

---

## KONNECT+: INFINITE VIDEO CONFERENCING SOFTWARE

**Kirti Pawar<sup>\*1</sup>, Durgacharan Nayak<sup>\*2</sup>, Sakshi Pawar<sup>\*3</sup>**

<sup>\*1</sup> Bachelor of Engineering, Department of Information Technology, Vasantdada Patil Pratishthan's College of Engineering & Visual Arts, Mumbai, Maharashtra, India.

<sup>\*2</sup> Bachelor of Engineering, Department of Information Technology, Vasantdada Patil Pratishthan's College of Engineering & Visual Arts, Mumbai, Maharashtra, India.

<sup>\*3</sup> Bachelor of Engineering, Department of Information Technology, Vasantdada Patil Pratishthan's College of Engineering & Visual Arts, Mumbai, Maharashtra, India.

---

### ABSTRACT

In the recent past, video conferencing has emerged as one of the most demanding technology features. It has become even more important tool to overcome the communication gap when physical presence is not possible or not important. In today's world, almost every industry uses video conferencing as it is increasingly popular way of communicating, for example classes and lectures are taken online for distance learning, companies hiring virtually. One of the major reasons for the rapid growth of video conferencing is that it saves a lot of time and money. Basically, for two or more people to communicate, all that matters is the audio and video of one another. So, video conferencing uses audio and video for real time communication to connect people together. With the advancement of bandwidth and network services, the resources are now available to provide better interaction in the virtual classroom via video conferencing. Due to this, people from the education sector are able to provide a better interactive session in the virtual classroom. Distance learning is now achieved.

**Keywords:** Video real time communication, Video conference, Screen sharing, Distance learning.

---

### I. INTRODUCTION

The aim of video conferencing is to create an experience which is comparable to face-to-face interactions. Nothing beats a face-to-face communication therefore advancements are being done in this field. Almost everything is advancing towards digitalization, this proposes a strong reason to improve towards video conferencing. So the main task is to make video conferencing create an experience such that one prefers it over face-to-face communication. Face-to-face meeting has been acknowledged as the best way for education, business, meetings etc. The reason is face-to-face communication allow people to pass their thoughts clearly not only through voice but also expressions and gestures. The world and technology are developing rapidly, so to keep up with the requirements, some changes are necessary. Sharing files dynamically with peers in the meeting is one of the biggest advantages. Now this leads to the question on how to implement these functionalities. Basically, the core part is computer networking. Already existing video conferencing applications offer these features using computer networking as the core. Further extending the thoughts and ideas, sockets can be used and manipulated to even share files at run time. Other functionalities remain same like switching audio on or off, video on or off, screen sharing features and chat feature. An interesting idea is to give user a little more freedom in the sense as to whom the user wants to share these messages and files. This is a powerful feature which can be used very effectively in education sector where teachers want to check current status of each and every student's work.

### II. REQUIREMENT GATHERING AND ANALYSIS

#### II.1 Eliciting Requirements:

##### PRODUCT:

Due to pandemic, online classes and meeting was the only solution due to which students faced a lot of problems in attending them and were in need of a platform with multiple features of not only videoconferencing, chat and screen sharing to attend classes, but also sharing files via the chat itself where from where they can share their work.

Having easy interactive features like nodding helps teachers as well in their work like attendance and also in full screen mode while screen sharing to interact. Also, a private message/File to host/teacher will help to interact with individual student. Even teachers needed an easy to use and multi-featured UI for teaching.

#### II.2 Implementation:

Basic objective was to create an interface for face-to-face communication using deoconferencing. Noise reduction/suppression, Session setup, Echo cancellation, Error concealment, Wideband codecs (voice and video), Dynamic jitter buffers, P2P protocols, Automatic Gain Control, Network Traversal are needed for video

conferencing. This makes it difficult to manage. WebRTC makes it much easier to use. Real-Time Communications is achieved using WebRTC.

The whole implementation can be divided into following steps:

In html page, a video element should be set up.

