MAY 2022

PCS II COURSE PROJECT

CHAT APPLICATION - PROJECT REPORT

Group Members:
Harsh Kumar (B20AI011)
Shubham Kumar (B20AI039)
Tanmay (B20AI047)

GitHub: https://github.com/tanmayiitj/PCS-II-Project-Chat-Application

ABSTRACT

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.In this project, we have created a realtime chat application which have features like text messaging, speech recognition and new message notifications.

PROBLEM STATEMENT

Based on the description above, the problem is how to design and build a dedicated web-based chat applications in real time to make share and communicate easier and user friendly.

OBJECTIVE

To make user friendly chat application on which people can chat with their friends.
To know about various aspects of node.js and socket programming.

PROJECT DESIGN DETAILS

In this project, we have created a realtime chat application which have multiuser interface connected by server. In this we have used SocketIO which enables bidirectional communication. It creates web socket between client and server with which both can communicate with each other.

FRONT-END:

- At frontend, we are mainly using **HTML** and **CSS**.

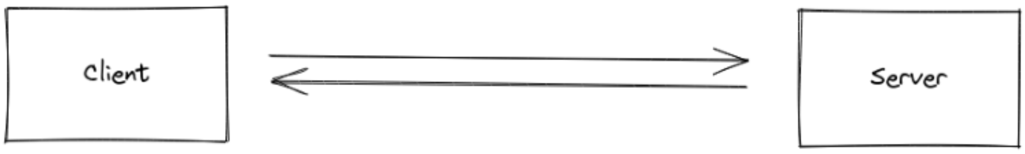
BACK-END:

- At backend, we are mainly using **JavaScript**, **NodeJs** and **SocketIO**.

What is Socket.IO?

- Socket.IO is a library that enables low-latency, bidirectional and event-based communication between a client and a server.

- It is built on top of the WebSocket protocol and provides additional guarantees like fallback to HTTP long-polling or automatic reconnection.



IMLEMENTATION STEPS

1.**Create a Directory & Install Dependencies**

We need an environment to house our chat server (back end) and our chat client (front end), so the first step in the development process is to create a directory and install the required dependencies like node.js

2.**Build the Front-End Chat Interface & Chat Client**

We built our basic structure of front-end UI using HTML and then we used CSS for designing. Chat interface contains a scrollinng message display screen and a text input box with send botton.

3.**Create and Connect the Back-End (Chat Server)**

The chat server is responsible for routing messages from sender to receiver and for administering other back-end functions that won't be stored locally on the user's device Our prefered back-end language is Node.Js(Javascript).Our chat server will need to listen for new messages, render existing messages already in the channel, and push new messages from sender to receiver.

CONCLUSION AND FUTURE SCOPE:

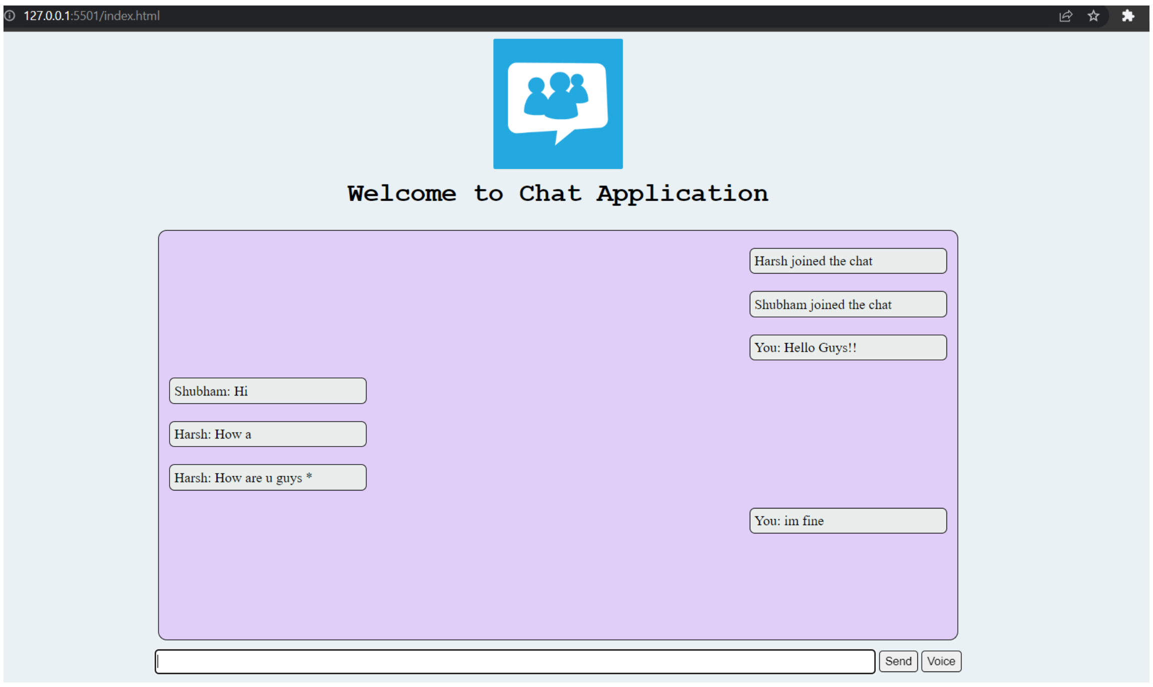
There is always a room for improvements in any software package, however good and efficient it may be done. But the most important thing is, it should be flexible enough to accept further modifications.

Right now we are just dealing with text communication. In future this software may be extended to include features such as:

1. **FILE TRANSFER**: This will enable the user to send files of different formats to others via the chat application.
2. **VOICE CHAT**: This will enhance the application to a higher level where communication will be possible via voice calling as in telephone.
3. **VIDEO CALL**: This will increase the user experience multifold on the app.

05

WEBSITE SCREEN-SHOTS



Welcome to Chat Application

Harsh joined the chat

Shubham joined the chat

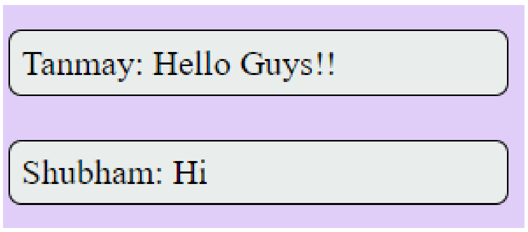
You: Hello Guys!!

Shubham: Hi

Harsh: How a

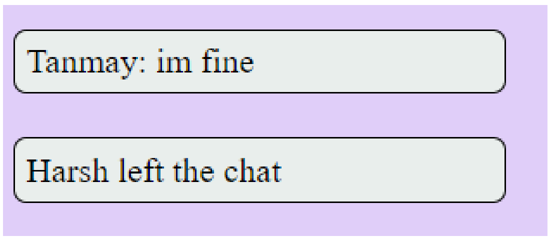
Harsh: How are u guys *

You: im fine



Tanmay: Hello Guys!!

Shubham: Hi



Tanmay: im fine

Harsh left the chat

CONTRIBUTIONS

Harsh Kumar (B20AI011): Back-end, Report and readMe

Shubham Kumar (B20AI039): Front-end, Back-end and readMe

Tanmay (B20AI047): Front-end, Report and Back-end

REFERENCE

Socekt.io Documentation: https://socket.io/docs/v4/

Speech Recognition: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API

Nodejs: https://www.youtube.com/watch?v=Oe421EPjeBE&ab_channel=freeCodeCamp.org