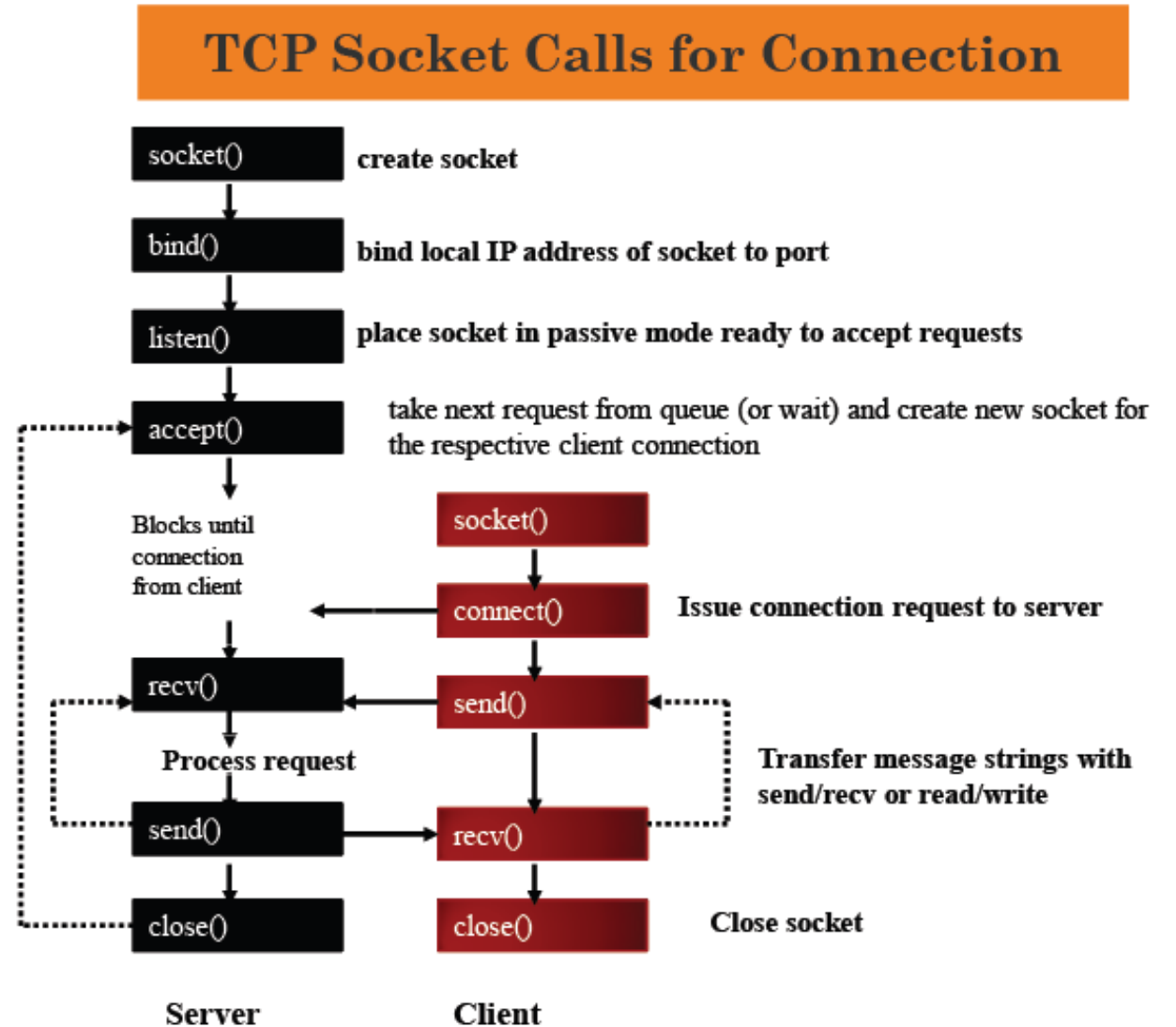


Methodology used

Flowchart for the project.



