An Autonomous Institute
(Affiliated to Visvesvaraya Technological University, Belagavi
Approved By AICTE, New Delhi,
Recognized by UGC under 2(f) & 12(B)
Accredited by NBA and NAAC)

MINI PROJECT

ON

CHAT APPLICATION

Submitted in partial fulfillment of requirements for the award of 6th sem degree

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:

MOHAMMED KHAMEERUDDIN

1MJ20CS122

Under the guidance of

Mrs. LYNSHA HELENA PRATHEEBA HP

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MVJ COLLEGE OF ENGINEERING BANGLORE-67

2022-2023

(Affiliated to Visvesvaraya Technological University, Belagavi
Approved By AICTE, New Delhi,
Recognized by UGC under 2(f) & 12(B)
Accredited by NBA and NAAC)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

Certified that the Database Management Systems mini project entitled **"CHAT APPLICATION"** is a bonafide work carried out by MOHAMEED KHAMEERUDDIN **(1MJ20CS122)** of **6th** semester in partial fulfillment for the award of degree of **Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University**, **Belagavi**, during the year **2022-2023.** It is certified that all corrections/suggestions indicated for internal assessments have been incorporated in the Report deposited in the departmental library. The Mini Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the **Bachelor of Engineering Degree**.

Signature of Guide

Assistant Professor, CSE Dept

Signature of H.O.D

HOD, CSE

Name of the examiners

1. _____

2. _____

Signature with date

_____

ABSTRACT

The latest development of the Internet has brought the world into our hands. Everything happens through Internet from passing information to purchasing something. Internet made the world as small circle. This project is also based on Internet. This paper shows the importance of chat application in day today life and its impact in technological world.

Chat application is a feature or a program on the Internet to communicate directly among Internet users who are online or who were equally using the internet. Chat applications allow users to communicate even though from a great distance. Therefore, this chat application must be real-time and multi platform to be used by many users. The development of information and communication technologies especially to develop this chat application. This chat application in the manufacture begins with the collection of relevant data that will be displayed in the web and mobile versions. The programming language used to build server is Node.js with express framework

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

I wish to place on record my thanks to our Principal **Dr P. Mahabaleswarappa**, and Vice Principal **Dr M. Brindha**, MVJ College of Engineering for providing facilities.

I express my sincere gratitude to our **Mr.Kumar** registrar and controller for examinations, MVJCE, Bengaluru for persistent guidance.

I would also like to thank the Computer Science HOD. **Dr.Kiran Babu**, who has been constant pillar of support and guidance.

I consider it a privilege and honor to express my sincere gratitude to my guide **Mrs Lynsha Helena pratheeba HP**, assistant professor, CSE Department of Computer Science and Engineering for their valuable guidance throughout the tenure of this mini-project work and whose support and encouragement made this work possible.

I wish to extend my profound sense of gratitude to my parents, friends and all my faculty members for all the sacrifices they made during my project and providing me with moral support and encouragement whenever required.

Thanking you

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction to chat application

Today Developers around the world are making efforts to enhance user experience of using application as well as to enhance the developer's workflow of designing applications to deliver projects and roll out change requests under strict time line. Stacks can be used to build web applications in the shortest span of time. The stacks used in web development are basically the response of software engineers to current demands. They have essentially adopted pre-existing frameworks (including Java Script) to make their lives easier. While there are many, MEAN and MERN are just two of the popular stacks that have evolved out of Java Script. Both stacks are made up of open source components and offer an end-to-end framework for building comprehensive web apps that enable browsers to connect with databases. The common theme between the two is Java Script and this is also the key benefit of using either stack. One can basically avoid any syntax errors or any confusion by just coding in one programming language, Java Script. Another advantage of building web projects with MERN is the fact that one can benefit from its enhanced flexibility. In order to understand MERN stack, we need to understand the four components that make up the MERN stack A database has the following implicit properties:

Mongo DB, Express.js, React and Node. js.

1.2 Relation to External Environment

This tool helps in two major aspects –

• Resolving the names of all the system connected in a network and enlisting them.