1. Local devices should be accessed like camera, speaker, microphone.
2. From a remote or local peer, a/v should be displayed.
3. Create a new meet room
4. Connect to remote peers
5. Screen Share and File Share
6. Chat
7. Nodding

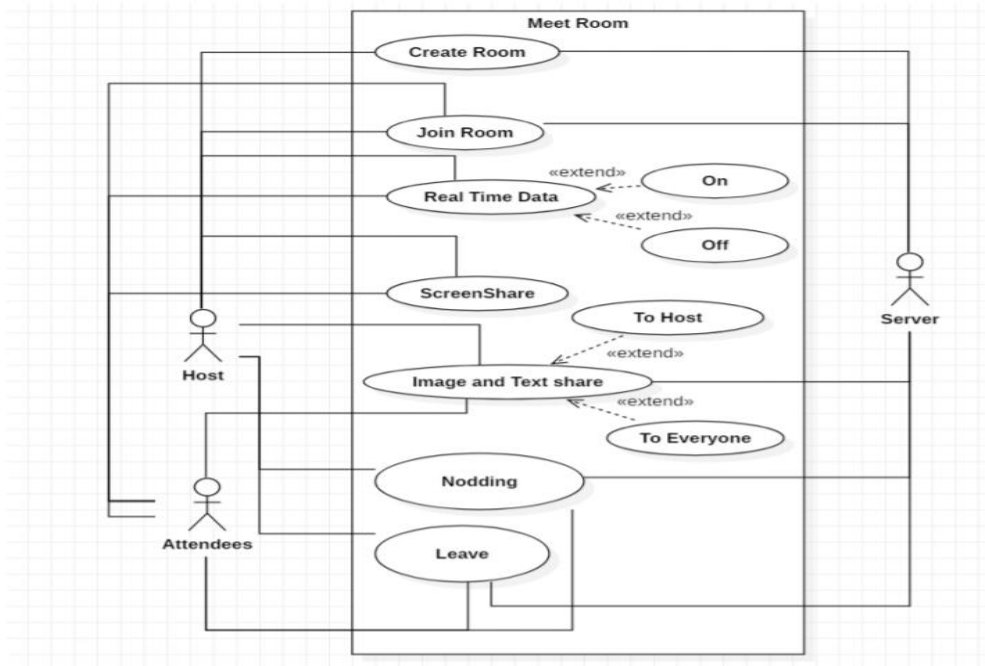


Figure 1: Use case diagram.

### II.3 Analysing Requirements:

One of the major problems is that every existing application follows a standard set of features. The world and technology is developing rapidly, so to keep up with the requirements, some changes are necessary. Sharing files dynamically with peers in the meeting is one of the biggest plus. Giving user a little more freedom in the sense as to whom the user wants to share the messages and files and this can be used very effectively in education sector where teachers want to check current status of each and every student’s work.

UX Design Features:

1. User Friendly Interface and Easy to use features.
2. The documentation is also provided for easier understanding of the functionalities.
3. All features like video/audio on/off, screen share and nodding buttons available on a single panel, a single click away.
4. A direct select radio button to send message to either host or everyone.
5. Image is uploaded directly from device in chat box.
6. By typing in the chat box and pressing enter will send messages instantly.
7. Is a browser based application hence, easily manageable and accessible.

