Networking laboratory

Client-Server based instant messaging

-- Y Prahasith, Vivek R Navale, Shashank Chandavarkar

Literature survey

Introduction

Humans are social beings and one of the most important aspects of our life is communication. We started with languages and could communicate at small distances at the start but as our lives have become sophisticated we had to communicate via larger distances and networks came into existence. As it is such an important aspect of our life various research have been carried out to make it more efficient to communicate in an easy and simple manner. We have taken up one of the most recent methods and implemented it.

DESIGN

We use tcp socket instances to create connections between a client and a server.A socket will be tied to some port on some host. In general, you will have either a client or a server type of entity or program.

In the case of the server, you will bind a socket to some port on the server (localhost). In the case of a client, you will connect a socket to that server, on the same port that the server-side code is using.So we can connect two different clients to our server instance and have instant messaging between the two client entities.

IMPLEMENTATION

We need two entities namely a server entity and a client entity.

Server

The server has to accept new connections from clients , identify all the clients. For this we can use the clients ip address or we can ask for unique usernames from the client. So the server entity will first allow clients to connect and choose a username.Beyond this, the server will collect incoming messages and then distribute them to the rest of the connected clients.

We now use sockets to create a server instance and bind it to a port(we use server_instance.bind( )). The client can connect to the server using the same port. We then listen for connections using the **.**listen( ) function. To keep track of all the sockets connected to the port we use the list data structure and a dictionary to keep track of clients.

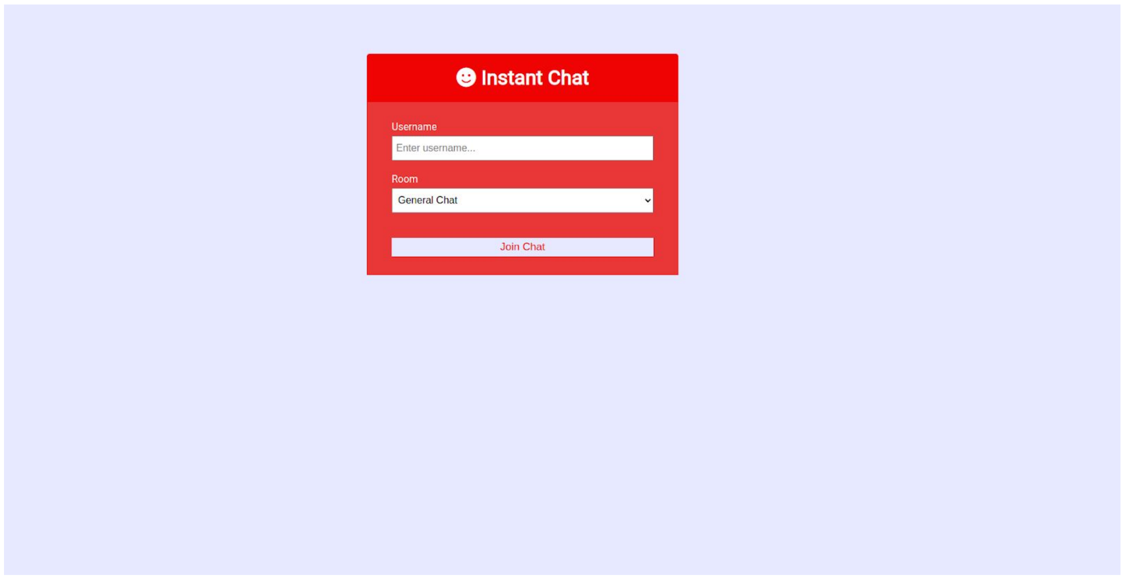
The server's main job is to receive messages, and then disperse them to the connected clients. For receiving messages, we're going to make a function that checks a message by checking if the header length is valid and then broadcasts the message to the concerned client. A new client is added by appending the socket to the list and the username to the dictionary. The server program has other functions as well like checking for disconnection of clients and to resolve multiple issues.

