International Journal of Scientific Research and Engineering Development-– Volume X Issue X, Year

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Real-time Communication Web Application

Samiksha Kamble*, Mrs. P. S Bedage**

*(B.Tech- Computer Science and Engineering, Sharad Institute of Technology College of Engineering)

** (Asst. Professor- Computer Science and Engineering, Sharad Institute of Technology College of Engineering)

--************************----------------------------------

Abstract:

Real-time communication (RTC) can be a new standard and broad-based initiative that expands the online browsing model, allowing access to information in areas such as communication, chats, video conferencing, and TV on the web, as well as integrated communication. With WebRTC (Web Real-Time Communication), you will add real-time communication power to your app that works beyond the open level. Supports video, voice, and general data to be sent among peers, allowing developers to create powerful voice and video communication solutions. The technology behind WebRTC is used as an open web standard and is available as standard JavaScript APIs entirely for large browsers. WebRTC can be State-of-the-Art technology that creates real- time communication capabilities for audio, video, and data communications in real-time communication using web browsers using JavaScript APIs (Application Programming Interfaces) without plug-ins. This aims to introduce a P2P video conferencing system supported by Web Real-Time Communication. During this project, we developed a real-time peer-to-peer communication system that allows users to talk about instant data transfer to a channel using WebRTC technology, HTML5 and use of a Node.js server address. Our experiments suggest that WebRTC could be a building site with live video conferencing within an Internet browser.

Keywords: **WebRTC, peer-to-peer communication, API, Node.js, HTML5**

--************************----------------------------------

I. INTRODUCTION

This Web Real-Time Communication (WebRTC) is another norm and industry exertion that broadens the web browsing model. Interestingly, browsers can straightforwardly trade constant media with different browsers in a peer-to-peer manner. The World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF) are mutually characterizing the JavaScript APIs (Application Programming Interfaces), the standard HTML5 labels, and the fundamental correspondence conventions for the arrangement and management of a reliable correspondence channel between any pair of cutting edge internet browsers. The normalization objective is to characterize a WebRTC API that empowers a web application running on any gadget, through secure admittance to the input peripherals like webcams and microphones, to trade continuous media and information with a remote party in a peer-to-peer design. It is an open source and free venture that used to give continuous correspondence to mobile applications and internet browsers with the assistance of API's. This task arose as another standard which expands the web-perusing model by empowering the program in a peer-to-peer way. JavaScript API's, HTML5 labels, underlying communication protocols and so forth were characterized by the W3C and IETF mutually to make a reliable correspondence channel between the future internet browsers. Essentially, the fundamental thought was to characterize the WebRTC API which permits the protected admittance to the input peripherals like microphones and webcams on a gadget to share or trade the media- information, real time information with a remote gadget in a peer-to-peer way. In this Task, Video calls, live video conference, sharing your favorite moments with all your loved ones are a few examples where WebRTC exists internally. Every one of the devices that you use on regular routine like cell phones, laptops, smart TVs and AI and so forth all are associated with the Internet. With the assistance of WebRTC, these gadgets can share voice, video, and real-time information easily and safely among each other on a common stage. WebRTC is the future of real-time communication for sure. But why Node.js is so important here? Ryan Dahl made Node.js and the primary arrival of this open-source JavaScript-based runtime environment was in 2009. Dahl fabricated this cross-platform runtime environment on the V8 JavaScript engine of Chrome. Node.js is extremely useful and it is an open-source, cross- platform, backend JavaScript runtime environment that sudden spikes in demand for the V8 motor and executes JavaScript code outside an internet browser. Node.js allows engineers to utilize JavaScript to compose command line tools and for server-side scripting—running scripts server-side to produce dynamic website page content before the page is sent to the client's web browser. Subsequently, Node.js addresses a "JavaScript everywhere" worldview, bringing together web application improvement around a solitary programming language, as opposed to various dialects for server side and client side scripts. However .js is the standard filename expansion for JavaScript code, the name "Node.js" doesn't allude to a specific document in this unique situation and is just the name of the product. Node.js has an event driven design fit for asynchronous I/O. These plan decisions expect to streamline throughput and adaptability in web applications with many input/output tasks, just as for real-time Web applications e.g., real-time communication projects and browsers games. Another important library is nothing but Socket.IO.Socket.IO is a JS library for real-time web applications. It empowers real-time, bidirectional communication between clients and servers. It Parts have an almost indistinguishable API.Like Node.js yes it is event driven. Next is EJS. EJS is a straightforward templating language that allows you to create HTML markup with plain JavaScript. No strictness about how to put together things. Another framework is Express. Express is a versatile Node.js web application framework that gives an amazing game plan features to make web and convenient applications. It supports the quick headway of Node based Web applications. Following are a bit of the middle features of Express framework –

- Permits to line up middlewares for responding to HTTP Requests.
- Characterizes a controlling table which is employed to perform different exercises reliant upon HTTP Method and URL.
- Permits to logically convey HTML Pages reliant upon passing conflicts to designs. Then PeerJS enhances WebRTC data for peers, video, and audio calls.

