET Framework and Python libraries, and other .NET languages can use Python code just as easily.*” Other advantage is being able to use IronPython with Silverlight in a web browser. We could say that it has the same limitations as Jython: many Python libraries (with C bindings) won’t work in IronPython.

* + **Good**: Compatibility with the .NET framework and Silverlight.
  + **Bad**: Lack of compatibility with CPython libraries.

[PyPy](http://pypy.org/) (my personal favorite): PyPy is a Python implementation written in Python (sounds crazy, we know). The beauty of PyPy is the [JIT compiler](https://en.wikipedia.org/wiki/Just-in-time_compilation) built with it. It’s arguably the best compiler written for Python. PyPy is [**crazy fast**](http://pypy.org/performance.html). As Python is a highly dynamic programming language, CPython can’t produce good compiled machine code while executing our programs. That’s why it’s usually said that “python is slow”. PyPy, by using a subset of Python known as RPython, is capable to compile the code to machine optimized code at runtime (just-in-time compilation) and optimize it to be really fast. PyPy is highly compatible with existing Python code, although sometimes it requires a little bit more work. The downside of PyPy is that its current main version is Python 2.7, and the Python 3 support is in beta and moving (arguably) slowly.

* + **Good**: Super fast. Good support of standard libraries and community libraries.
  + **Bad**: Might require more work to use some libraries. Python 3 support limited.

Other implementation/variants worth checking:

* + [Skulpt](http://www.skulpt.org/): Python to Javascript to use in a web browser.
  + [Pyston](https://github.com/dropbox/pyston): A JIT implementation by Dropbox. Looks promising.
  + [Pyjion](https://github.com/Microsoft/Pyjion): A JIT interface for CPython by Microsoft. Guido [has been involved](https://github.com/gvanrossum/Pyjion).
  + [mypy](http://www.mypy-lang.org/): Static typing for Python. Guido is [actively collaborating](https://github.com/python/mypy/commits?author=gvanrossum) with it.

**INTRODUCTION TO SIMPLE CHAT IN PYTHON**

**Socket programming**

Sockets can be thought of as endpoints in a communication channel that is bi-directional, and establishes communication between a server and one or more clients. Here, we set up a socket on each end and allow a client to interact with other clients via the server. The socket on the server side associates itself with some hardware port on the server side. Any client that has a socket associated with the same port can communicate with the server socket.

**Multi-Threading**

A thread is sub process that runs a set of commands individually of any other thread. So, every time a user connects to the server, a separate thread is created for that user and communication from server to client takes place along individual threads based on socket objects created for the sake of identity of each client.  
We will require two scripts to establish this chat room. One to keep the serving running, and another that every client should run in order to connect to the server.

**Server Side Script**

The server side script will attempt to establish a socket and bind it to an IP address and port specified by the user (windows users might have to make an exception for the specified port number in their firewall settings, or can rather use a port that is already open). The script will then stay open and receive connection requests, and will append respective socket objects to a list to keep track of active connections. Every time a user connects,  
a separate thread will be created for that user. In each thread, the server awaits a message, and sends that message to other users currently on the chat. If the server encounters an error while trying to receive a message from a particular thread, it will exit that thread.

**Usage**

This server can be set up on a local area network by choosing any on computer to be a server node, and using that computer’s private IP address as the server IP address.  
For example, if a local area network has a set of private IP addresses assigned ranging from 192.168.1.2 to 192.168.1.100, then any computer from these 99 nodes can act as a server, and the remaining nodes may connect to the server node by using the server’s private IP address. Care must be taken to choose a port that is currently not in usage. For example, port 22 is default for ssh, and port 80 is default for HTTP protocols. So these two ports preferably, shouldnt be used or reconfigured to make them free for usage.  
However, if the server is meant to be accessible beyond a local network, the public IP address would be required for usage. This would require port forwarding in cases where a node from a local network (node that isnt the router) wishes to host the server. In this case, we would require any requests that come to the public IP addresses to be re routed towards our private IP address in our local network, and would hence require port forwarding.