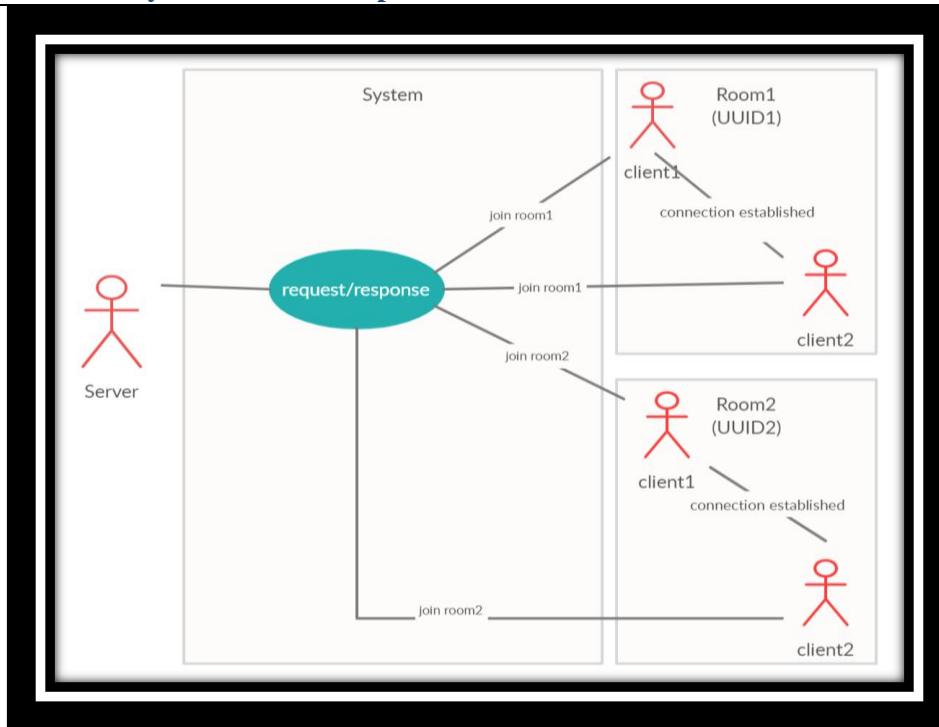


Figure 2: Architecture diagram.

### III. TECHNOLOGIES USED

#### III.1 WebSocket and Socket.IO

A full-duplex channel over a single TCP connection is possible due to WebSocket and by this means, a connection will be established by the client. If connection is not established then the client will fall back on HTTP long polling if not. Because of WebSocket, communication between client and server is possible in real time. Once the client and server are connected, they are able to share data with each other inside the same connection. In the application we are using socket.io which is library that wraps WebSocket. WebSocket adds additional metadata to each packet due to which a WebSocket client and socket.io client can never connect to each other. The client and server are able to establish a bi-directional communication because of Socket.IO. The client should have Socket.IO in the browser and a server should integrate a Socket.IO to start a bi-directional communication. JSON is used to send data in many and simplest forms. Engine.IO is used by Socket.IO to exchange data and to establish a connection. For client, Engine.IO-client is used whereas for server implementation, Engine.IO is used.

#### Emitting Events of Socket.IO:

Send and receive events - with any data.

Objects that can be encoded as JSON and binary data is supported.

Example:

The EventEmitter class is extended on both the sides by socket object therefore:

- socket.emit() is used to send an event
- socket.on (<event name>, <listener>) is used to receive an event.

on() is a Socket.IO function.

The arguments associated with on() are:

The name of the event

a callback which will be executed after every connection event

A socket object is returned by connection event. The callback function will pass it.

#### Reliability of Socket.IO:

Connections can also be established in the presence of antivirus software, personal firewall, load balancers and proxies. Engine.IO upgrades to a better transport by achieving a long-polling connection first.