• Used for communication between multiple systems enlisted in the resolved list.:

1.3 Methodology

The whole idea of this proposed application is to avoid a centralized system (registration, login and buddy list) as that found in Skype (Baset and Schulzrinne, 2006; Azab et al., 2012). Using Skype, during registration, user profile will be stored in a centralized database and one can use the credential to login at anytime and anywhere as preferred. This certainly provides certain level of robustness although the question arises as to how secure the centralized database can be to prevent from attacks. Recently, a study on decentralized system was proposed but only for the purposed of improving the buddy list (Kundu, 2012). The idea was about developing a robust index system using distributed hash table for decentralized chat application. An indexing system is responsible for storing IP address and port of all users once they joined the chat. Users initialize their own buddy list by contacting the centralized indexing system once they logged in. When a user A wants to communicate to user B, B will act as a server and authenticate client A. As authentication is one-way, this opens up an opportunity for attackers to masquerade as user B. To cater some of these problems, in our proposed application, we come out with the following principle ideas:.

Chapter 2

SYSTEM REQUIREMENTS AND SPECIFICATIONS

2.1 Relation to External Environment

This tool helps in two major aspects –

• Resolving the names of all the system connected in a network and enlisting them.

• Used for communication between multiple systems enlisted in the resolved list..

2.2 System specification

• List form:

In this form, all the names of the systems connected to a network are enlisted. These names can later be used for communication with the help of mouse event, or in simple language: a click or a double click.:

•Chat form

This form is called only when an element is selected from the List form.
In this form, a connection is created between the host system and the selected system with the help of a socket.

2.3 Operational Concepts and Scenarios

Operation of the application based on the inputs given by the user

List Form:

•When initialized, returns a list containing the names of all the system connected in a network.

•Contains two buttons: Refresh and Connect.

• When Refresh button is clicked refreshes the list of names.

• When the Connect button is clicked or a name is double clicked, the chat form is initialized with a connection between the host and the client machine,

Note: If no name is selected, and connect button is clicked an error box is displayed.

2.3 Dependencies

```
{
  "name": "nodeserver",
  "version": "1.0.0",
  "description": "khameer ka chatapp",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
  "prompt-sync": "^4.2.0",
    "socket.io": "^4.5.4"
  }
}
```

Chapter 3

SYSTEM ARCHITECTURE AND DESIGN

3.1 Application Architecture

SERVER

A server may be a computer dedicated to running a server application. Organizations have dedicated computer for server application which has to be maintained periodically and has to computers accept clients over network connections that are requested. The server responds back by sending responses being requested. There are many different server applications that vary based on their dedicated work. Some are involved for accepting requests and performing all dedicated works like business application servers while others are just to bypass the request like a proxy server. These server computers must have a faster Central processing unit, faster and more plentiful RAM, and bigger hard disc drive. More obvious distinctions include redundancy in power supplies, network connections, and RAID also as Modular design.

It should be monitored continuously for traffic loads would never let them go down which affects the company's revenue. Most organizations have a separate monitoring system to keep an eye over their server so that they can find their server downtime before its clients. These server

CLIEENT

A client is a software application code or a system that requests another application that is running on dedicated machine called Server. These clients need not be connected to the server through wired communication. Wireless communication takes place in this process. Client with a network connection can send a request to the server.