PeerJS wraps WebRTC browser implementation to provide a complete, customizable, and easy-to-use peer-to-peer API. Outfitted with only an ID, a peer can make a P2P data or media transfer association with a remote peer.

1.1 Background

In May 2010, Google purchased Global IP Solutions or GIPS, a VoIP and video conferencing programming organization that had created a huge number of components needed for RTC, for example, codecs and echo retraction procedures. Google open- sourced the GIPS innovation and drew in with important principles bodies at the IETF and W3C to guarantee industry agreement.

In May 2011, Google delivered an open-source project for browser based real-time communication known as WebRTC. This has been trailed by progressing work to normalize the applicable protocols in the IETF and program APIs in the W3C.

In May 2011, Ericsson Labs constructed the main execution of WebRTC utilizing a changed WebKit library. In October 2011, the W3C distributed its first draft for the spec. WebRTC milestones incorporate the principal cross-browser video call-February 2013, first cross-browser data transfer -

February 2014, and as of July 2014 Google Hangouts was "somewhat" utilizing WebRTC.

The W3C draft API depended on fundamental work done in the WHATWG. It was alluded to as the Connection Peer API, and a pre-principles idea execution was made at Ericsson Labs. The WebRTC Working Group anticipates that this specification should develop altogether dependent on:

- Outcomes of continuous trades in the companion RTCWEB bunch at IETF to characterize the arrangement of real-time communications in web browsers. While nobody flagging protocol is commanded, SIP over Web Sockets (RFC 7118) is frequently utilized part of the way because of the materialness of SIP to the vast majority of the imagined communication situations just as the accessibility of open- source software like JsSIP.
- Privacy gives that emerge while uncovering local abilities and local streams
- Technical conversations inside the group, on carrying out data channels specifically
- Experience acquired through early experimentation
- Feedback from different communities and people

In November 2017, the WebRTC 1.0 particular changed from Working Draft to Candidate Recommendation.

In January 2021, the WebRTC 1.0 particular changed from Candidate Recommendation to Recommendation.

1.2 WebRTC

WebRTC is a plugin-free present day real-time communication innovation. It doesn't need any extra modules or applications for sound, real time video conferencing and data sharing. It utilizes JavaScript, application programming interfaces (APIs), and HTML5 to install the communication advances inside the browser. Products like Google Hangouts, Whatsapp, Facebook Messenger, ZOOM Team Communication, Zendesk Customer Support, and Skype for Web and so on all are coordinated with WebRTC. Browsers can straightforwardly trade real-time media with different browsers in a peer-to-peer manner. Offers a high level safety than different other streaming frameworks, without the requirement for outsider software. It is accessible for nothing and is worked overall which is the fundamental supporter for this innovation. So there are many advantages of using WebRTC technology. All video calling platforms like Skype rely on programming programs, modules or applications to run. However, WebRTC needn't bother with the whole of that you can essentially use your program and use WebRTC to talk or video call. Chrome was the chief program that maintained WebRTC when it was dispatched in 2012. In a little while Firefox trailed in like manner and subsequently was joined by the Norwegian program Opera. Microsoft in 2015 offered assistance for WebRTC with the introduction of the Edge program. Apple was the last to make their program practical with Edge program with the appearance of Safari 11. This time there was an opening in choice of WebRTC as Safari was the most used program after Chrome. As of now WebRTC can expect to get a wide reception as it offers a steady response for online video visiting from different contraptions which maintain web programs.

II. RELATED WORK

In [1] this research carries out a basic brought together communication-based WebRTC utilizing node.js signaling server. This framework supports p2p associations after signaling negotiation, accordingly clients can perform real-time communication, and for example, chatting, video conferencing, audio conferencing (click-to-call), and file transfer. So, this system has many beneficial features. Also, if this framework facilitated on the web, this framework support browser-mobile devices and supports multimedia real-time connection between the mobile phones. This framework execute HTML5, EasyRTC Client API, JavaScript and CSS on the client side, in any case the HTTP worker foster utilizing node.js Express module and arrange Google's public test server as STUN server. The examination result is the framework running on the HTTP server have minimal expense equipment, real-time communication work running great in the entire communication flow. This framework runs great utilizing Google Chrome version 51.0.2704.103 m (64-bit) and Firefox version 47.0.

