

Name: Samarth Laxmikant Kulkarni.



**Project Name:**

**Video Chatting website.**

**College Name:**

**MITCORER, Barshi.**

## ABSTRACT:

Web-based video chat application that allows users anywhere in the world to join real-time streaming video chat rooms. This product is similar to social networking sites that allow web-based video conferencing.

The demand for social networking sites is increasing day by day. A social networking site that allows you to video chat online is the primary inspiration for my project. The goal of my project is to build an online video chatting tool that enables users to join real-time streaming video chat rooms where users can share their video with multiple users. Users can send instant messages and share their live web cam data to other users in the chat room.

Users can directly enter webcam chat rooms. Also this video chat application doesn't require any additional software installation on the client side.

The main technology used in this project is WEBRTC and Firebase. Real-time communication for the web With WebRTC, you can add real-time communication capabilities to your application that works on top of an open standard. It supports video, voice, and generic data to be sent between peers, allowing developers to build powerful voice- and video-communication solutions.

The technology is available on all modern browsers as well as on native clients for all major platforms. The technologies behind WebRTC are implemented as an open web standard and available as regular JavaScript APIs in all major browsers. For native clients, like Android and iOS applications, a library is available that provides the same functionality. The WebRTC project is open-source and supported by

Apple, Google, Microsoft and Mozilla, amongst others. This page maintained by the Google WebRTC team.

Firebase is a real-time database that allows storing a list of objects in the form of a tree. We can synchronize data between different devices. Google Firebase is Google-backed application development software which allows developers to develop Android, IOS, and Web apps. Firebase manages all data real-time in the database. Hence, if you are looking to develop mobile apps such as live streaming, chat messaging, etc., you can use Firebase. Firebase allow syncing the real-time data across all the devices- Android, iOS, and the web without refreshing the screen

## Table of Contents:

| Sr.no | Title                                    | Page |
|-------|--|------|
| 1     | <b>Introduction</b>                      |      |
| 2     | <b>Existing Method</b>                   |      |
| 3     | <b>Proposed method with Architecture</b> |      |
| 4     | <b>Methodology</b>                       |      |
| 5     | <b>Implementation</b>                    |      |
| 6     | <b>Conclusion</b>                        |      |

# **1.Introduction:**

Web-based video chat application that allows users anywhere in the world to join real-time streaming video chat rooms. This product is similar to social networking sites that allow web-based video conferencing.

The demand for social networking sites is increasing day by day. A social networking site that allows you to video chat online is the primary inspiration for my project. The goal of my project is to build an online video chatting tool that enables users to join real-time streaming video chat rooms where users can share their video with multiple users. Users can send instant messages and share their live web cam data to other users in the chat room

Users can directly enter webcam chat rooms after signing up. Also this video chat application doesn't require any additional software installation on the client side.

Over the last decade, the demand for video conferencing has increased enormously. Video conferencing is a software technology that allows group of people anywhere in the world to meet in a virtual room by simply connecting to internet using their personal computer and sharing live webcam. Video conferencing has its application in online distance learning, virtual meeting rooms in business, and social networking sites. Online distance learning allows students and teachers to meet in a virtual class room without the need to waste time and money on commuting.

A virtual meeting room has significant benefits in the global market where the employees anywhere in the globe can share their ideas

without needing to physically present in the company. A social networking site allows friends and families to connect and get in touch with their loved ones. Video conferencing in social networking sites allows friends and families to come together in a chat room and share their voice and video.

The problem with most of the commercial video conferencing systems is that, they have complex user interface (UI), platform dependent, high cost and require additional software installation. My goal of this project is to build a video conferencing social networking site that allows users anywhere in the world to enter virtual chat rooms.

The main technology used in this project is WEBRTC and Firebase. Real-time communication for the web With WebRTC, you can add real-time communication capabilities to your application that works on top of an open standard. It supports video, voice, and generic data to be sent between peers, allowing developers to build powerful voice- and video-communication solutions.

The technology is available on all modern browsers as well as on native clients for all major platforms. The technologies behind WebRTC are implemented as an open web standard and available as regular JavaScript APIs in all major browsers. For native clients, like Android and iOS applications, a library is available that provides the same functionality. The WebRTC project is open-source and supported by Apple, Google, Microsoft and Mozilla, amongst others. This page maintained by the Google WebRTC team.

Firebase is a real-time database that allows storing a list of objects in the form of a tree. We can synchronize data between different devices. Google Firebase is Google-backed application development software which allows developers to develop Android, IOS, and Web apps.

Firebase manages all data real-time in the database. Hence, if you are looking to develop mobile apps such as live streaming, chat messaging, etc., you can use Firebase. Firebase allow syncing the real-time data across all the devices- Android, iOS, and the web without refreshing the screen

## 2.Existing Method:

### WebRTC:

The very definition of WebRTC is peer-to-peer communication between web browsers. The reality is, most browsers run in computers that operate behind a NAT (network address translation) device, and optionally a firewall. The NAT device, usually a router or modem, allows computers with private IP addresses to connect to the Internet via a single public IP address.