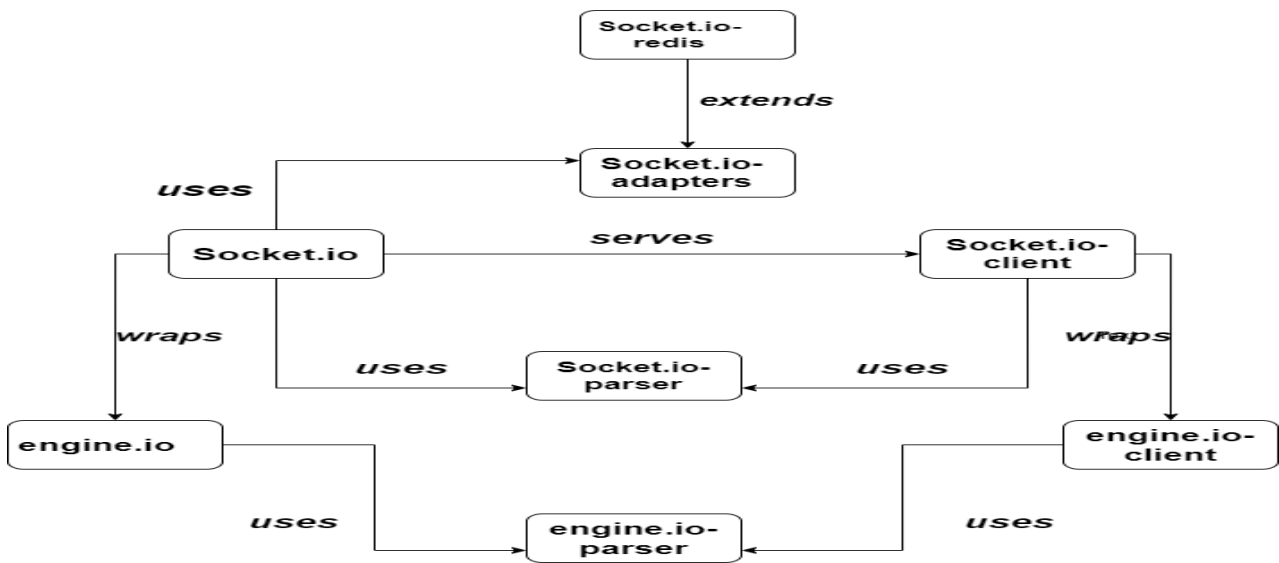


Figure 3: Reliability of Socket.io

### III.2 EXPRESS

Express is a minimalistic web framework which is un-opinionated and also fast for Node.js.

Following are the tasks performed by middleware functions:

- Any code can be executed.
- Response objects and request objects can be changed.
- Request-response cycle can be ended.
- From the stack, the next middleware function can be called.

Application-level middleware: In our project we have used express as a Application-level-middleware that is from our client (host and other participants) requests a new room or join a room and website sends request to the express server and returns response to website and from website to our client(host and other participants). By using app.Method() and app.use(), the instance of an app and application level middleware can be bind.

### Basic Concept

As a web server, Node.JS and Express frameworks can be used. Once landing page is loaded the user has an option to either host a meet or to join an existing meet. If a user requests to host a meet, a new room is created identified by a unique ID generated by UUID library. When a new user joins a meet with a valid room ID it is allocated a unique user ID by PeerJS in order to differentiate between all users and to ease the communication. It is then emitted back to the server through socket to keep a track and store the user details like room ID, user ID and user-name in the JSON object, after validation of a room ID the new user is connected to the socket and room.

### III.3 Chat Functionality

The chat functionality is implemented using Socket.IO. The user has an option to either communicate through chat with everyone or privately with the host. User can also communicate with either text or image format. The main challenge is to transfer large images considering low bandwidth. To overcome the same, when user uploads an image to be shared it is first compressed according to threshold value of size and then encoded using base64 format. According to the selected options all required values like type of message, sender ID, message and choice to send media to either host all participants is stored in JSON object and emitted to the socket. It is collected by the socket at the server side and it checks from which user and room the message is received. If it is to be send to all the users, an iterative loop is executed to emit the JSON object which contains all required parameters to each user present in that room with the help of user IDs which are already stored at server side. All host IDs are also stored separately to speed up direct emitting of message to host. When a user receives a message from server-side socket event it displays message on chat window according to its format.

### III.4 Base64 Encoding

When the binary data (videos, images) is converted to 6-bit character representation so as to convert it into ASCII (plain-text) string format, Base64 encoding takes place.

### III.5 Nodding

The Nodding function for the user emits the selected 'yes' or 'no' value as a JSON object to the socket. The socket event at the server side receives the JSON object along with the user ID and in response it emits the value to the host which is displayed as browser notification. The notification is visible in full screen mode during screen share.

