CHAT ROOM APPLICATION USING SOCKET.IO

A COURSE PROJECT REPORT

By

KAUSTUBH GUPTA (RA2011003010476)
PRATEEK (RA2011003010491)
AYUSH ABHIGYAN (RA2011003010493)
AFRAZ TANVIR (RA2011003010499)
SOHAM GHOSH (RA2011003010504)

Under the guidance of

Mr. G. Balamurugan

In partial fulfilment for the Course

of

18CSC302J - COMPUTER NETWORKS

in Computer Science and Engineering

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chengalpattu District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report "**Chat Room Application using socket.io**" is the bonafide work of **KAUSTUBH GUPTA (RA2011003010476), PRATEEK (RA2011003010491), AYUSH ABHIGYAN (RA2011003010493), AFRAZ TANVIR (RA2011003010491)** and **SOHAM GHOSH (RA2011003010504)** who carried out the project work under my supervision.

SIGNATURE

Mr. G. Balamurugan
Assistant Professor
Dept. of Computing Technologies
SRM Institute of Science and Technology

ABSTRACT

We will use Node.js for the backend which will create rooms and handle the backend part of our application. Additionally, we will use socket.io to enable real time communication between Client and server. Teleconferencing or chatting, is a method of using technology to bring people and ideas “together” despite of the geographical barriers. The technology has been available for years but the acceptance it was quite recent. Our project is an example of a chat server. We created a real-time chat application with room functionality where you can create your own room and share the room name with your friends to have text-based communication.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professor Dr.Annapurani Panaiyappan, Professor and Head, Department of Networking and Communications** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our Course project Faculty **Mr. G. Balamurugan, Assistant Professor, Department of Computing Technologies,** for his assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr. M. Pushpalatha, Head of Department, Department of Computing Technologies** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Definition

To make a chat room application where users can make their own room functionality where users share the room with their friends to have text-based communication.

1.2 Project Scope

Teleconferencing or chatting, is a method of using technology to bring people and ideas "together" despite of the geographical barriers. The technology has been available for years but the acceptance it was quite recent. Our project is an example of a chat server. We created a real-time chat application with room functionality where you can create your own room and share the room name with your friends to have text-based communication.

1.3 Relation To External Environment

This tool helps in too major aspects-

- List of the names of online users in the Room.
- Used for communication between multiple systems enlisted in the list

2. LITERATURE SURVEY

As we know the use of internet has increased greatly and internet has become one of the easiest and cheapest sources of communication there are many messaging and chatting applications coming up. There are already many applications available for communication. The oldest one we use is electronic mails. Other applications available are the various social websites, SMS, Mobile Chatting applications and much more.

Electronic Mails

Every day, the citizens of the Internet send each other billions of e-mail messages. If you're online a lot, you yourself may send a dozen or more e-mails each day without even thinking about it. Obviously, e-mail has become an extremely popular communication tool. The real e-mail system consists of two different servers running on a server machine. One is called the **SMTP server**, where SMTP stands for Simple Mail Transfer Protocol. The SMTP server handles outgoing mail. The other is either a **POP3 server** or an **IMAP server** , both of which handle incoming mail. POP stands for Post Office Protocol, and IMAP stands for Internet Mail Access Protocol. Whenever you send a piece of e-mail, your e-mail client interacts with the SMTP server to handle the sending. The SMTP server on your host may have conversations with other SMTP servers to deliver the email.

The chatting applications added with mailing service allow the live chat. The transfer of messages takes place within seconds. Here the numbers of peoplecommunicating are two. So the salability issue does not come into picture.

Instant Messaging

IM is the private network communication between two users, whereas a chat session is the network communication between two or more users. Chat sessions can either be private, where each user is invited to join the session, or public, where anyone can join the session. There are on the order of 100million Internet IM users, where a user is defined as a unique name on one of the major public IM networks.