Methodology used_(cont.)



Firstly, WSA is initialized and socket is opened and then the setsockopt is called to connect the clients to the server through IP address of the server's machine.



The clients will join the server using their local ports.



Then the server will listen to the requests from up to 5 clients and the other clients will wait in the waiting room until any client disconnects.



The server will wait till the time any client requests to join.



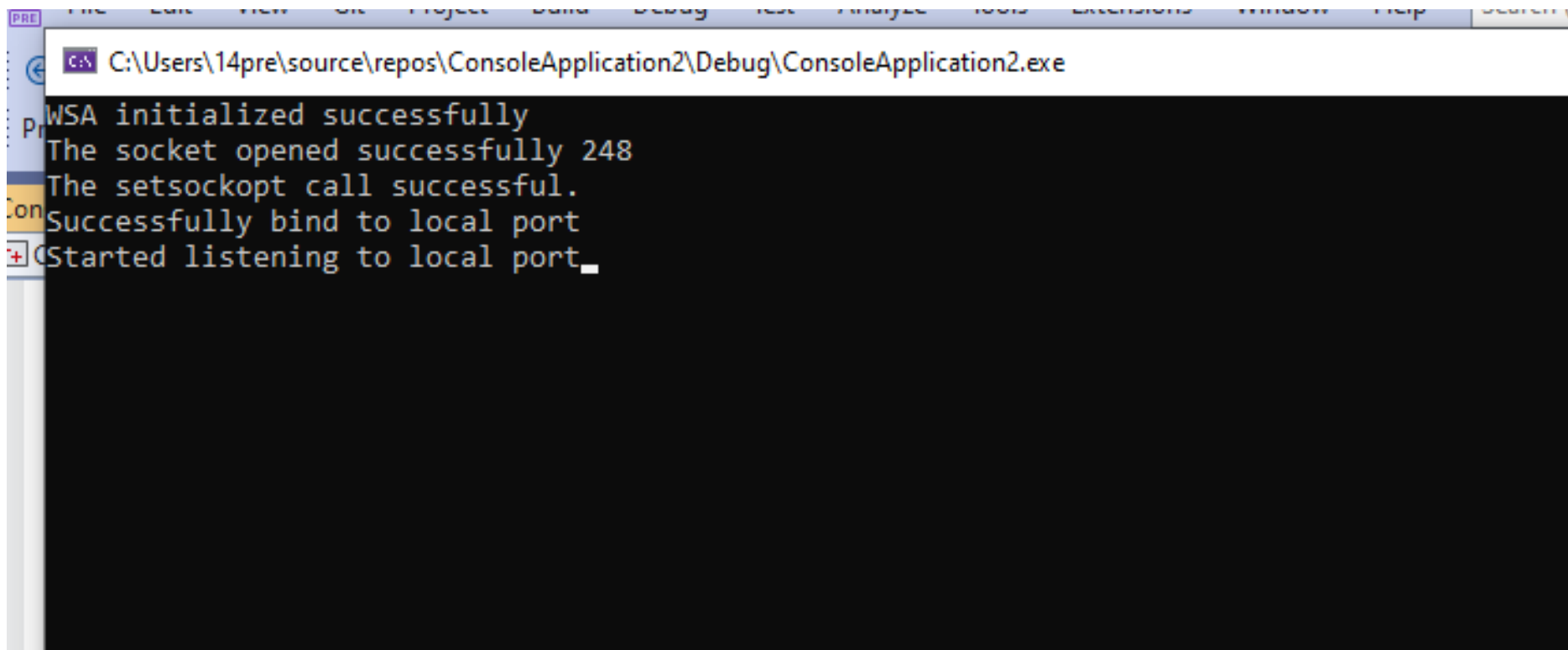
When the client requests the server to connect, the server starts processing the data from that client's port.



Not more than 5 clients can join at the same time and if any client disconnects the server will inform the other clients about the same and will make space for other client whose waiting to join.