In [2] from this proposed system one can really understand how WebRTC is beneficial! The utilization of WebRTC innovation empowered the execution of secure and high data transmission between clients as peer-to-peer or peer-to-peer connection in real-time communication, consequently, anybody can make their own Webpage or application like real- time sharing documents, real-time communication climate as informing talk or video/sound conferencing this opened the route for software engineers and designers to enter the genuine occupation advertise and contend with web-based media proprietors. Utilizing WebRTC innovation permits us to make a webpage with the most remarkable highlights, permitting every client to associate with another by means of Text messages, video/sound call by utilizing basic JavaScript APIs and Node JS. The server related with Google STUN and TURN server. This model really has great features like good user interface, Friend request and chat, calls, etc.

In [3] this paper, you'll discover how to understand the essentials of WebRTC, how to set up and utilize both data and media associations, and the sky is the limit from there. This paper clarifies about empowering the online students to have live audio/video talks with the other companion and instructor so they convey well while disclosing the issue to the next companion. The paper likewise demonstrates such ability by utilizing developing expertise like WebRTC and WSC.

In [4] there is such system in which, when the network is set up, the browsers can share media transfers like sound and video from the audio and camera, share documents or send and get messages through the quickest way: peer-to-peer. With this peer to peer innovation as the establishment for the web application, the framework can give different learning abilities. The student finds the master and can set up an immediate association. WebRTC handles all the significant traffic and transfers the media streams among the peers. Extra functionalities are likewise upheld that advantage the student like texting, document sharing and screen sharing which assists the student with effectively interfacing with the master for the reasonable comprehension of a specific concept.

In [5] this paper, there is a web-based peer-to-peer real-time communication framework utilizing the Mozilla Firefox together with the ScaleDrone administration that empowers clients to communicate with rapid data transmission over the communication channel utilizing WebRTC innovation, HTML5 and use Node.js server address. This analysis also shows that WebRTC is an able structure block for adaptable live video conferencing inside a web browser. This system has beneficial features. In this paper there is a study related to thermal camera for the videoconference framework to identify the temperature body results with respect to the COVID 19 crisis.

III. METHODOLOGY

3.1 Real-time communication in the browser

A WebRTC web application ordinarily composed as a blend of HTML and JavaScript collaborates with web browsers through the normalized WebRTC API, permitting it to appropriately endeavor and control the constant browser work. The WebRTC web application additionally interfaces with the browser, utilizing both WebRTC and other normalized APIs, both proactively e.g., to query browser abilities and responsively e.g., to get browser-created notifications. The WebRTC API should thusly give a wide range of functions, similar to connection management (in a peer-to-peer style), encoding/decoding abilities exchange, determination and control, media control, firewall and NAT element traversal, and so on. The plan of the WebRTC API addresses an issue. It conceives that a constant, real-time flow of information is gushed across the organization to permit direct correspondence between two browsers, with no further mediators along the way. This obviously addresses a progressive way to deal with web-based communication.

Allow us to envision an ongoing audio and video call between two browsers. Correspondence, in such a situation, may include direct media streams between the two browsers, with the media way arranged and started up through a mind boggling succession of connections including the following entities:

- The caller browser and the caller JavaScript application (e.g., through the referenced JavaScript API)
- The caller JavaScript application and the application supplier (commonly, a web server)
- The application supplier and the callee JavaScript application
- The callee JavaScript application and the callee browser (through the application-browser JavaScript API)

3.2 System Design

For the most part, a WebRTC application needs following things:

- Audio or Video transfers
- Network data
- A correspondence medium to report mistakes (assuming any), start, halt or end meetings
- Player for the Audio/Video transfers

