**Simple Chat Room using Python**

**Dr.Muthusundari1 , Jafar Sadiq A2 , Jagan J 3 , Jeeva K4 , Jaganath B5**

1Associate Professor, Department of Computer Science and Engineering, R.M.D Engineering College, Kavaraipettai, Tiruvallur, sms.cse@rmd.ac.in.

2, 3, 4,5Student, Department of Computer Science and Engineering, R.M.D Engineering College, Kavaraipettai,Tiruvallur

**ABSTRACT:**

Chat refers to the process of communicating, interacting and/or exchanging messages over the Internet. It involves two or more individuals that communicate through a chat-enabled service or software. Chat may be delivered through text, audio or video communication via the Internet.A chat application has basic two components, viz server and client. A server is a computer program or a device that provides functionality for other programs or devices. Clients who want to chat with each other connect to the server.The chat application we are going to make will be more like a chat room, rather than a peer to peer chat. So this means that multiple users can connect to the chat server and send their messages. Every message is broadcasted to every connected chat user.

**INTRODUCTION:**

Python is an interpreted high-level programming language for general-purpose programming. Python features a dynamic type system and automatic memory management. Python's simple syntax, modules and packages makes it possible for us to develop applications rapidly. Professionally, Python is great for backend web development, data analysis, artificial intelligence, and scientific computing. Many developers have also used Python to build productivity tools, games, and desktop apps.

The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**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.

**CLIENT SIDE SCRIPT**

The client side script will simply attempt to access the server socket created at the specified IP address and port. Once it connects, it will continuously check as to whether the input comes from the server or from the client, and accordingly redirects output. If the input is from the server, it displays the message on the terminal. If the input is from the user, it sends the message that the users enters to the server for it to be broadcasted to other users.

**LITERATURE REVIEW:**

Authors:Vicky, Chau Ka Ki.

Internet communication is getting more and more popular among the public. Apart from using telephones or automobiles and sending mails, people can now communicate with each other through the chat technology. The chat, in fact, is a kind of Internet technology that supports human-to-human communication. ICQ, for instance, is one of the latest chat. Over the past two years, with the advanced level of technology, there is an increasing trend of using ICQ for communication. With ICQ, users can chat, send messages, files and URL’s or play games with others users in real time. Because of the proliferation of using the chat like ICQ, studies have been focused mainly on its impact on our society.

Much of the work stresses the good impact of the chat. Hauben’s (1997) writing suggested that as the impact or influence of first impressions is removed, users are free to communicate without fears, limits or apprehension through the chat. This statement actually points out the main reason for the increasingly use of the chat. Only one advantage, however, seems inadequate to attract such a huge number of users to use the chat, so it seems that there may be other benefit. Accordingly, Licklider (1997) claimed that people can communicate online with others who have similar goals and interests, thus their life will be enriched and communication will be more productive and more enjoyable then. Although Licklider is actually the prophet of the Net, it seems that the chat really has this benefit.

Some studies, however, have taken a different approach by looking not so much on the advantages of the chat, but focusing more on its related problems. Randall (1997), for instance, mentions that problems have actually been existed. First of all, there is no doubt that the chat users will not use their real identities for communication. They will rather create a new cyberspace identities which are very different from their real ones. Because of this, Randall argued that such behavior makes people difficult to switch back and forth between these two identities. To him, those who have developed multiple cyberspace identities for Internet communication are the most sophisticated rhetoricians on the Internet. In fact, Randall questions whether it is credible to create a new identity when communicating through the chat.

In Randall’s viewpoint, on the other hand, the purpose of people who use the chat is for socializing. But he emphasized that such kind of socializing is different from that in the real world, as the former only involves the exchange of words with other users but the latter means to interact with others face-to-face. While the trend of using the chat is increasing, Randall suggested that children and youths will be discouraged from the normal social contact but will adopt cyberspace contact instead. However, although the chance of happening this phenomenon is quite high, it seems neither right nor wrong according to Randall.

While educators and students are expecting online education to be existed, Randall has already shown concern on the consequence of using such kind of education. In his point of view, the traditional teacher dominance of the classroom will be reduced, no matter this education is workable or not, because of the poor financial situation of the government. It seems that unemployment will be resulted in the near future.

To sum up, the chat has good impact on the society but problems exist at the same time. However, these problems are not serious in fact. Therefore, even if these problems exist continuously, the chat technology will still become central to our lives and it has already begun actually.

**METHODOLOGY**

This Simple Chat Room application will enable the user to chat with the logged users in the chat room. The server should be active. The users shall connect to this server at the predefined port number. To start using this tool, the user has to register with this tool. Through the valid login, password details, the application shall allow the user to use the chat room. The user shall be able to connect to this server and can chat with other users. The logged in user can view other active users in the chat room. The user can initiate the conversation through the chat window. For the communication to happen, both the users should have connected to the server. This application is developed using visual Basic projects. The windows sockets are used in the application for communication between client and server. They provide a robust way of communication between nodes. The user also can clear the chat transcript if required. It also provides an option to save the chat transcript. Users can view the status of other users in the chat room like if the user is currently online or offline or busy.