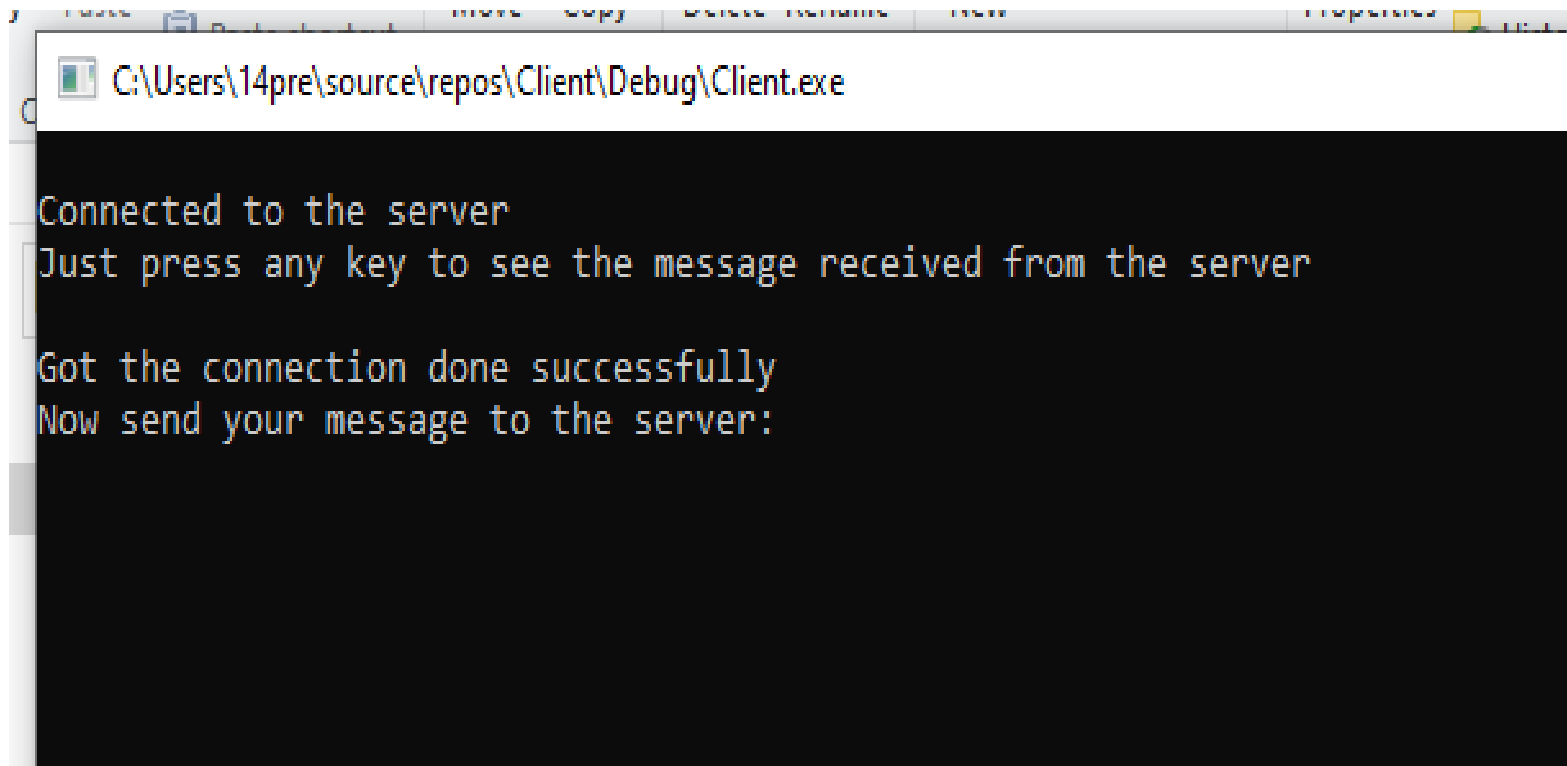
Results and Outputs

- The server needs to start listening to the request of the clients (upto 5).



