

Video Chatting Web App using WebRTC

by

| | |
|---------------------------|-----------|
| Rufus Sam John Immanuel J | 19BLC1031 |
| Shrimayee Deshpande | 19BLC1034 |
| Sunil Kumar GV | 19BLC1055 |

A project report submitted to

Dr M. Saranya Nair

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

ECM2001 – DATA COMMUNICATION NETWORKS

in

B.Tech. ELECTRONICS AND COMPUTER ENGINEERING



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

VIT CHENNAI

Vandalur – Kelambakkam Road

Chennai – 600127

December 2021

BONAFIDE CERTIFICATE

Certified that this project report entitled “**VIDEO CHATTING WEB APP USING WEBRTC**” is a bonafide work of **RUFUS SAM JOHN IMMANVEL J (19BLC1031), SHRIMAYEE DESHPANDE (19BLC1034) AND SUNIL KUMAR GV(19BLC1055)** who carried out the Project work under my supervision and guidance.

Dr M. Saranya Nair

Assistant Professor (Sr)

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

Due to the pandemic, the whole education system has shifted to an online mode of education. Many working professionals also prefer the ‘work from home’ system. Due to this there has been a massive increase in the use of conferencing apps. Some of the major companies like Microsoft, Google, Zoom, Cisco etc have developed video voice and text conferencing applications which are being extensively used from the past few years in numerous ways.

The main objective for developing this project is to understand what goes on in the backend of these applications and develop a model ourselves. The focus of our project has majorly been on understanding the webRTC architecture. WebRTC helps us in establishing transmission of audio and video data. Furthermore, we understood how the communication is established between two users.

We implemented an application which can be used for video and text conferences between two users. We added features like screen sharing, recording etc for the video conference. We also added a feature where people can connect to random users if they want to, just for entertainment. We also hosted the app. Developing this application was a great learning experience.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr. M. Saranya Nair, Assistant Professor (Sr.), School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to Dr. A. Sivasubramanian, Dean of the School of Electronics Engineering (SENSE), VIT Chennai, for extending the facilities of the School towards our project and for her unstinting support.

We express our thanks to our Programme Chair Dr. D. Thiripurasundari for their support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

RUFUS SAM JOHN IMMANVEL J

SHRIMAYEE DESHPANDE

SUNIL KUMAR GV

TABLE OF CONTENTS

| S. NO. | CONTENT | PAGE NO. |
|-------------------|-----------------------------------|---------------------|
| | ABSTRACT | 3 |
| | ACKNOWLEDGEMENT | 4 |
| 1 | INTRODUCTION | 6 |
| 1.1 | RELATED WORK | 6 |
| 1.2 | OBJECTIVES AND GOALS | 6 |
| 1.2 | FEATURES | 6 |
| 1.3 | APPLICATIONS | 6 |
| 2 | PROPOSED DESIGN | 7 |
| 2.1 | FLOW CHART | 7 |
| 2.2 | REQUIREMENTS | 8 |
| 2.3 | IMPLEMENTATION METHODOLOGY | 9 |
| 2.4 | RESULTS OF IMPLEMENTATION | 9 |
| 2.5 | TRADE OFFS & COST ANALYSIS | 10 |
| 3 | CONCLUSION AND FUTURE WORK | 11 |
| 3.1 | CONCLUSION AND INFERENCE | 11 |
| 3.2 | FUTURE WORK | 11 |
| 4 | REFERENCES | 11 |
| 5 | YOUTUBE LINK | 11 |

1. INTRODUCTION

1.1 RELATED WORK

There are many video conferencing applications like MS Teams, Google Meet, Zoom conference etc which are prominent. They have several features and are used for different purposes. Due to the pandemic, the use of these platforms has widely increased.

1.2 OBJECTIVES AND GOALS

The main objective of this project is to learn what happens in the backend of these conferencing applications. To understand how the communication is established between users, how the video and audio data is transferred.

Our main goal is to develop a basic application which serves video calling functionality.

Additionally we also need a provision for text conferences between two people. Furthermore, improvise the application by adding additional features.