**MATERIALS AND METHODS:**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

Import the Tkinter module.

Create the GUI application main window.

Add one or more of the above-mentioned widgets to the GUI application.Enter the main event loop to take action against each event triggered by the user.

Tkinter Widgets:

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

Tk(screenName=None, baseName=None, className=’Tk’, useTk=1): To create a main window, tkinter offers a method ‘Tk(screenName=None, baseName=None, className=’Tk’, useTk=1)’. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

m=tkinter.Tk() where m is the name of the main window object

mainloop(): There is a method known by the name mainloop() is used when you are ready for the application to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.

m.mainloop()

import tkinter m = tkinter.Tk() '''

widgets are added here '''

m.mainloop()

Tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

pack() method : It organizes the widgets in blocks before placing in the parent widget.

grid() method : It organizes the widgets in grid (table-like structure) before placing in the parent widget.

place() method : It organizes the widgets by placing them on specific positions directed by the programmer. There are a number of widgets which you can put in your tkinter application. Some of the major widgets are explained below:

Button : To add a button in your application, this widget is used. The general syntax is:

w=Button (master, option=value)

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the Buttons. Number of options can be passed as parameters separated by commas. Some of them are listed below.

activebackground: to set the background color when button is under the cursor. activeforeground: to set the foreground color when button is under the cursor. bg: to set he normal background color command: to call a function.

font: to set the font on the button label. image: to set the image on the button. width: to set the width of the button. height: to set the height of the button.

import tkinter as tk r = tk.Tk()

r.title('Counting Seconds')

button = tk.Button(r, text='Stop', width=25, command=r.destroy) button.pack()

r.mainloop()

Output:

Entry : It is used to input the single line text entry from the user.. For multi- line text input, Text widget is used.

The general syntax is:

w=Entry(master, option=value)

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

bd: to set the border width in pixels. bg: to set the normal background color. cursor: to set the cursor used. command: to call a function.

highlightcolor: to set the color shown in the focus highlight.

width: to set the width of the button.

height: to set the height of the button.

from tkinter import \* master = Tk()

Label(master, text='First Name').grid(row=0) Label(master, text='Last Name').grid(row=1) e1 = Entry(master)

e2 = Entry(master) e1.grid(row=0, column=1) e2.grid(row=1, column=1) mainloop()

Scrollbar : It refers to the slide controller which will be used to implement listed widgets.

The general syntax is:

w = Scrollbar(master, option=value)

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

width: to set the width of the widget.

activebackground: To set the background when mouse is over the widget.

bg: to set he normal background color.

bd: to set the size of border around the indicator.

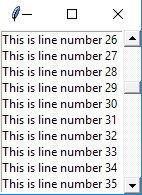
cursor: To appear the cursor when the mouse over the menubutton. from tkinter import \*

root = Tk()

scrollbar = Scrollbar(root) scrollbar.pack( side = RIGHT, fill = Y )

mylist = Listbox(root, yscrollcommand = scrollbar.set ) for line in range(100):

mylist.insert(END, 'This is line number' + str(line)) mylist.pack( side = LEFT, fill = BOTH ) scrollbar.config( command = mylist.yview ) mainloop()



Sockets allow communication between two different processes on the same or different machines. To be more precise, it's a way to talk to other computers using standard Unix file descriptors. In Unix, every I/O action is done by writing or reading a file descriptor. A file descriptor is just an integer associated with an open file and it can be a network connection, a text file, a terminal, or something else.

To a programmer, a socket looks and behaves much like a low-level file descriptor. This is because commands such as send() and recv() work with sockets in the same way they do with files and pipes

A socket is one endpoint of a two-way communication link between two programs running on the network.

A socket is bound to a port number so that the tcp layer can identify the application that the data is destined to be sent to.

Normally, a server runs on a specific computer and has a socket that is bound to a specific port number.

The server just waits, listening to the socket for a client to make a connection request.

On the client side : The client knows the host name of the machine of which the server is running and the port number in which the server is listening.

To make a connection request .

The client also needs to identify itself to the server so it binds to local port number that it will use during connection. This is usually assigned by system.

If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote and endpoint set to the address and port of client.

It needs a new socket for connection requests while trending to the needs of connected client.

On the client side, if the connection is accepted a socket is successfully created and the client can use the socket to communicate with the server.

The client and server can now communicate by writing to or reading from their sockets.

Server Socket Methods

Method & Description

1 s.bind()

This method binds address (hostname, port number pair) to socket.

2 s.listen()

This method sets up and start TCP listener.

3 s.accept()

This passively accept TCP client connection, waiting until connection arrives (blocking).

THE CLIENT:

This module manage any client willing to connect on a specific host and port. It authorizes the client to send and

receive messages. To do:

Write or improve docstrings

Create a specific log file to log crashes and exceptions.

Client Socket Methods

Method & Description

1 s.connect()

This method actively initiates TCP server connection.