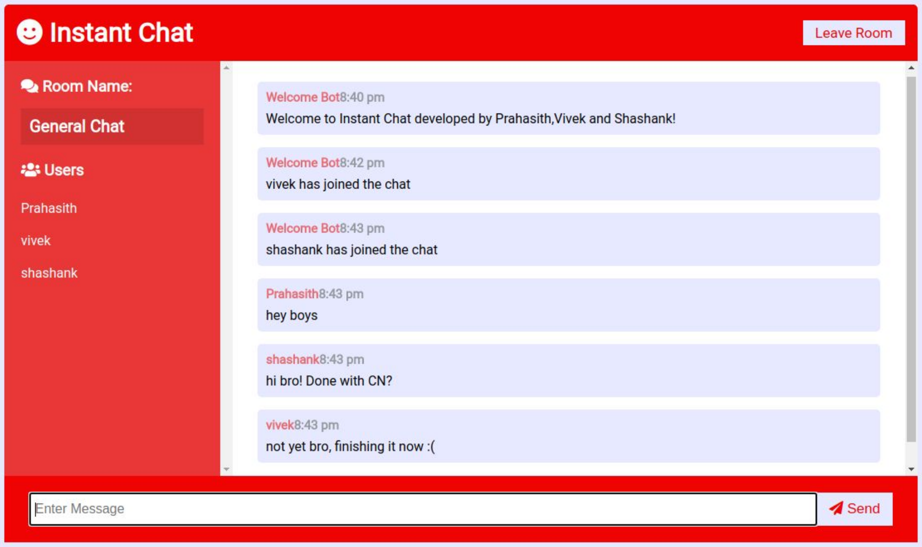
Client

We connect the socket instance of the client to the port of the server. At the start of the service we ask the user for his username. We accept the multiple messages he sends using a loop and send it to the server. Another client instance(the receiver) has to now receive the message. The client code has an infinite loop that checks for messages and this loop breaks only when there is an error that no more messages are sent. The message is displayed as {username}>> message to both the clients(the sender and the receiver).

The client program contains code to act as a receiver and a sender at the same time. To synchronize the messages we use the concepts of threads.

Results





Instant Chat

Leave Room

Room Name:

General Chat

Users

Prahasith

vivek

shashank

Welcome Bot8:40 pm
Welcome to Instant Chat developed by Prahasith,Vivek and Shashank!

Welcome Bot8:42 pm
vivek has joined the chat

Welcome Bot8:43 pm
shashank has joined the chat

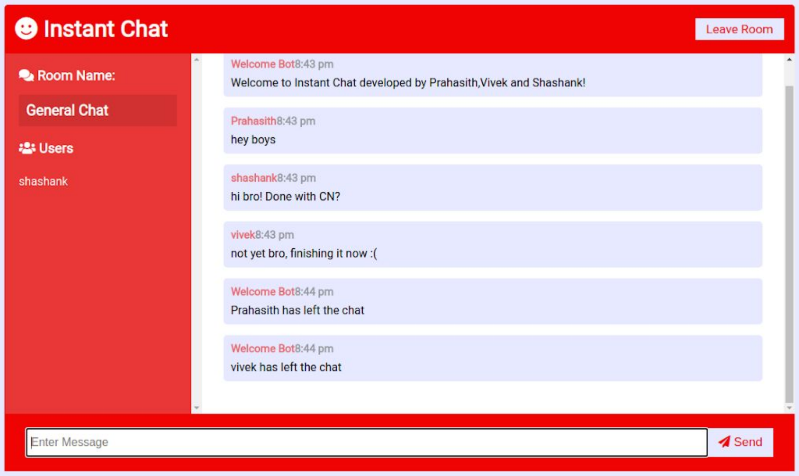
Prahasith8:43 pm
hey boys

shashank8:43 pm
hi bro! Done with CN?

vivek8:43 pm
not yet bro, finishing it now :(

Enter Message

Send



Instant Chat

Leave Room

Room Name:

General Chat

Users

shashank

Welcome Bot8:43 pm
Welcome to Instant Chat developed by Prahasith,Vivek and Shashank!

Prahasith8:43 pm
hey boys

shashank8:43 pm
hi bro! Done with CN?

vivek8:43 pm
not yet bro, finishing it now :(

Welcome Bot8:44 pm
Prahasith has left the chat

Welcome Bot8:44 pm
vivek has left the chat

Enter Message

Send

Conclusion

- Github link - https://github.com/YPrahasith/Instant-Chat-Messenger