A fundamental issue faced by IM service providers, and thus designers of the protocols, is how the systems will scale with large numbers of users.

Ideally, each provider desires to have millions of customers logged on to their systems at each time. This in turn requires that organizations have a system architecture that can scale with the number of users. Two approaches are available here: *symmetric* and *asymmetric*. In a symmetric architecture, each server performs identical functions, such that a client need not distinguish which server it contacts to engage in an activity with. In an asymmetric approach, each server is dedicated to a particular activity such as logging in, discovering other users on the network, maintaining a chat room, or forwarding instant message. The client-server architecture allows IM service providers to keep some degree of control over their users. On the positive side, it helps overcome some of the technical issues associated with traversing the firewalls that the clients are often behind. On the negative side, since both control and data paths go through the central servers, scaling the service to millions of users is difficult.

Social Networking Sites

The chats on social network are mainly peer-to-peer, they may happen in groups. As the chats take place in peer-to-peer they do not need to apply any queue to chat application. They use the algorithm for showing up the latest news in window and the friends available online. The friends to which we have chatted frequently are shown in the list. The newly updated news is at top on the page. The scalability of the chat is checked so that multiple chats can be carried out simultaneously. Here too the scalability issue comes in picture. As the numbers of chats are carried out simultaneously the delay time to reply the chats is not fixed. If the reply time is fixed then delay study of the scalability with time constraint is a problem faced.

3. REQUIREMENTS

3.1 Hardware Requirement

In hardware requirement we require all those components which will provide us the platform for the development of the project. The minimum hardware required for the development of this project is as follows:

- Ram- minimum 128 MB
- Hard disk-minimum 5 GB
- Processor-Pentium 3
- Floppy drive 1.44” inch
- CD drive

These all are the minimum hardware requirement required for our project. We want to make our project to be used in any. Type of computer therefore we have taken minimum configuration to a large extent.128 MB ram is used so that we can execute our project in a least possible RAM.5 GB hard disk is used because project takes less space to be executed or stored. Therefore, minimum hard disk is used. Others enhancements are according to the needs.

3.2 Software Requirements

Software requirements Software's can be defined as programs which run on our computer .it act as petrol in the vehicle. It provides the relationship between the human and a computer. It is very important to run software to function the computer. Various software's are needed in this project for its development. Which are as follows:

- Operating system-Windows 7
- Others-Visual Studio

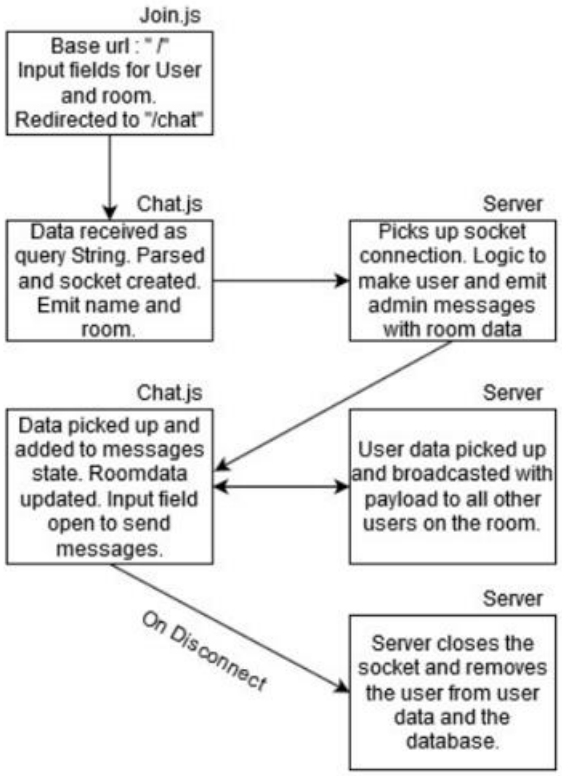
We will be using visual basic as our front hand because it is easier to use and provides features to the users which is used for the development of the project.