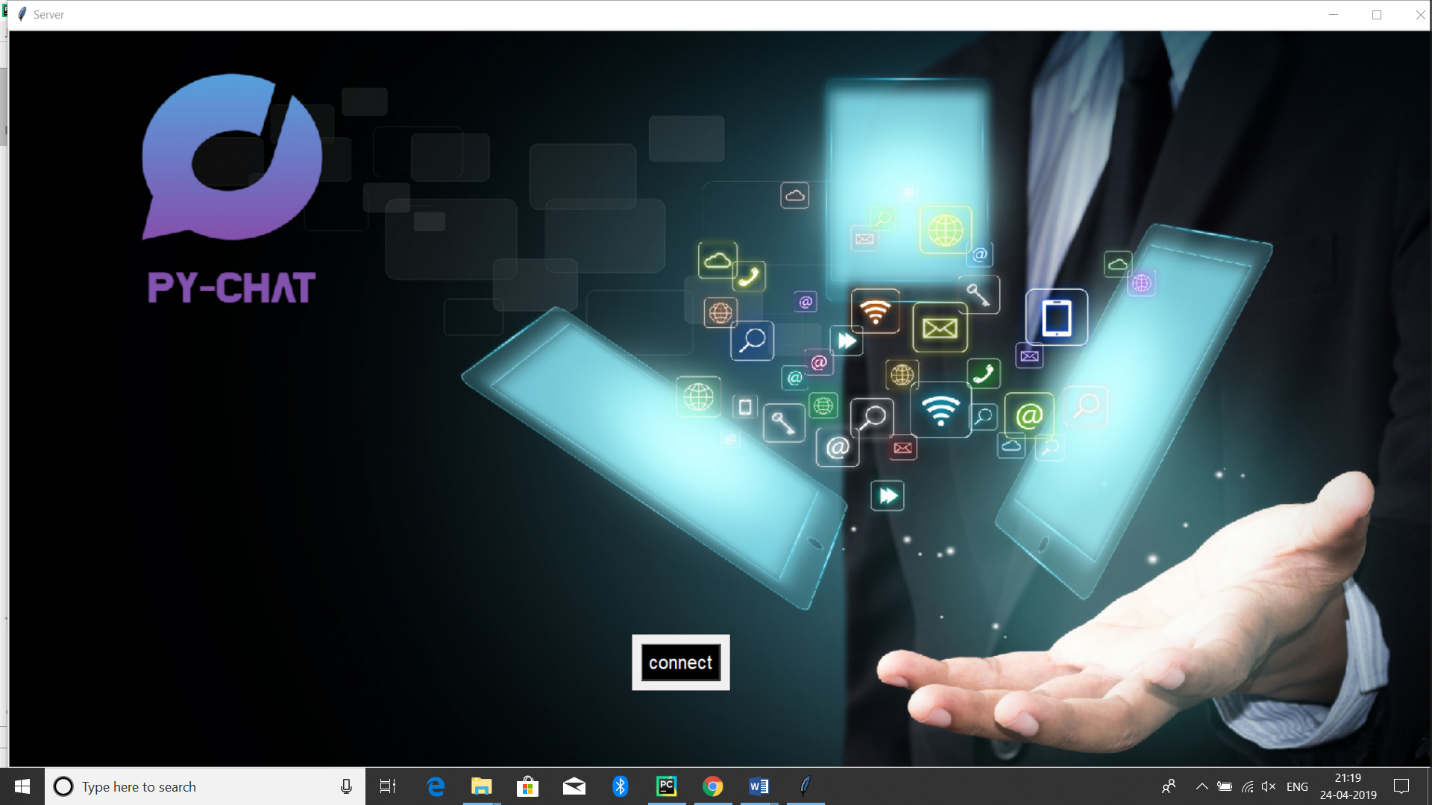
**SOURCE CODE**

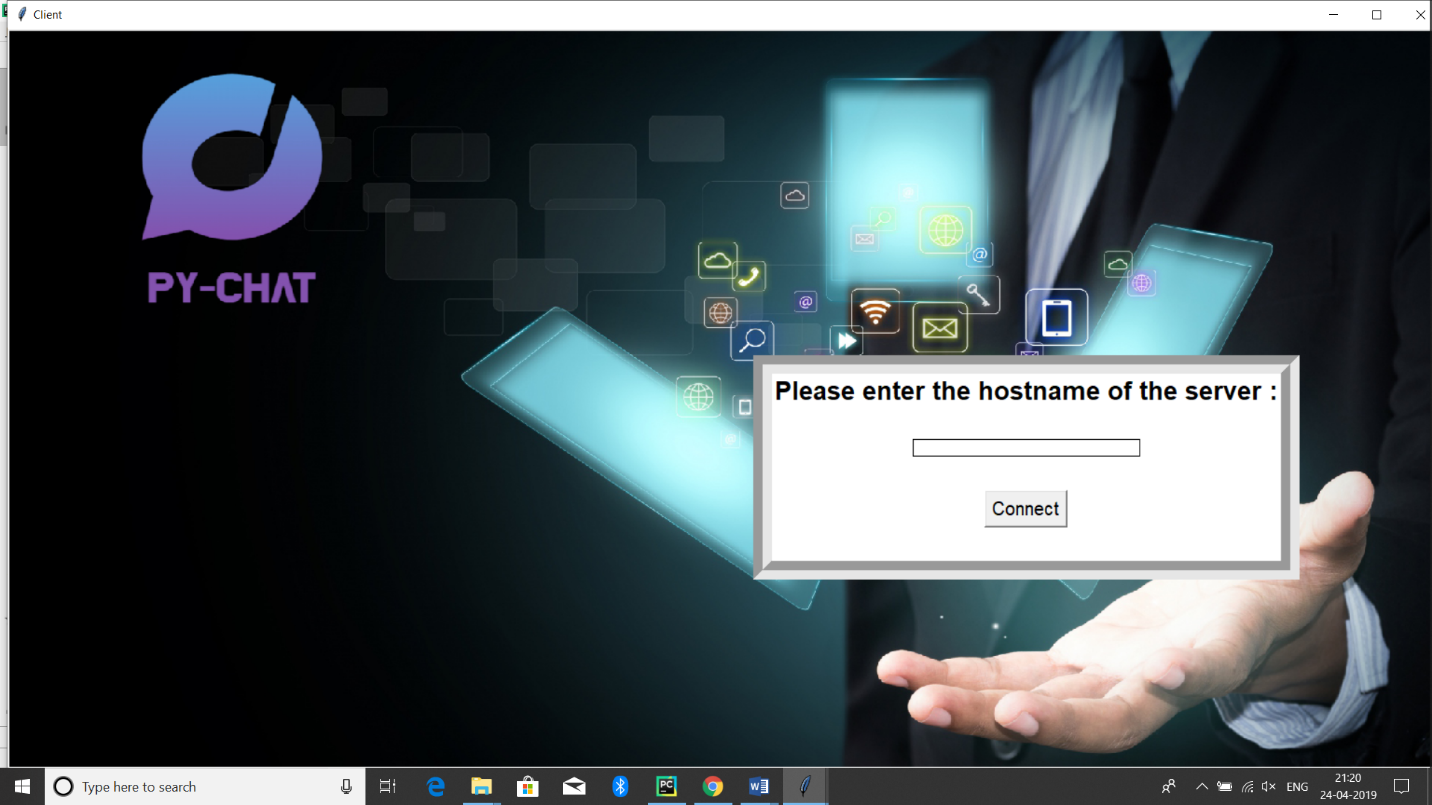
**SERVER SCRIPT :**

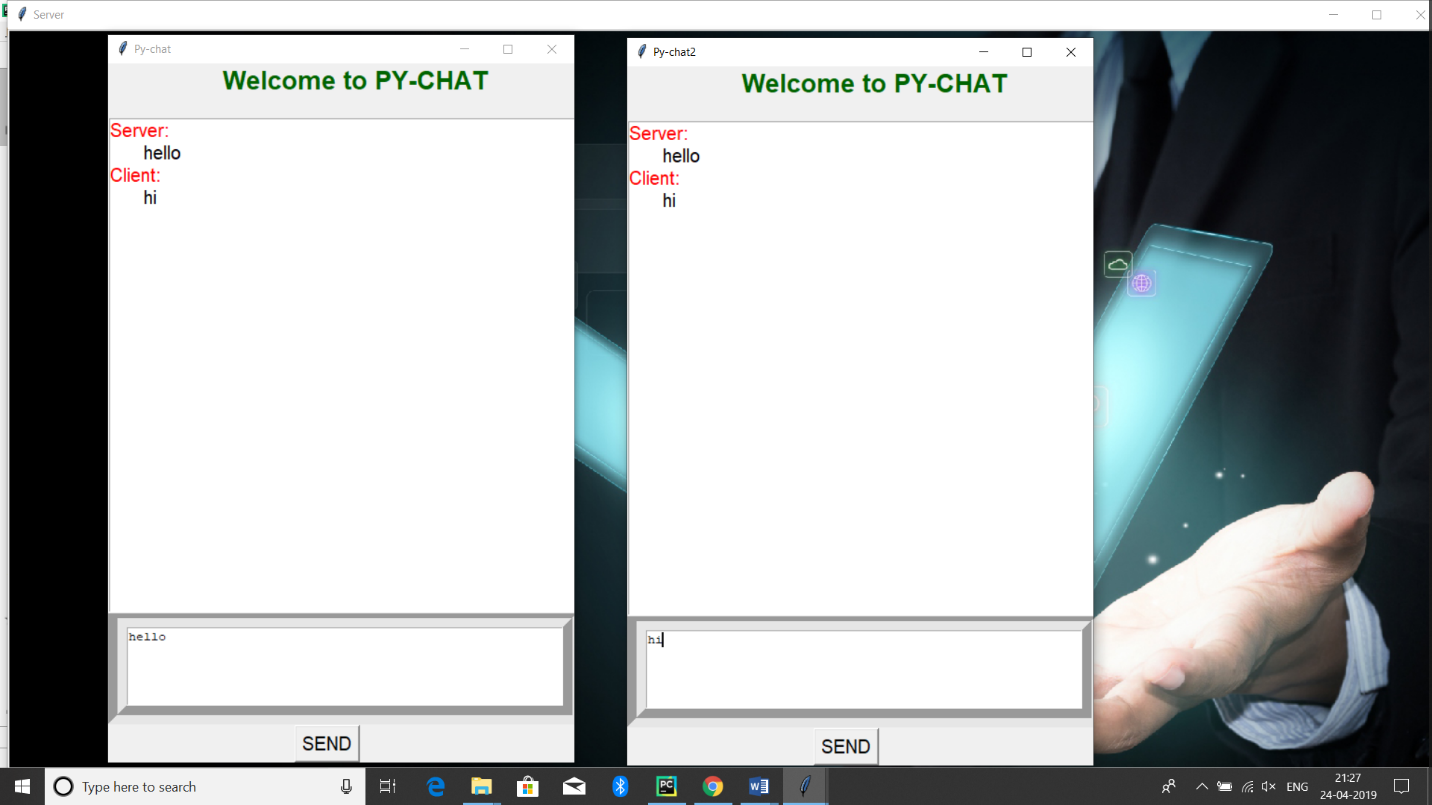
**from** tkinter **import** \*  
**from** time **import** sleep  
**import** socket  
**import** sys  
**from** PIL **import** Image,ImageTk  
**from** threading **import** Thread  
s = socket.socket()  
host =**'127.0.0.1'**port=8081  
s.bind((host, port))  
a=[]  
**def** rec\_mes():  
  
 **while True**:  
 incoming\_message = conn.recv(1024)  
 *# sleep(5)* **if True**:  
 incoming\_message1 = incoming\_message.decode()  
 com = [**"Client:"**, **" {}"**.format(incoming\_message1)]  
  