3.2 FLOW DAIGRAM

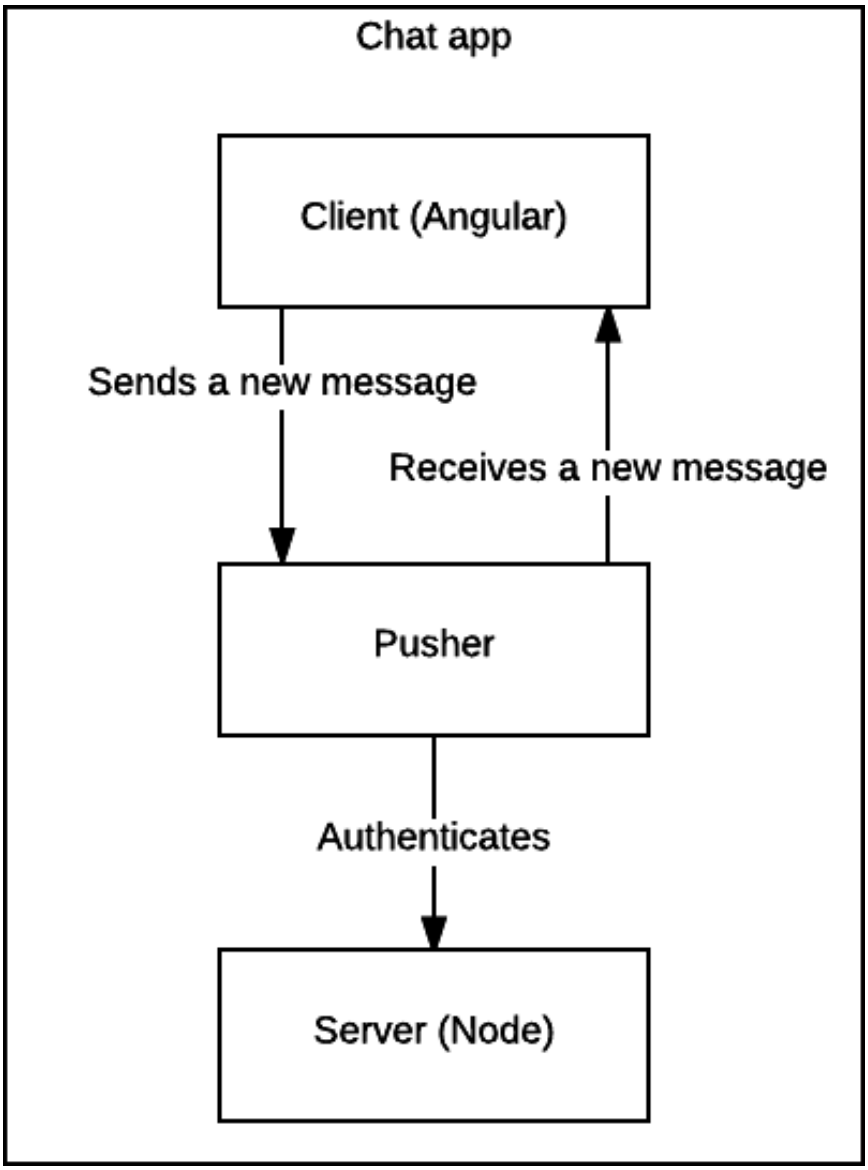

Fig-4.1.4

Chapter 4

SYSTEM IMPLEMENTATION AND DEPENDENCIES

4.1 Implementation

Implementation is a vital step in ensuring the success of new system even a well designed system can fail if it is not a properly implemented. Implementation activities are needed to transform a newly developed information system into an operational system for end users.

- **Acquiring Hardware Software And Services:**

These resources acquired from many sources in the computer industry. Some sources are as follows a-hardware- IBM, HP, Apple computer etc. b-software- Microsoft, Oracle etc..

4.2 Component:

Components are the building blocks of any React app and a typical React app will have many of these. Simply put, a component is a JavaScript class or function that optionally accepts inputs i.e. properties(props) and returns a React element that describes how a section of the UI (User Interface) should appear.

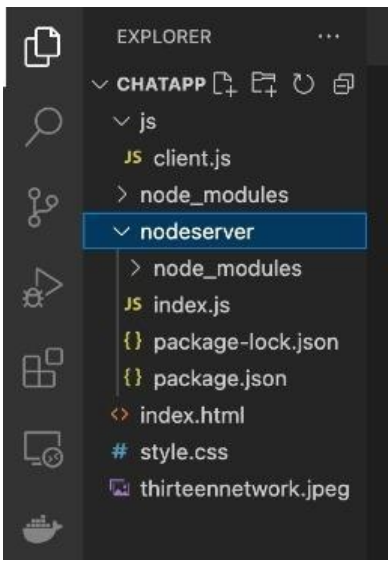

Fig-4.1.4

4.3 Chat Server Engine

This is a core of the chat architecture that handles message delivery and dispatch. In our version of chat architecture, it includes the following components

- **Chat WebSocket Server**

It is responsible for transmitting messages between users. The Chat App communicates with the Chat WebSocket Server via the Chat WebSocket Client Library. This connection is open two ways; that means users don't have to make requests to the server if there are any messages for them, they just get them right away