4. ARCHITECTURE AND DESIGN

4.1 Design Consideration

We will be using Node.js for the back end, which will be responsible for creating rooms and managing users in those rooms. Additionally, we will be using Socket.io to enable real-time, bidirectional communication between the web server and the client (browser).

4.2 Flow Chart



This represents the flow of logic and the main files. All other files are either helper functions, router or react components.

5. IMPLEMENTATION

5.1 Languages and Tools used

5.2 Methodology

- The user will interact with the tool using a GUI
- The user enters a username and a room name
- A room is created and it has a list form and a chat form
- The list form contains the usernames of the people in the room
- The chat form makes the actual communication possible in the form of text

5.3 Code Snippets of Functionalities

Joining user

```
socket.on('join', function(userName){
    console.log('user change name to : ' + userName);

    socket.userName = userName;
    users.push(userName);

    //notice it is not socket.emit('refreshUserList', users)
    socketio.sockets.emit('refreshUserList', users);
});
```

Making messages in chat

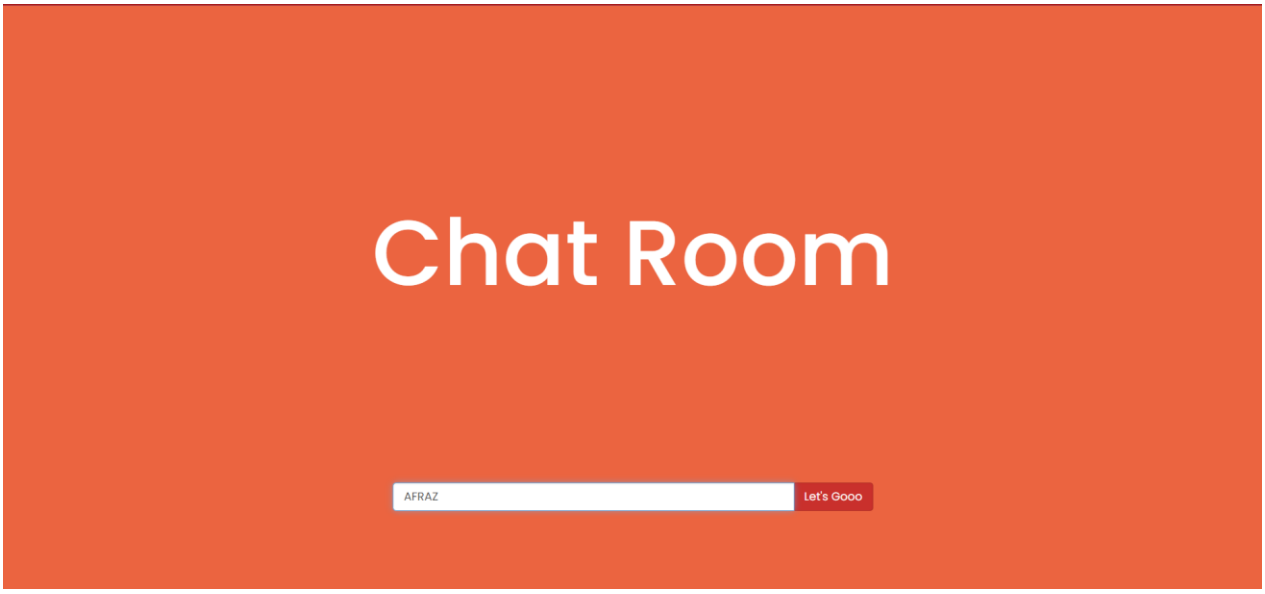
```
socket.on('message', function(message){
    console.log(socket.userName + ' says: ' + message);

    var data = {
        userName: socket.userName,
        message: message
    };

    socketio.emit('message', data);
});
```