### III.6 WebRTC

It is a short form of web real time communication. A very naive approach of communicating with peers can be a server client system with the server connected to multiple clients and broadcasting the media data from the source client to all the connected peers. But this is a legacy system of RTC and faces a lot of challenges when implemented for a real world application. For example, this method requires intensive server intervention, additional coding decoding algorithms for sending media, lack of end to end encryption etc. WebRTC is a clear alternative for this type of communication. It does not require any additional plug-in or third party application support for running as this will run on the client's browser. WebRTC basically works on peer to peer (P2P) connection for communication, minimizing the server workload. This technology also provides End-to-End Encryption for all the media services. It will be a futile practice if we send the audio and video data as it is through the internet as it may lead to increased latency while streaming. Hence WebRTC divides the audio waves and video frames into small blocks and then sends them over the channel. This process is called Codec and is implemented automatically along with WebRTC. WebRTC will require client's permission to access the microphone as well as the camera to start a final communication but before this there are certain processes that take place that basically allow all the peers to be connected in a channel. Firstly, the clients must know how to locate other client's web browser and create a session with it. This is simply done with the help of the server which will return the desired public IP address of the peer. Once all the peers are connected in a channel and protocol related negotiations have taken place, WebRTC comes into action and the real time communication initiates. Providing media stream here is necessary in order to call or answer the call but if the media stream is not present while answering the call, then the call will be a one-way call.

### III.7 PeerJS

The video call, audio call, peer-to-peer data of WebRTC is simplified by PeerJS. A peer is able to connect with other peer with just an ID. WebRTC implementation of a browser is wrapped so as to access an easy, configurable and complete peer-to-peer connection.

### III.8 Screen Sharing

This project also brings forth the idea of sharing screen. This project is made so that it can be harnessed in organizations for virtual meeting and teaching purposes. It becomes cumbersome to track employee's progress or a student's work verbally when communication is taking place virtually. Therefore, the idea of "Screen Share" works out as the best way to make virtual communication more tangible. The screen share feature will allow the peers to share their chrome window, application window or the entire screen to all the participants present in the meeting. This feature is implemented by keeping the list of every participant ID and rendering it for everyone through the channel. The client who wants his/her screen to be shared needs to capture the screen window first and an object of the captured screen is created so that it can be passed to every participant ID present in the list. To stop sharing, this captured object is removed from the frame of every user.

### III.9 Web Hosting:

When a website (post, images, code) is to be published on the World Wide Web, the hosting provider provides the web server to buy or rent some space. This process is nothing but Web Hosting.

## IV. APPLICATIONS

1. Unlimited number of Participants: This application can have number of participants connected together over a video call and share ideas, thoughts. Participants are allowed to share screen with each other. They also get an option to chat with other participants which makes the meeting more interactive.
2. Share screen feature: The share screen feature of the application helps you to share your entire screen or a particular tab with other participants which will make the meet more interesting.
3. Modern communications for your dispersed team: Your entire work-team, colleagues can collaborate, work easily in agility.
4. Single platform for meetings, phone, webinars & chat: By using this single tool, people/employees/teachers/students can share content, communicate, share their ideas and thoughts and meet easily at anytime of the day.
5. Reimagining Online Video Conferencing: Using video and voice, this proposed system brings innovation into meetings and webinars.



## V. RESULT

A web application is created which will allow user to connect to other users through audio and video along with this the user can share screen and send text messages and images through chat-box to all the users present in the meeting or specifically to the host. This application is created while focusing on the needs of the teacher which leads to the addition of nodding feature which lets students to nod by choosing "yes" or "no" button, generating notification allowing teacher to view it even in full screen mode at the time of screen sharing.