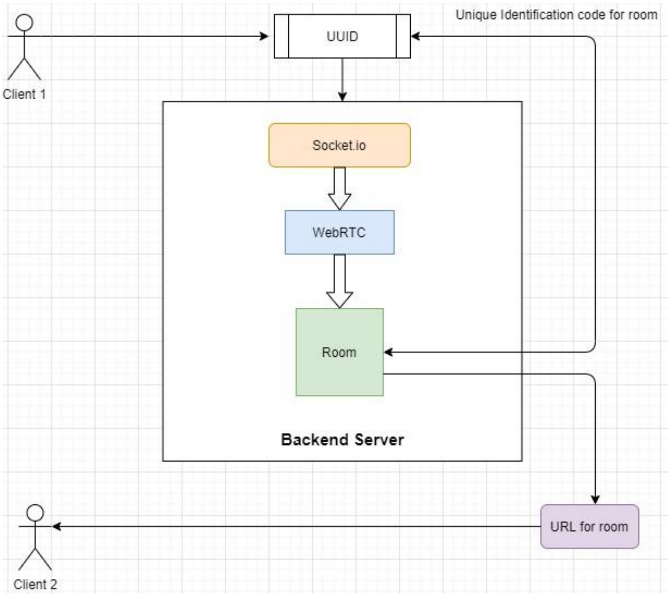

Fig.1 Backend design of proposed system

A Peer Connection permits two clients to impart straightforwardly, browser to browser. It at that point addresses a relationship with a remote peer, which is typically another case of a similar JavaScript application running at the remote end. Correspondences are facilitated through a signaling channel gave by prearranging code in the page by means of the web server, e.g., utilizing XMLHttpRequest or WebSocket. When a peer connection is set up, media transfers (privately connected with impromptu characterized MediaStream objects) can be sent straightforwardly to the remote browser.

The content is distributed using servers that use high-security protocols to ensure that all transmitted information is safe and secure. For the creation of a video conference room, the server application generates a UUID for the data flow. An URL with UUID as a parameter will be created for identifying users and connecting them to the room.

The server records all requests and informs the user of incoming messages and calls based on that information. While the server is listening for calls or requests, information about the paths between

the participants and the parameters required for each call will be transferred.
Two servers connected together; Node.JS server for implementing WebRTC technology. Videos and image files uploaded by a user to the server will be compressed while maintaining the high resolution of uploaded files.
Using EJS, CSS, JavaScript, and JSON, AJAX Techniques on the client side, as well as Express Socket.IO, Peer.JS, and Node.JS on the server side.
Peer connections are used to communicate directly between two browsers, representing a connection with a remote peer, usually another JavaScript instance running at the remote end. A communication channel is established with a web server by means of scripting code in the page, using a protocol like XMLHttpRequest or WebSocket. Media streams can be sent directly to the remote browser once the peer connection has been established.

3.3 Requirements

Hardware Requirements:
- Processor Name: Intel/AMD
- Processor Speed: 3.2 GHz
- Storage: SSD cloud
- Memory: 512 RAM (Minimum)

Software Requirements:
Technology stack
- Backend - Node.js, socket.io, expressjs, webRTC
- Frontend - EJS, CSS

For Website:
- Technology Implemented: Node JS express server
- Language Used: Node JS
- User Interface Design: JSON API
- Software to fetch API request: Postman

For Software Product:
- Operating System: Ubuntu Linux
- Programming Language: Node JS/ JavaScript
- Software: NPM

3.4 Experiment Result and Analysis

Let's have a look at some apps Google Meet, Zoom, and Microsoft Teams. Zoom has a cut-off for the individuals who are utilizing its free plan. In spite of the fact that you can make an unlimited amount of calls, each call can simply last as long as 40 minutes. But in this project, there is no time limit and also the service is also totally free. Google Meet and Microsoft

Teams also don't have any time limit. These apps are brand apps. But in this project, we have used minimum server hardware resources i.e. 512 MB of RAM Minimum storage as per the size of the project, and run the Node.js server on top of the free Heroku hosting service. There are some bandwidth requirements for video quality and this bandwidth may vary from device to device. So these brand apps like Google Meet their minimum hardware requirement would be starting from 1 Mbps bandwidth client side. In this project, one can have a conversation with people with a minimum of 800kbps (client- side bandwidth). All this is possible because of Nodejs, express server, and its asynchronous request handling from server side. In this project, No download, plug-in, or login is required, this is an entirely browser-based system, also unlimited users by accordingly increasing resources. For now one can use this model for having a conversation with friends, family and also for arranging small-scale meetings, conferences, etc. You can access this service from all devices. In the future, there will be many more features.
This is how the model looks.

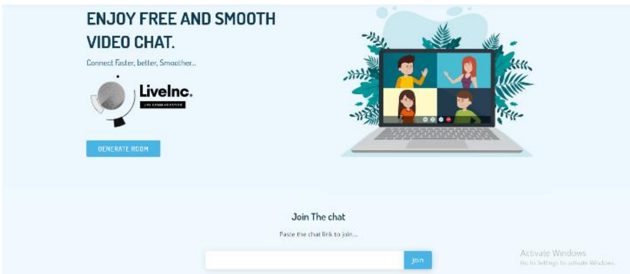

Fig.2 User interface of web page

One can paste the link in that box and join the conversation or also one can create their own room.