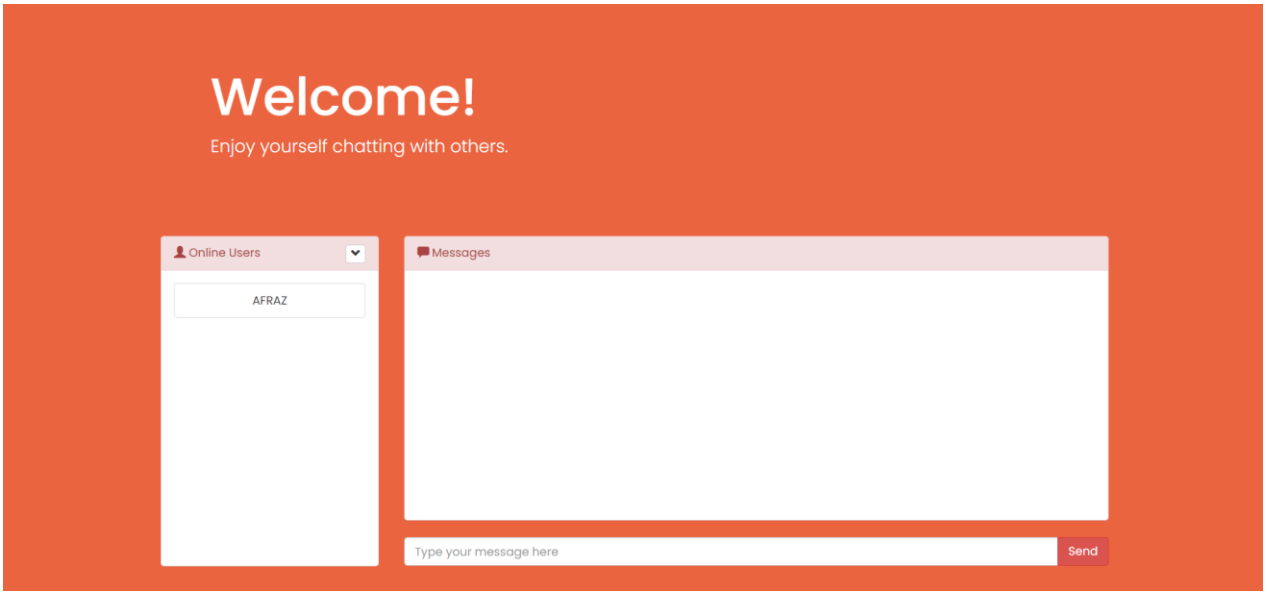
Making users Leave the chat room

```
socket.on('disconnect', function(){

    //when user log off, the name should be removed from the user list
    var removedUserIndex = users.indexOf(socket.userName);
    if(removedUserIndex >= 0){
        users.splice(removedUserIndex, 1);
    }

    //notice it is not socket.emit('refreshUserList', users)
    socketio.sockets.emit('refreshUserList', users);

    console.log('user ' + socket.userName + ' disconnected');
});
```