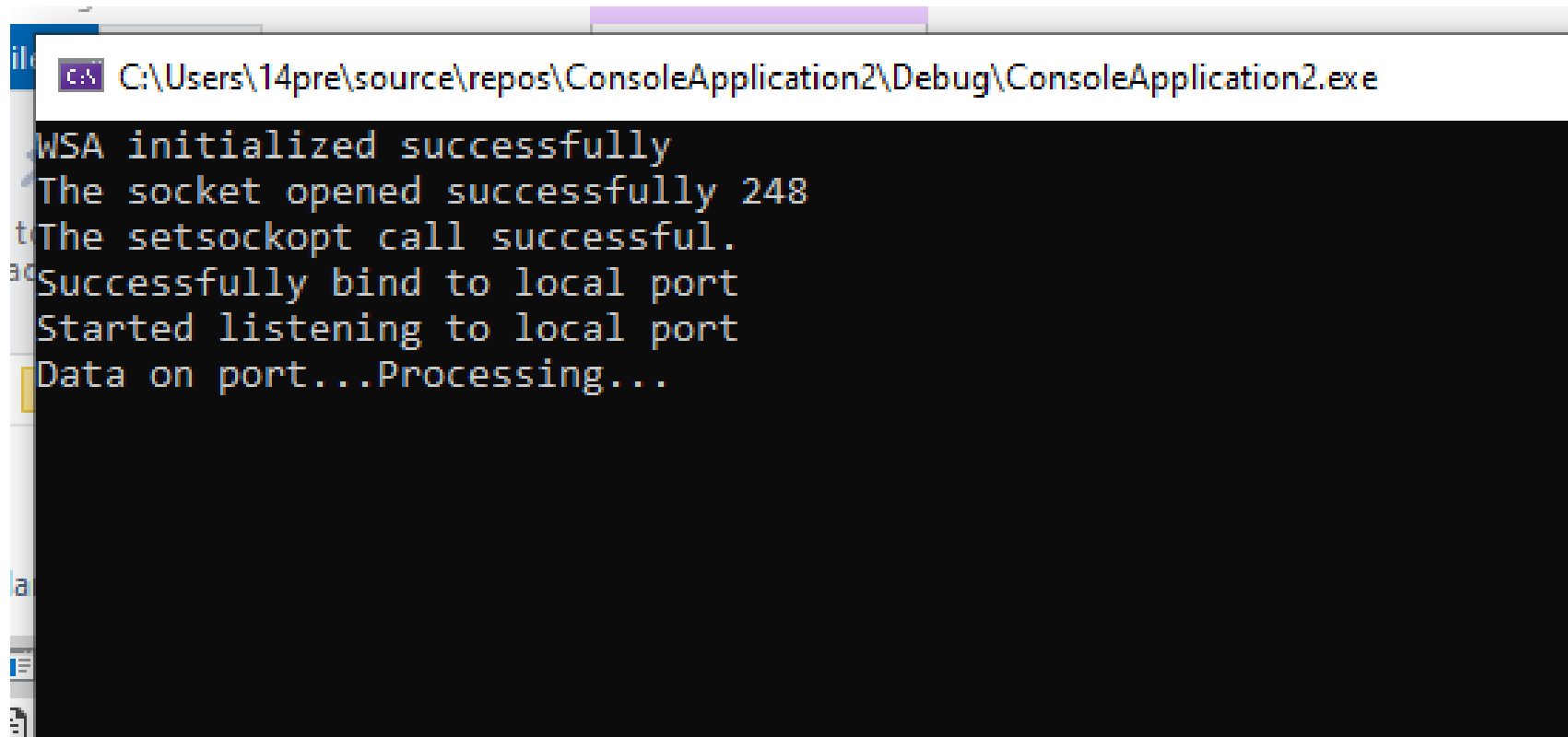
```
C:\Users\14pre\source\repos\ConsoleApplication2\Debug\ConsoleApplication2.exe
WSA initialized successfully
The socket opened successfully 248
The setsockopt call successful.
Successfully bind to local port
Started listening to local port_
```