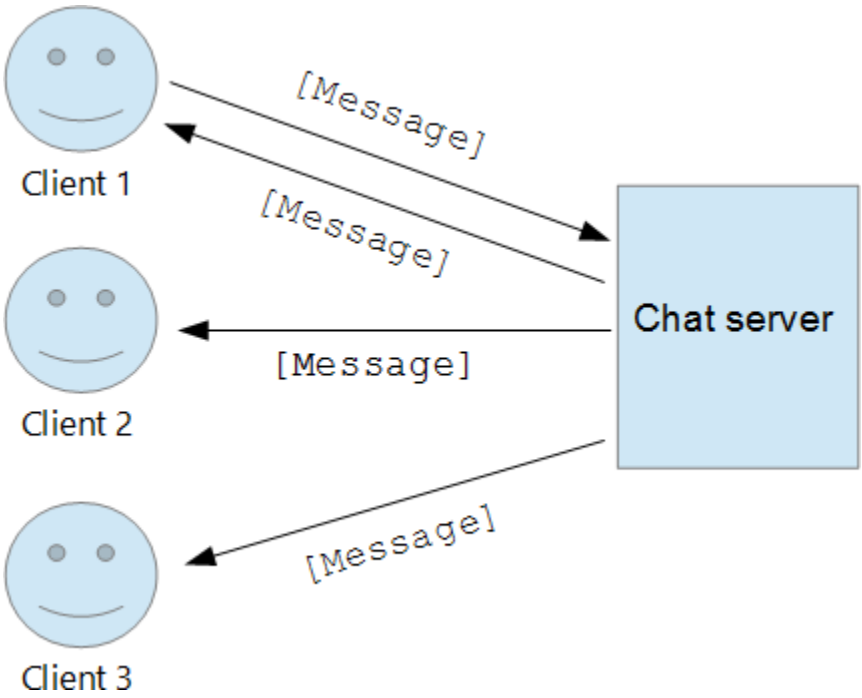

Fig-4.1.4

4.4 literature survey

1. "Online Chat App" was published by the International Journal of Research in Engineering

Published in : Oct 2019
Authors : Jhalak Mittal , Arushi Garg and Shivani Sharma .

- This study presents an architecture for building a scalable chat application using distributed systems and message queues, emphasizing fault tolerance and high availability.

2. Design and Implementation of a Scalable Chat Application

Published in : Jan 2020
Authors : Smith et al.

- Chat applications, focusing on various aspects such as architecture, security, user experience, and emerging trends.

- It mainly focuses on Design and architechture

3. Security Challenges in Chat Applications

Published in : July 2021
Authors :Zhang and Wang.

- This research proposes a secure chat application that employs end-to-end encryption
- to ensure message confidentiality, integrity, and authenticity, while mitigating the risk of unauthorized access.

4.Emerging Trends and Technologies

Published in : dec 2021
Authors : Rahman et al

- This paper reviews the integration of artificial intelligence (AI) technologies, including natural language processing, machine learning, and chatbots.