6. RESULTS AND DISCUSSION

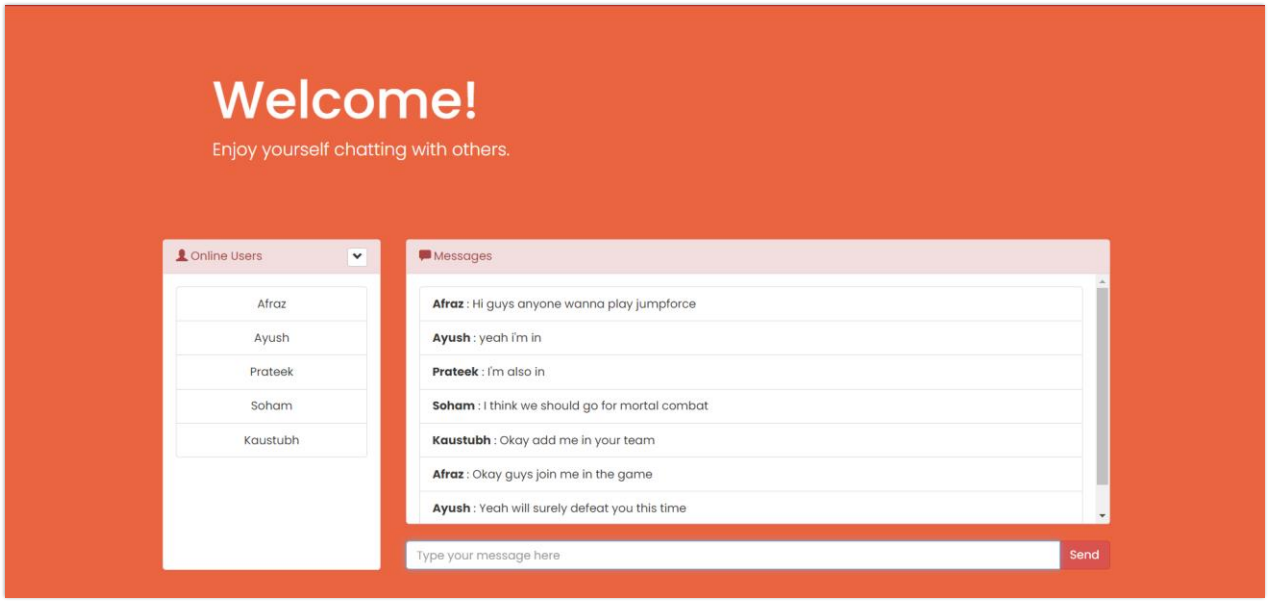
6.1 Checking if users are able to join the Chat Room





User Successfully joined the chat room

6.2 Checking if users can chat in the Chat Room




```
$ node Server/app.js
Running on PORT: 3000
A user connected.
user change name to : AFRAZ
A user connected.
A user connected.
A user connected.
A user connected.
user change name to : Ayush
user change name to : Prateek
user change name to : Soham
user change name to : Kaustubh
AFRAZ says: Hi guys anyone wanna play jumpforce
Ayush says: yeah i'm in
Prateek says: i'm also in
Soham says: I think we should go for mortal combat
Kaustubh says: Okay add me in your team
AFRAZ says: Ok guys join me in the game
Ayush says: Yeah will surely defeat you this time
user AFRAZ disconnected
A user connected.
user Ayush disconnected
A user connected.
user Prateek disconnected
A user connected.
user Soham disconnected
A user connected.
user Kaustubh disconnected
A user connected.
```

The users can join and leave the chatroom easily.

7. CONCLUSION AND FUTURE ENHANCEMENT

A chat room application where users can make their own room functionality where users can share the room with their friends to have text-based communication is constructed and deployed successfully.

In future we will try to optimize it so that it can handle a huge amount of traffic at a single time. We can roll out the feature of video chatting in the future in our chatroom.

In future, there are some issues which are faced on a greater extent and need to be worked on to improve the performance of the Chat Applications. Issues
increase the number of clients to
be processed simultaneously, increase the performance rate by reduction in time delay, agent availability, Memory leak. The online support provided by the business to its client needs to have high performance as clients prefer carrying out there work online as it saves their time of actually going to the exact location and work out there. If the service provided online is up to the mark, then it will surely benefit their business in this internet world today.

REFERENCES

- MDN Docs
- W3S Schools
- Npm docs
- Socket io.docs
- Stackoverflow
- Geeks for geeks
- Paul Goes, Noyak Ilk, J. Leon Zhao, Onadmission control policy for multi-taskingLive-Chat Service Agents.
- Arora A., Sinha M.(2012). Web ApplicationTesting: A Review on Techniques, Toolsand State of Art. In International Journal ofScientific & Engineering Research, Volume3, Issue 2, February-2012 1 ISSN 2229-5518