1.3 FEATURES

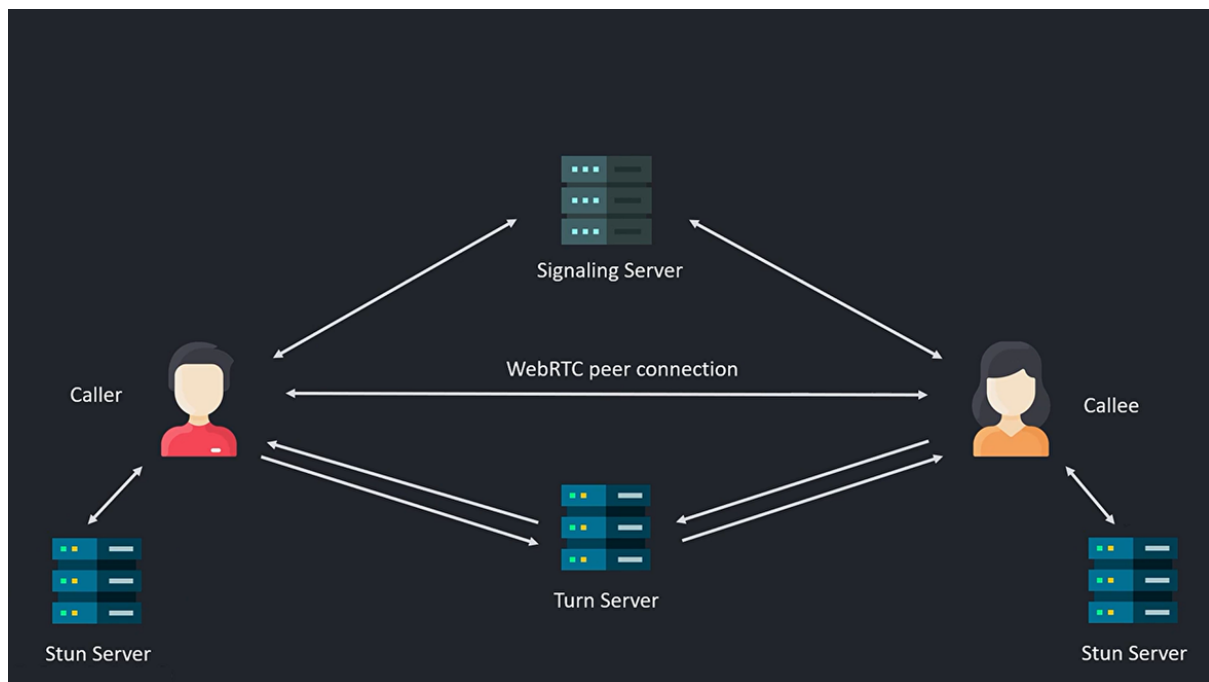
1. Audio and Video Communication
2. Screen sharing
3. Built - in text message chat box in meeting
4. Meeting can be recorded
5. Just a chat call (only text chat) can be made.
6. Connecting to a new person via the 'Connect with stranger' option.
7. High quality video streaming
8. Dynamic Screen sharing options

1.4 APPLICATIONS

1. During pandemic situations, it can be used as an **online school** for conducting meetings.
2. **Chat with friends** in a free video call.
3. It can be also used as an **online recorder** as it has recording capability.