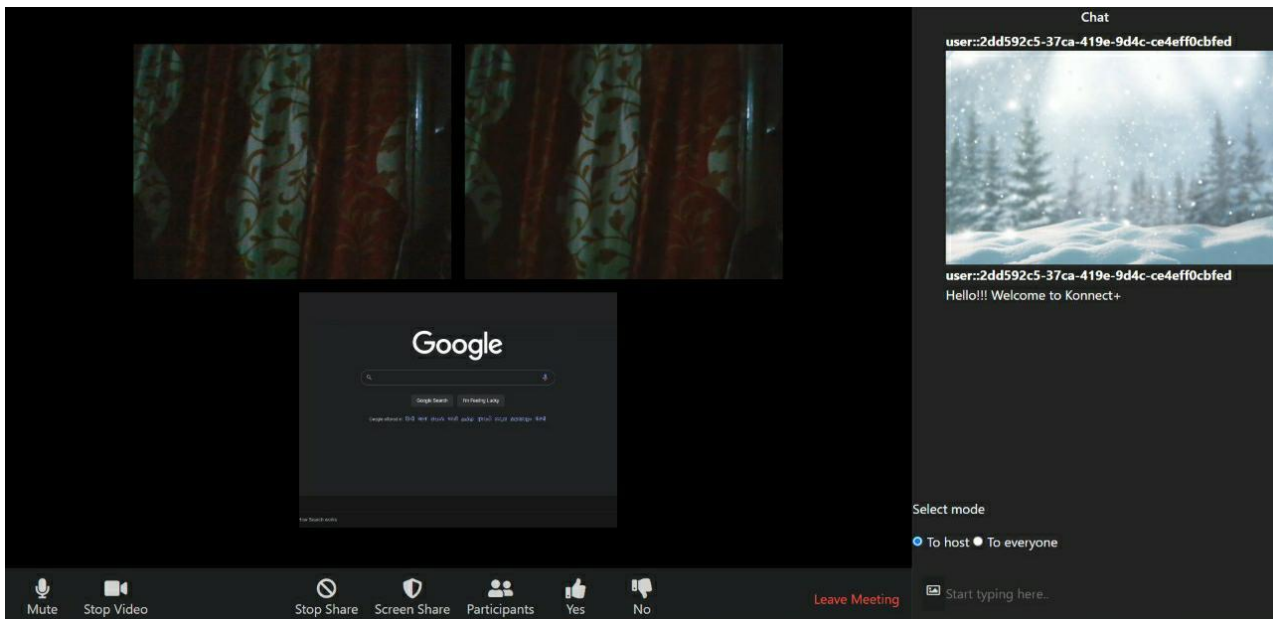


Figure 4: Screenshot

## VI. CONCLUSION

The use of socket.IO and PeerJS initiates peer-to-peer connection bypassing the server which helps in decreasing the load on server as well as resulting in faster transmission of real time data, whereas the chat feature is handled by server for directing the messages to all the users or to the host depending upon the request of user. To send images using same bandwidth as chat functionality images are compressed as base64 encoded format. Instead of using plain HTML/CSS, EJS files are used to render content at real time to allow more dynamic interaction with the server. The use of unique room ID for every room permits users with valid room ID to join the room and each user is uniquely identified with a user ID associated with the user. Also, the user who creates a room is identified as a host. In the future depending upon the demands, this application can be extended to store media files shared during the meet in database and give access to the participant. A facility can also be provided to share screen by multiple users at the same instance of time.

## VII. REFERENCES

- [1] J. M. Denstadli, T. E. Julsurud, "Videoconferencing as a Mode of Communication," Journal of business and technical communication, vol. 26(1), pp. 65–91, January 2012.
- [2] P. Slovak, "Effect of Videoconferencing Environments on Perception of Communication," Journal of Psychosocial Research on Cyberspace, vol. 1, 2007.
- [3] K. V. Rop, N. Bett "Video conferencing and its applications in distance learning," Annual Interdisciplinary Conference, The Catholic University of Eastern Africa, Nairobi Kenya. Volume: 1, Jun 2012.
- [4] C. B. Kumar, A. Potnis and S. Gupta, "Video conferencing system for distance education," 2015 IEEE UP Section Conference on Electrical Computer and Electronics (UPCON), Allahabad, 2015, pp. 1-6, doi: 10.1109/UPCON.2015.7456682.
- [5] C. Chiang, Y. Chen, P. Tsai and S. Yuan, "A Video Conferencing System Based on WebRTC for Seniors," 2014 International Conference on Trustworthy Systems and their Applications, Taichung, 2014, pp. 51-56, doi: 10.1109/TSA.2014.17.
- [6] M. M. Archibald, R. C. Ambagtsheer, "Using Zoom Videoconferencing for Qualitative Data Collection: Perceptions and Experiences of Researchers and Participants", International journal of qualitative methods, Vol. 11, pp 65, September 2019.