 **for** i **in** com:  
 a.append(i)  
 lb1.delete(0,END)  
 **for** j **in** range(len(a)):  
 lb1.insert(END, a[j])  
 **if** j % 2 == 0:  
 lb1.itemconfig(j, {**'fg'**: **'red'**})  
**def** thread\_control():  
  
 **def** send\_mes():  
  
  
 message = textbox1.get(**"1.0"**, **"end-1c"**)  
 print(message)  
 com=[**"Server:"**, **" {}"**.format(message)]  
  
  
 **for** i **in** com:  
 a.append(i)  
 lb1.delete(0, END)  
 **for** j **in** range(len(a)):  
 lb1.insert(END,a[j])  
 **if** j % 2 == 0:  
 lb1.itemconfig(j, {**'fg'**: **'red'**})  
  
 print(a)  
  
 message = message.encode()  
 conn.send(message)  
  
 t2 = Thread(target=send\_mes)  
 t2.start()  
  
  
**def** mess():  
 **global** self1  
 self1 = Tk()  
 f2 = Frame(self1,borderwidth=20,width=0, bg=**"white"**, relief=GROOVE)  
 f2.grid(row=0, column=0)  
 f2.place(x=0, y=585)  
 self1.geometry(**"500x750"**)  
 self1.title(**"Py-chat"**)  
 name = Label(self1, text=**"\tWelcome to PY-CHAT"**, fg=**"dark green"**, font=**"Helvetica 20 bold"**)  
 name.grid(row=1,column=0)  
 **global** lb1  
 lb1 = Listbox(self1, width=500, font=35, height=22)  
 lb1.grid(row=9, column=0)  
 lb1.place(x=0, y=58)  
 *#name.place(x=0,y=0)* **global** textbox1  
 can\_button = Button(self1, text=**"SEND"**, font=**"5 "**, command=thread\_control)  
 can\_button.grid(row=1,column=2)  
 can\_button.place(x=200,y=710)  
 textbox1 = Text(f2, height=5, width=58)  
 textbox1.grid(row=0,column=2)  
 *#textbox1.place(x=0,y=570)* t31= Thread(target=rec\_mes)  
 t31.start()  
 self1.mainloop()  
  
  
**def** waiting():  
 **global** conn, addr  
 *#a=None* **while True**:  
 a=s.listen(1)  
  
 **if True**:  
  
 **global** conn,addr  
 conn, addr = s.accept()  
 mess()  
  
**def** liste():  
 t1=Thread(target=waiting)  
 t1.start()  
  
  
*#mess()*self=Tk()  
self.title(**"Server"**)  
self.geometry(**"1550x850"**)  
image = Image.open(**r"C:\Users\Shekhar\Desktop\Certificate\a.jpg"**)  
img = image.resize((1530, 790))  
photo = ImageTk.PhotoImage(img)  
label1 = Label(image=photo)  
label1.grid()  
label1.place(x=0, y=0)  
f2=Frame(self,borderwidth=10)  
f2.grid(row=5,column=0)  
f2.place(x=670, y=650)  
f1=Frame(self,bg=**"black"**)  
f1.grid(row=5,column=0)  
f1.place(x=40, y=60)  
b1=Button(f2,text=**"connect"**,command=liste,font=40,bg=**"black"**,fg=**"white"**).grid(row=9,column=0)  
self.mainloop

**CLIENT SCRIPT:**

**import** socket  
*#import sys***from** time **import** sleep  
**from** tkinter **import** \*  
**from** threading **import** \*  
**from** PIL **import** Image,ImageTk  
  
root=Tk()  
a=[]  
**def** conec():  
 host=uservalue.get()  
 port = 8081  
 s.connect((host,port))  
 l3 = Label(f1, text=**" Connected to chat server"**).grid(row=11,column=0)  
 sleep(2)  
 root.destroy()  
 mes2()  
**def** rec\_mes2():  
 *# message =textbox1.get("1.0","end-1c")* **while True**:  
 incoming\_message = s.recv(1024)  
 **if True**:  
 incoming\_message1 = incoming\_message.decode()  
 com = [**"Server:"**, **" {}"**.format(incoming\_message1)]  
  