2. PROPOSED DESIGN

2.1 FLOW CHART

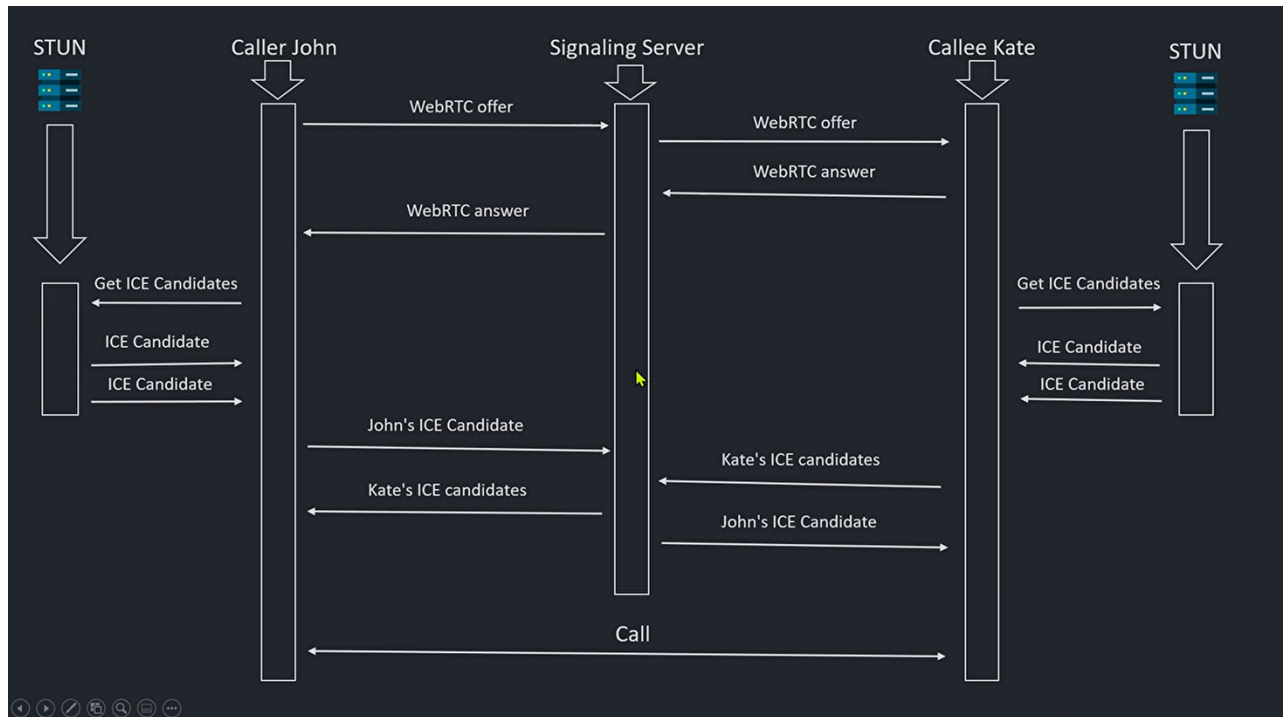


Step 1: Signaling: both peers connect to a signaling server (using websockets over socket.io) and exchange information (about their media capabilities, public IP:port pairs when they become available, etc.)

Step 2: Discovery: Devices connected to LAN or mobile networks are not aware of their public IP (and port) where they can be reached at so they use STUN/TURN servers located on the public Internet to discover their ip:port pair (ICE candidates). In the process they punch a hole through the NAT/router which is used in step3.

Step 3: P2P connection: once the ICE candidates are exchanged through the initial signaling channel each peer is aware of each other's ip:port (and holes have been punched in NATs/routers) so a peer to peer UDP connection can be established.

The above mentioned three steps are shown as a sequential diagram for better understanding.



2.2 REQUIREMENTS

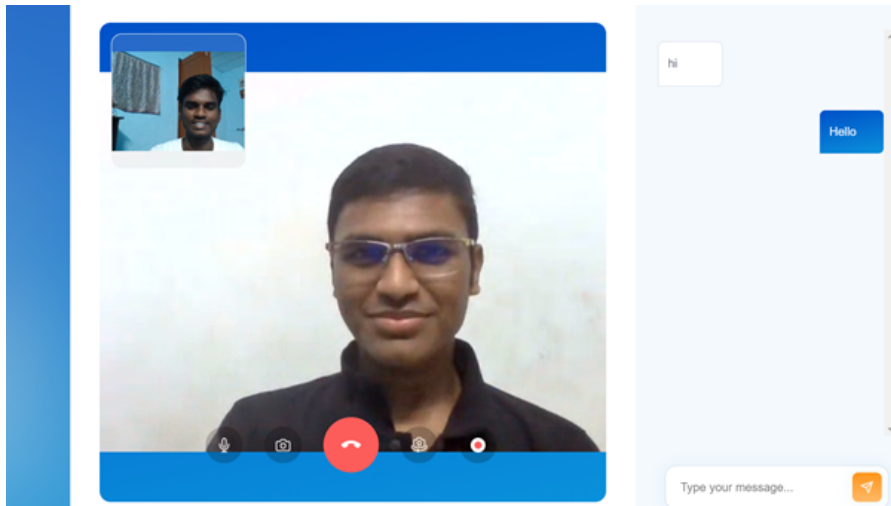
Since it is a software project, all requirements are software based.

1. NodeJS
2. ExpressJS
3. Socket.io
4. Media recorder API
5. Web browser to run the webapp
6. Front - end tools like HTML, CSS and JavaScript

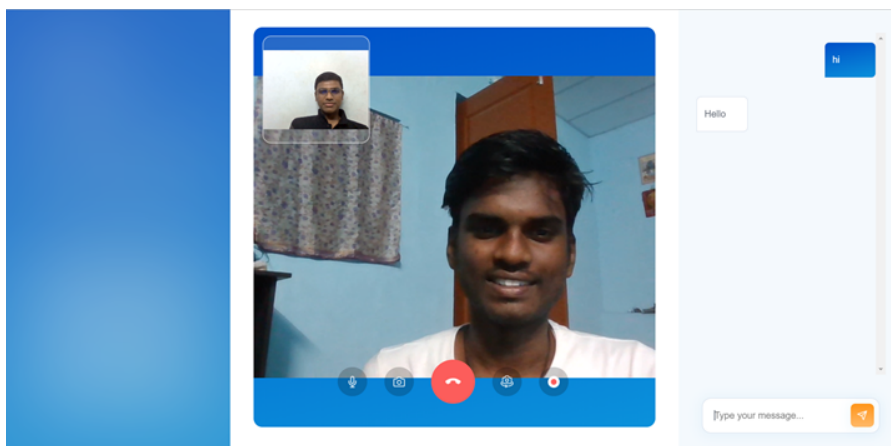
2.3 IMPLEMENTATION METHODOLOGY

- Prepare environment
- Build frontend of our application with React
- Connect with SocketIO and creating necessary logic
- WebRTC implementation for group calls
- Add WebRTC data channels for chat functionality
- Add functionality to connect only with audio
- Get TURN server credentials
- Deploy application