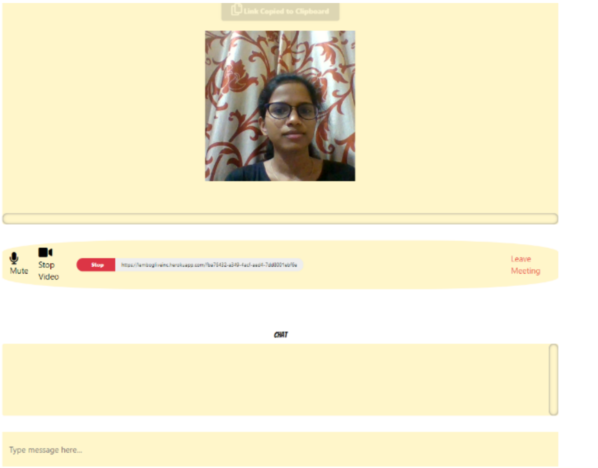

Fig.3 Room creation

After creation of room one can create link and invite others

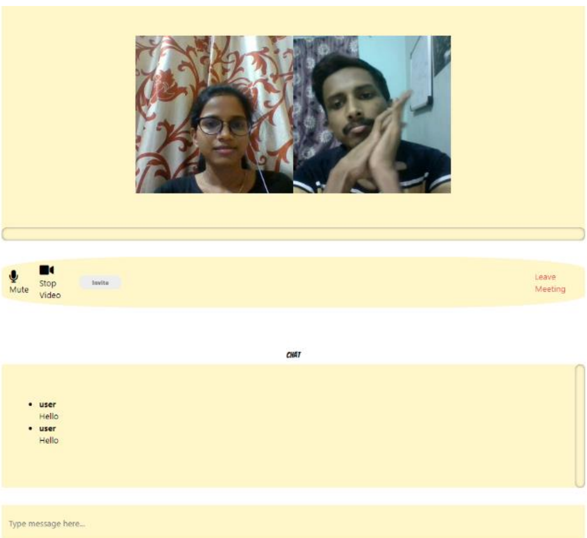

Fig.4 Chat box

So now one can have a conversation. Chat box feature is also there.

IV. FUTURE SCOPE

The future scope of this work can be to do further work on it and to make it better and efficient. Following are some features:

- Webcam streaming
- Audio streaming
- Screen sharing to present documents, slides, and images.
- GIF, stickers sharing
- Full Screen Mode on click
- Possibility to Change UI Themes
- An annotation tool
- Background Feature
- To provide more efficiency
- Meetings, Conferences without much lagging

V. CONCLUSION

In the present huge scope market of WebRTC, Node.js is the ideal decision to construct an ideal WebRTC empowered video chat application. This application can give clients huge performance, makes API's, and handles parallel requests, and demands adaptability to develop effective video/voice chat applications in Android, iOS, and Web. With WebRTC, you can add real-time communication to your application that chips away at top of an open norm. It upholds video, voice, and conventional information to be sent between peers, permitting engineers to construct amazing voice and video communication solutions. The innovation is accessible on all cutting-edge browsers just as on native clients for every single significant platform. The advances behind WebRTC are executed as an open web standard and accessible as regular JavaScript APIs in every significant browser. For native clients, like Android and iOS applications, a library is accessible that gives a similar usefulness. The WebRTC project is open source and upheld by Apple, Google, Microsoft and Mozilla, among others.

REFERENCES:

[1] Research of Web Real-Time Communication - the Unified Communication Platform using Node.js Signaling Server Florensia Unggul Damayanti Department of Information Technology, Swiss German University, Tangerang 15143, Indonesia (2018)

[2] Peer to Peer Multimedia Real-Time Communication System based on WebRTC Technology Article in International Journal for the History of Engineering & Technology · February 2019

[3] Real Time Information and Communication Center based on WebRTC Himanshu Bhardwaj1, Adarsh Lunthi2, Hitesh Bhat3, Karan Singh Rawat4, Aashita Chabbra5 1-4Student, Department of Information Technology ADGITM, Delhi, India 5Assistant Professor, Department of Information Technology ADGITM, Delhi, India International Research Journal of Engineering and Technology (IRJET) – (2020) www.irjet.net

[4] WebRTC Peer to Peer Learning H. Fateh Ali Khan 1, A. Akash 2, R. Avinash 3, C. Lokesh 4 1,2,3,4 Department of Information Technology, Valliammai Engineering College, Chennai, India (2018) www.ijert.org

[5] G. Suciu, S. Stefanescu, C. Beceanu and M. Ceaparu, "WebRTC role in real-time communication and video conferencing," 2020 Global Internet of Things Summit (GIoTS), 2020, pp. 1-6, doi: 10.1109/GIOTS49054.2020.9119656.