ADVANTAGES:

If developed further by adding a database, frontend and buying a server place on the internet, this project can be converted into a real time chat application.

LIMITATION:

The clients can only chat by connecting to a local server.

Two clients on different networks cannot chat together.

Hardware Requirments:

Processor Pentium IVRAM

128 MB

Monitor 15 in

Hard disk 20 GB

Floppy drive 1.44 MB

CD drive

Key board

Mouse

Software Requirements:

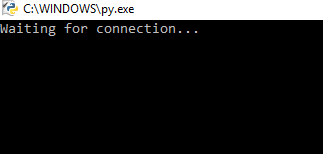
OperatingsystemWindowsXP ProfessionalLanguage Visual Basic 6.0

Windows sockets

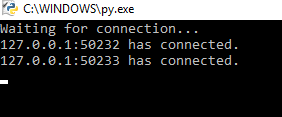
1. RESULTS AND DISCUSSION:

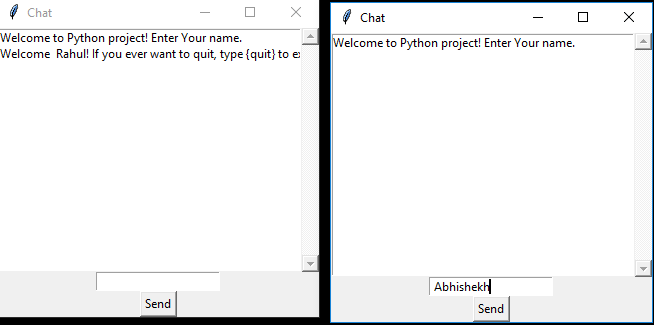
Working model:

SERVER IS RUNNING AND WAITING FOR CLIENT

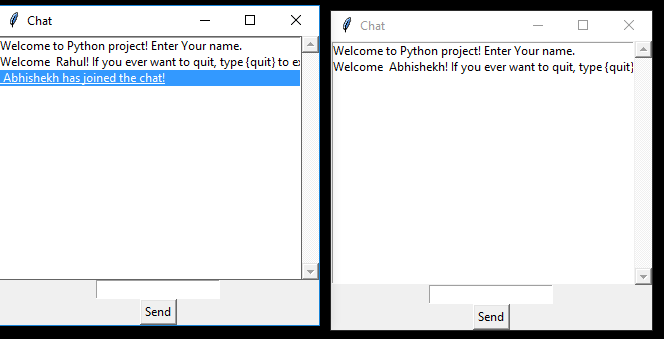


TWO CLIENTS ARE CONNECTED TO SERVER

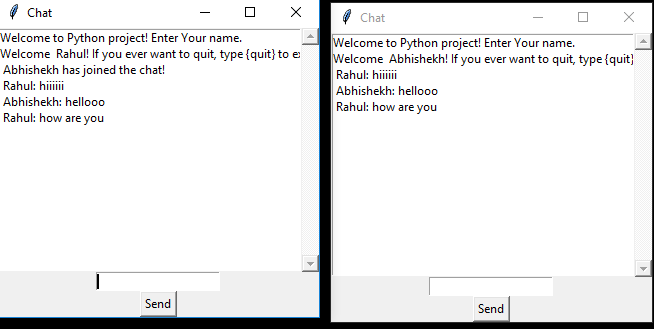


CLIENTS ENTERING THEIR NAME

I

CLIENTS GETTING NOTIFIED WHO HAS JOINED THE CHAT

CLIENTS ARE CHATTING BY CONNECTING TO SERVER



6. Conclusion:

In this project we have used the basics of networking in python. We also learned how to make a GUI for our application. This project teaches us how to create a basic chat application by creating a local network and using TCP sockets.

7. References :

Randall. N (1997). Chapter 19: Future Communities: Socializing and Educating in the Internet’s Future, The soul of the Internet: net gods, netizens and the wiring of the world(pp.333-344). London: Computer Press.

Randall. N (1997). Epilogue: The Soul of the Internet, The soul of internet: net gods, netizens

and the wiring of the world(pp.345-358). London: Computer Press.

Hauben .M (1997). Chapter 1:The Net and Netizens: The Effect the Net Has on People’s Lives, Netizens(3-34). Los Alamitos: Computer Society Press.

Hauben. M (1997). Chapter2: The Evolution of Usenet: The Poor Man’s ARPANET, Netizens(35-47). Los Alamitos: Computer Society Press.

Hauben. M (1997). Chapter 6: Cybernetics, Time-sharing, Human-Computer Symbiosis and Online Communities: Creating a Supercommunity of Online Communities, Netizens(76-95).Los Alamitos: Computer Society Press

<https://levelup.gitconnected.com/learn-python-by-building-a-multi-user-group-chat-gui-application-af3fa1017689>

<https://pythonprogramming.net/server-chatroom-sockets-tutorial-python-3/>

<https://www.pubnub.com/blog/socket-programming-in-python-client-server-p2p/>

.