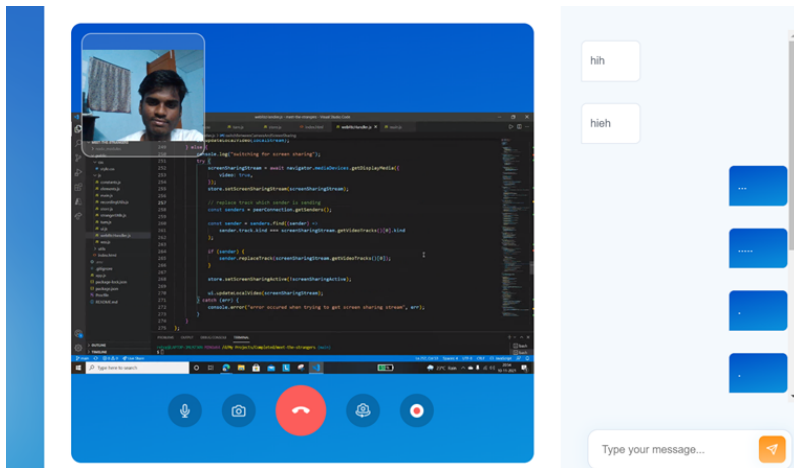
2.4 RESULTS OF IMPLEMENTATION



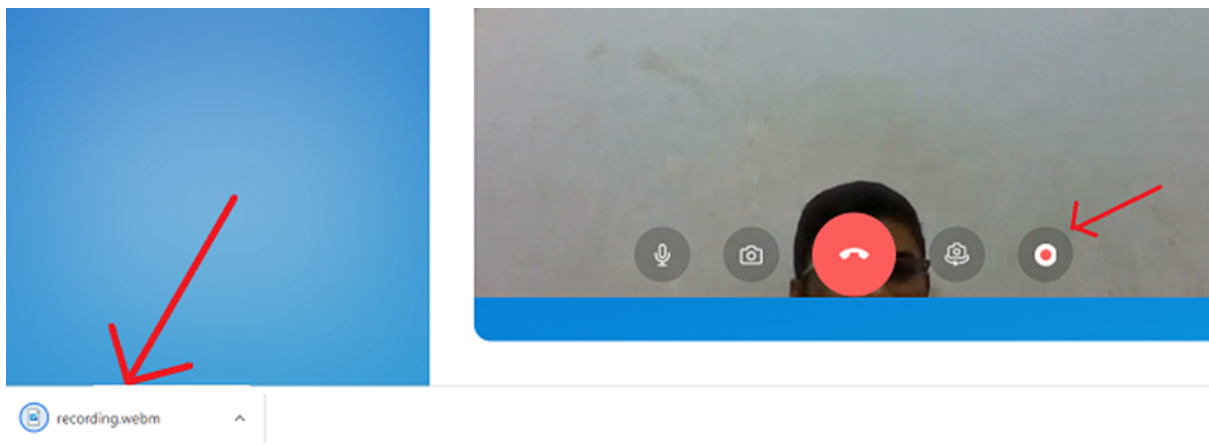
Caller 1 side



Caller 2 side



Screen Sharing



Recording

2.5 TRADE OFFS & COST ANALYSIS

1. As the Number of participants increases, the video quality decreases to maintain a good latency.
2. When IP address couldn't be found we need to make use of turn servers. Turn server connection costs a lot of money.

3. CONCLUSION AND FUTURE WORK

3.1 CONCLUSION & INFERENCE

Until now, we have successfully established communication between two users, for both video and text calls. We have implemented the caller-callee communication network using WebRTC. We also hosted the webapp using Heroku for remote communication.

3.2 FUTURE WORK

1. For further work, implementation of a multi-user interface so that multiple users can join in the same room by creating a mesh structure.
2. We can also improve the user interface of the webapp.
3. Adding additional features like linking a whiteboard, adding backdrop to the videos etc.

4. REFERENCES

1. https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API
2. https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Protocols
3. https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Build_a_phone_with_peerjs
4. [Build a Real Time Chat App With Node.js And Socket.io - YouTube](#)
5. [Stack Overflow - Where Developers Learn, Share, & Build Careers](#)

5. YOUTUBE LINK

<https://youtu.be/r8aaMbQ8DdA>

6. WebSite Link:

group-5-dcn.herokuapp.com