- The clients will send request to the server to join the chat.



```
C:\Users\14pre\source\repos\Client\Debug\Client.exe  
Connected to the server  
Just press any key to see the message received from the server  
Got the connection done successfully  
Now send your message to the server:
```

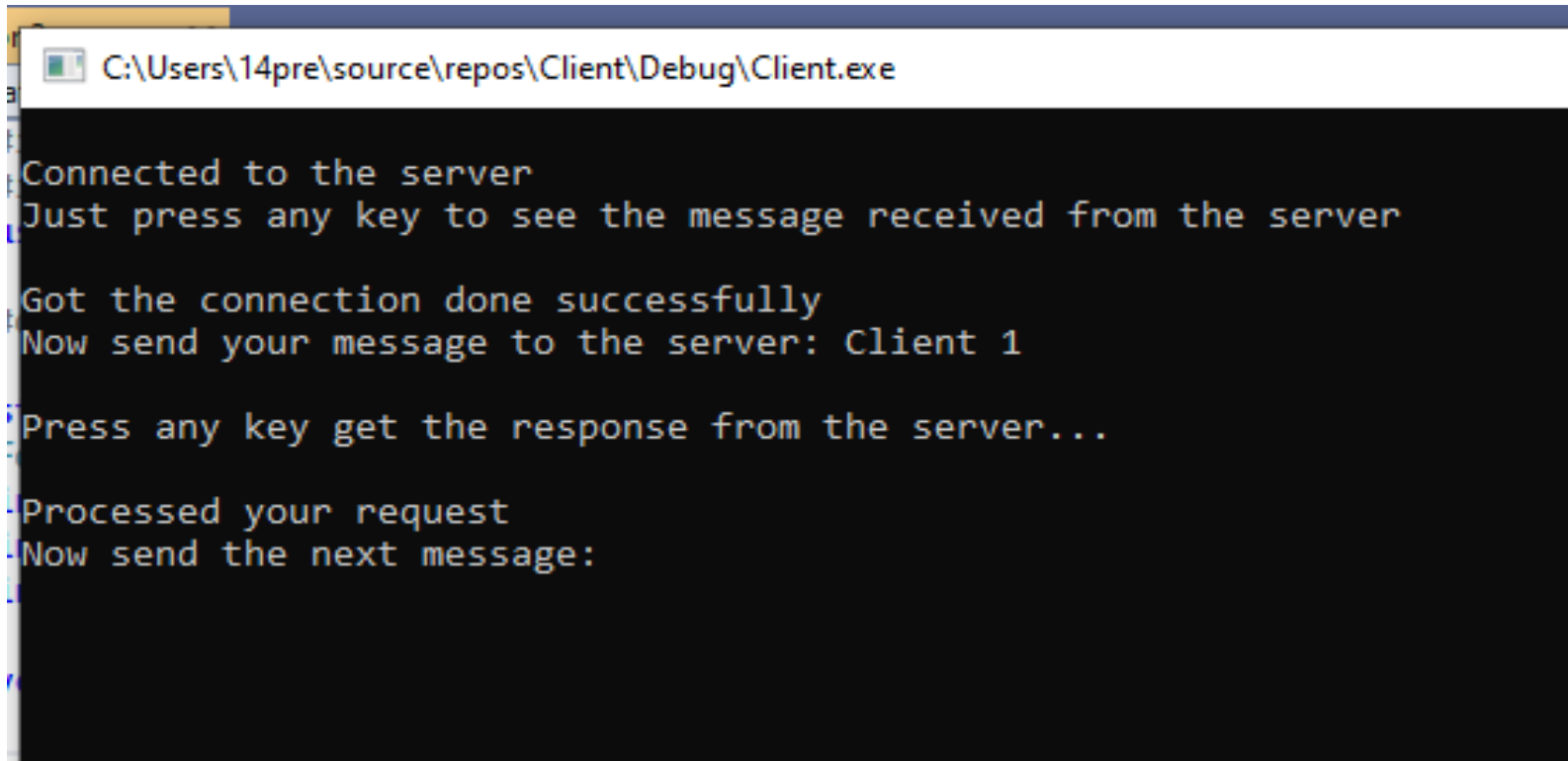
- The server will start to process the data of the clients.