 **for** i **in** com:  
 a.append(i)  
 lb1.delete(0, END)  
 **for** j **in** range(len(a)):  
 lb1.insert(END, a[j])  
 **if** j % 2 == 0:  
 lb1.itemconfig(j, {**'fg'**: **'red'**})  
**def** chatpy2():  
  
 **def** send\_mes2():  
 message = textbox1.get(**"1.0"**, **"end-1c"**)  
  
 com = [**"Client:"**, **" {}"**.format(message)]  
  
 **for** i **in** com:  
 a.append(i)  
 lb1.delete(0, END)  
 **for** j **in** range(len(a)):  
 lb1.insert(END, a[j])  
 **if** j % 2 == 0:  
 lb1.itemconfig(j, {**'fg'**: **'red'**})  
  
 message = message.encode()  
 s.send(message)  
 t4 = Thread(target=send\_mes2)  
 t4.start()  
  
**def** mes2():  
 **global** self1  
 self1 = Tk()  
 f2 = Frame(self1, borderwidth=20, width=0, bg=**"white"**, relief=GROOVE)  
 f2.grid(row=0, column=0)  
 f2.place(x=0, y=585)  
 self1.geometry(**"500x750"**)  
 self1.title(**"Py-chat2"**)  
 name = Label(self1, text=**"\tWelcome to PY-CHAT"**, fg=**"dark green"**, font=**"Helvetica 20 bold"**)  
 name.grid(row=1, column=0)  
 **global** lb1  
 lb1 = Listbox(self1, width=500, font=35, height=22)  
 lb1.grid(row=9, column=0)  
 lb1.place(x=0, y=58)  
 *# name.place(x=0,y=0)* **global** textbox1  
 can\_button = Button(self1, text=**"SEND"**, font=**"5 "**, command=chatpy2)  
 can\_button.grid(row=1, column=2)  
 can\_button.place(x=200, y=710)  
 textbox1 = Text(f2, height=5, width=58)  
 textbox1.grid(row=0, column=2)  
 t3 = Thread(target=rec\_mes2)  
 t3.start()  
 self1.mainloop()  
s = socket.socket()  
root.title(**"Client"**)  
root.geometry(**"1550x850"**)  
image = Image.open(**r"C:\Users\Shekhar\Desktop\Certificate\a.jpg"**)  
img = image.resize((1530, 790))  
photo = ImageTk.PhotoImage(img)  
label1 = Label(image=photo)  
label1.grid()  
label1.place(x=0, y=0)  
f1=Frame(root,borderwidth=20,bg=**"white"**,relief=GROOVE)  
f1.grid(row=5,column=0)  
f1.place(x=800, y=350)  
l1 = Label(f1, text=**"Please enter the hostname of the server :\n "**,font=**"Helvetica 20 bold"**,bg=**"white"**).grid(row=6,column=0)  
uservalue=StringVar()  
userentry = Entry(f1,textvariable=uservalue,width=40,relief=SOLID)  
userentry.grid(row=7,column=0)  
l1 = Label(f1, text=**"\n "**,bg=**"white"**).grid(row=8,column=0)  
ll=Button(f1,text=**"Connect"**,command=conec,font=20).grid(row=9,column=0)  
l1 = Label(f1, text=**"\n "**,bg=**"white"**).grid(row=10,column=0)  
mainloop()







**References :**

1. **Books**

* Computers Today by Sanders.
* Fundamentals of Computers TTTI Publication.
* Learning Python by Mark Lutz, 5th Edition
* Python Cookbook, by David Beazley , 3rd Edition

1. **Web URLs**

* <https://www.tutorialspoint.com/>
* <https://www.geeksforgeeks.org/>
* <https://www.w3schools.com/>