Chapter 5

SCREENSHOTS

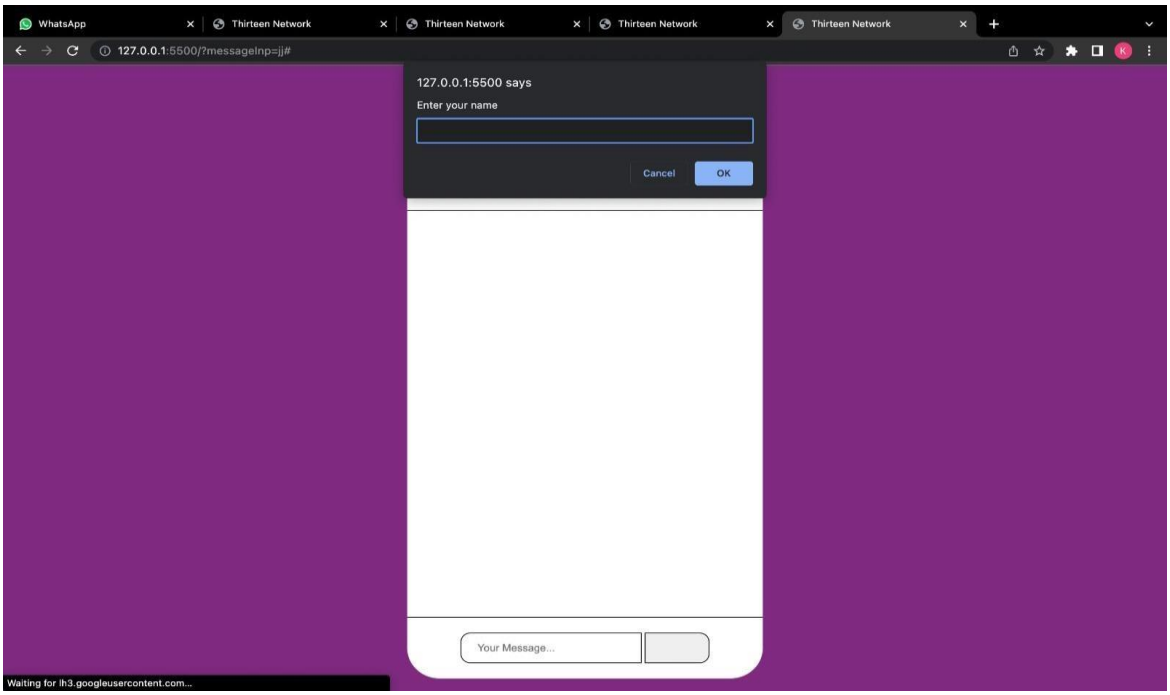

Fig-4.1.4

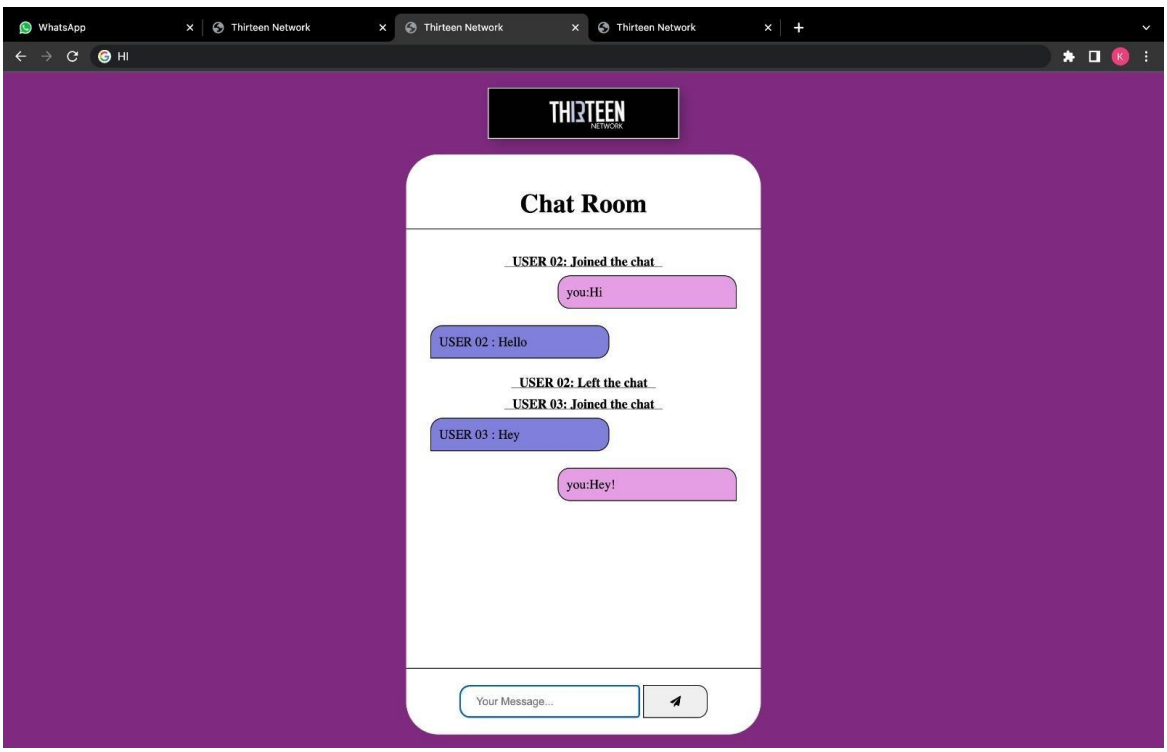

Fig-4.1.4

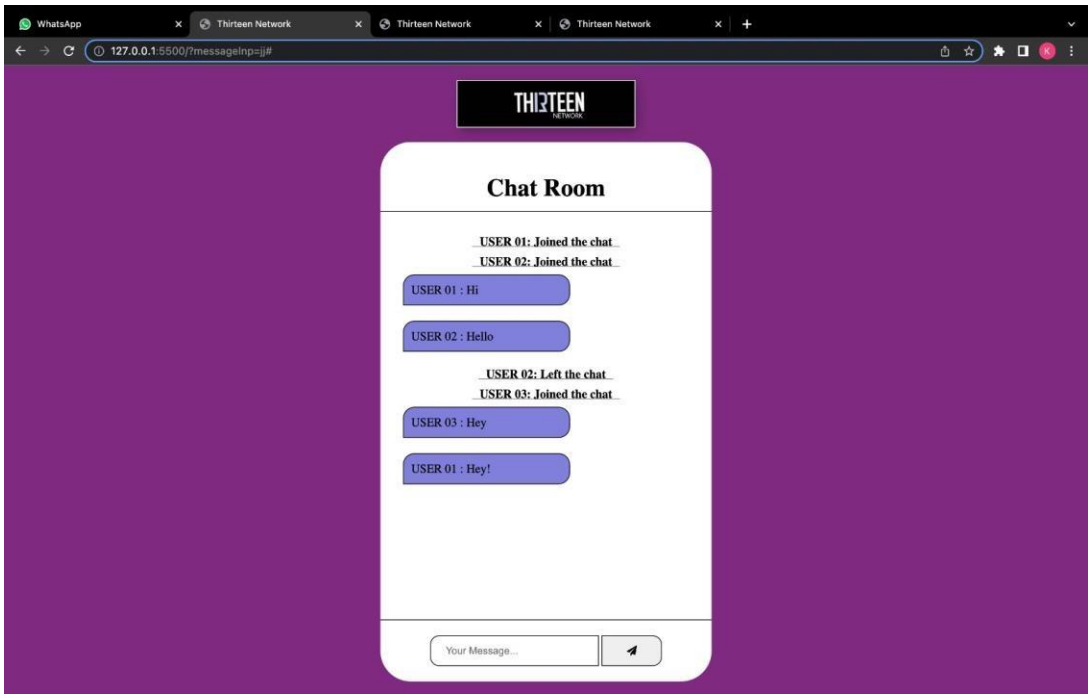

Fig-4.1.4

Chapter 6

DESCRIPTION

6.1 DESCRIPTION

1. A static Server socket is created in beginning which is then bind with host and port .

2. After server instantiation Socket in particular host, it begins to listen in the particular port. Then the server is made to accept the request from the client through the particular port.

3. After starting the server, it can accept the requests from clients.

4. The socket is instantiated in client side to connect to the server.

5. A new Server Thread using socket is created to accept all the requests from multiple clients.

6. After accepting the request both read and write operation occurs simultaneously, clients who request the server can communicate with each other and share resources.

6.2 Client program

Client.js

```
const socket = io('http://localhost:8000');
const form = document.getElementById('send-container');
const messageInput = document.getElementById('messageInp')
const messageContainer = document.querySelector(".chatcontainer")
const append = (message, position) => {
const messageElement = document.createElement('div');
messageElement.innerText = message;
```

```
messageElement.classList.add('message')
    messageElement.classList.add(position);
    messageContainer.append(messageElement);
}

form.addEventListener('submit', (e) => {
    e.preventDefault();
    const message = messageInput.value;
    append(`you:${message}`, 'right');
    socket.emit('send', message);
    messageInput.value = ''
})

const kname = prompt("Enter your name");