A screenshot of a Windows command prompt window. The title bar shows the file path: `C:\Users\14pre\source\repos\ConsoleApplication2\Debug\ConsoleApplication2.exe`. The command prompt has a black background with white text. The output of the application is as follows:

```
WSA initialized successfully
The socket opened successfully 248
The setsockopt call successful.
Successfully bind to local port
Started listening to local port
Data on port...Processing...
```

- The clients will start messaging the server. They will get a response that their request has been processed and they can send the next message.



```
C:\Users\14pre\source\repos\Client\Debug\Client.exe

Connected to the server
Just press any key to see the message received from the server

Got the connection done successfully
Now send your message to the server: Client 1

Press any key get the response from the server...

Processed your request
Now send the next message:
```

- Clients can talk among themselves.

```
C:\Users\14pre\source\repos\ConsoleApplication2\Debug\ConsoleApplication2.exe
WSA initialized successfully
The socket opened successfully 248
The setsockopt call successful.
Successfully bind to local port
Started listening to local port
Data on port...Processing...
Data on port...Processing...
Processing the new message for the client socket: 252
The message received from the client is : Client 1

*****
Data on port...Processing...
Data on port...Processing...
Processing the new message for the client socket: 256
The message received from the client is : Client 2

*****
Data on port...Processing...
Data on port...Processing...
Processing the new message for the client socket: 260
The message received from the client is : Client 3

*****
Data on port...Processing...
Processing the new message for the client socket: 252
The message received from the client is : Hello from Client 1

*****
Data on port...Processing...
Processing the new message for the client socket: 256
The message received from the client is : Hello from Client 2

*****
Data on port...Processing...
Processing the new message for the client socket: 260
The message received from the client is : Hello from Client 3

*****
```

- If something goes wrong and the client disconnects, the server will inform other clients by closing the connection of the client's socket.

```
Processing the new message for the client socket: 256
The message received from the client is : Hello from Client 2

*****

Data on port...Processing...
Processing the new message for the client socket: 260
The message received from the client is : Hello from Client 3

*****

Data on port...Processing...
Processing the new message for the client socket: 256
Something wrong happened. Closing the connection for client
```


- The server will have no effect if something happens with a client and other clients can still talk. If a client disconnects, the server will open room for other client to enter the chat, whose waiting to connect.

```
*****
Data on port...Processing...
Processing the new message for the client socket: 260
The message received from the client is : Hello from Client 3
*****
Data on port...Processing...
Processing the new message for the client socket: 256
Something wrong happened. Closing the connection for client
Data on port...Processing...
Processing the new message for the client socket: 260
The message received from the client is : What hapenned to Client 2
*****
Data on port...Processing...
Processing the new message for the client socket: 252
The message received from the client is : Don't know
*****
```

Conclusion

- In this project we have briefly discussed the idea and formulation of the structure of a chatting application and its implementation. We have used and also described the concept of socket programming. The chatting application enables the user to do seamless interaction with another user through a server connectivity. But we are yet to add advance features like voice calling and video connectivity to our mini chat application to remove the communication barrier, making virtual connectivity more interactive. A lot has been implemented and there is a lot to improve.



Future Scope

- With messaging apps becoming more and more advanced in the real world, with the advancement of this project in the future our main focus is going to be on the speed of communication and the security of the data that is being shared between the two users.
- We will also work on adding more interactive mode of communication like video calls, voice calls, voice notes and also sharing of media, to make the communication more fun and interactive.