socket.emit('new-user-joined', kname);

socket.on('user-joined', kname => {
    append(`${kname}: Joined the chat`, 'center')
})

socket.on('receive', data => {
    append(`${data.kname} : ${data.message}`, 'left')
})

socket.on('left', kname => {
    append(`${kname}: Left the chat `, 'center')
})
```

6.3 Server programs

Server.js

```
const io = require('socket.io')(8000, {
cors: {
origin: "*"
}
});
const users = {};


io.on("connection", (socket) => {
socket.on('new-user-joined', kname => {
console.log("new user", kname)

 users[socket.id] = kname;
socket.broadcast.emit('user-joined', kname);
});

socket.on('send', message => {
socket.broadcast.emit('receive', { message: message, kname: users[socket.id] });
});

socket.on('disconnect', () => {
socket.broadcast.emit('left', users[socket.id]);
delete users[socket.id]
});
```

Chapter 6

CONCLUSION

There is always a room for improvements in any apps.
Right now, we are just dealing with text communication. There are several chat apps which serve similar purpose as this project, but these apps were rather difficult to use and provide confusing interfaces. A positive first impression is essential in human relationship as well as in human computer interaction. This project hopes to develop a chat service Web app with high quality user interface.

In future we may be extended to include features such as:

- File Transfer .
- Voice Message
- Video Message .
- Audio Call
- Video Call
- Group Call

REFERENCES

- https://www.researchgate.net/publication/360483603_Research_paper_on_Group_chatting_Application
- https://www.ijrti.org/papers/IJRTI2206316.pdf
- https://www.w3schools.com/nodejs/
- https://socket.io/