NAT devices protect personal computers from being directly exploited by malicious users over the Internet via the IP address. Unfortunately, it also blocks devices with private IP addresses from connecting to another private IP device over the Internet.

Simple WebRTC is a platform that provides an easy and cost-effective service for developers to build and deploy custom real-time applications using React. Specifically, they provide the following:

- Simple WebRTC SDK: a front-end library
- Hosting: STUN/TURN and SFU (Selective Forward Unit) servers
- Technical support
- Custom app development and WebRTC consulting services
- Single-tenant and on-premise infrastructure

### Firebase:

Firebase is a real-time database that allows storing a list of objects in the form of a tree. We can synchronize data between different devices. Google Firebase is Google-backed application development software which allows developers to develop Android, IOS, and Web apps.



Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.

Firebase is Google's mobile application development platform that helps you build, improve, and grow your app. Firebase is Google's mobile application development platform that helps you build, improve, and grow your app.

Node js:

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project! Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.

We will include the video chat portion into our app. We will walk through building events that allow two peers to communicate. We will also walk through building the signaling server that is used for peer to peer connections.

### **What are the requirements?**

- Basic Knowledge of Node.JS
- Basic Knowledge of NPM packages: PeerJS peer
- Basic Knowledge of CSS
- Basic Knowledge of HTML

Web RTC is an open source project to enable real time communication of audio, video and data in Web and native apps. Web RTC has several JavaScript APIs click the links to see demos. Get User Media() : capture audio and video.

### **3.Proposed method with Architecture :**

Firstly create html page using attractive website and they provide links video calling and chatting module.They are aslo 3 module can we used

- 1.Html and css
- 2.Node js
- 3.WebRTC
- 4.Firebase

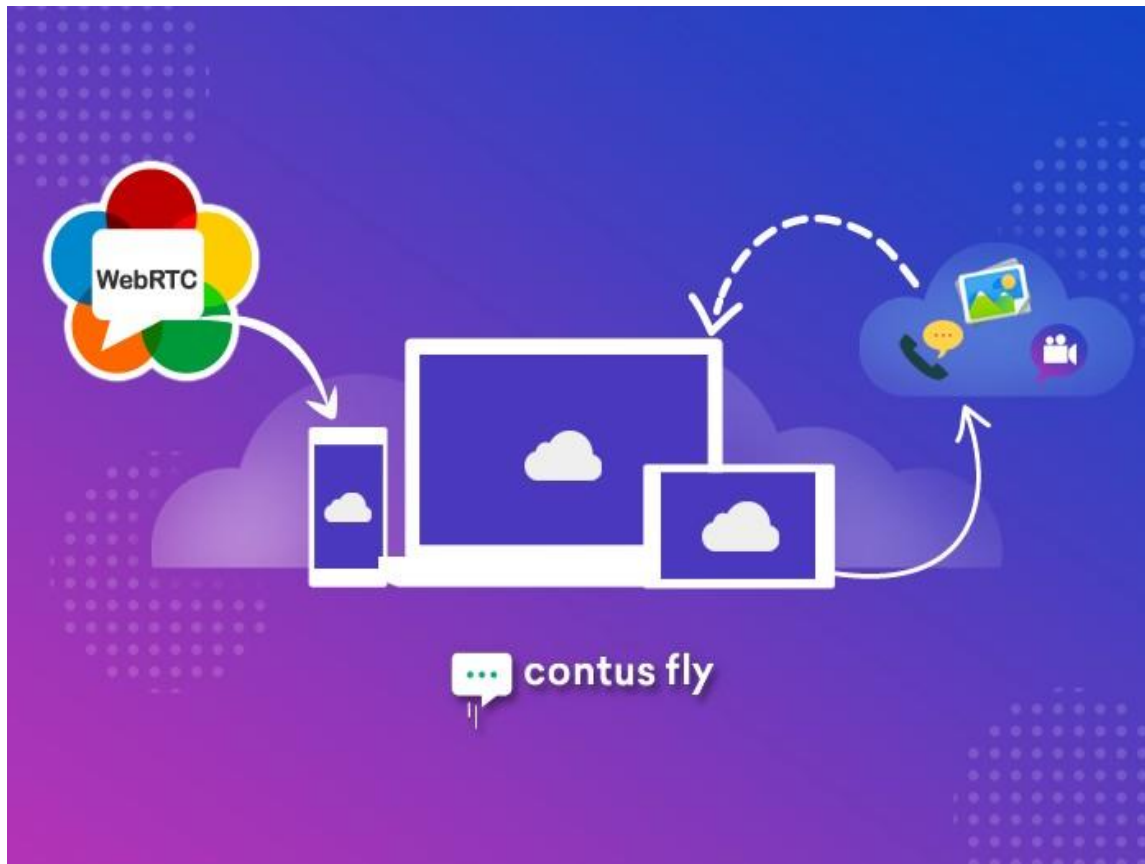
1.Html and css:

They aslo using responsive website and easy to handle of users.

2.Node js:

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project! Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.

We will include the video chat portion into our app. We will walk through building events that allow two peers to communicate. We will also walk through building the signaling server that is used for peer to peer connections.



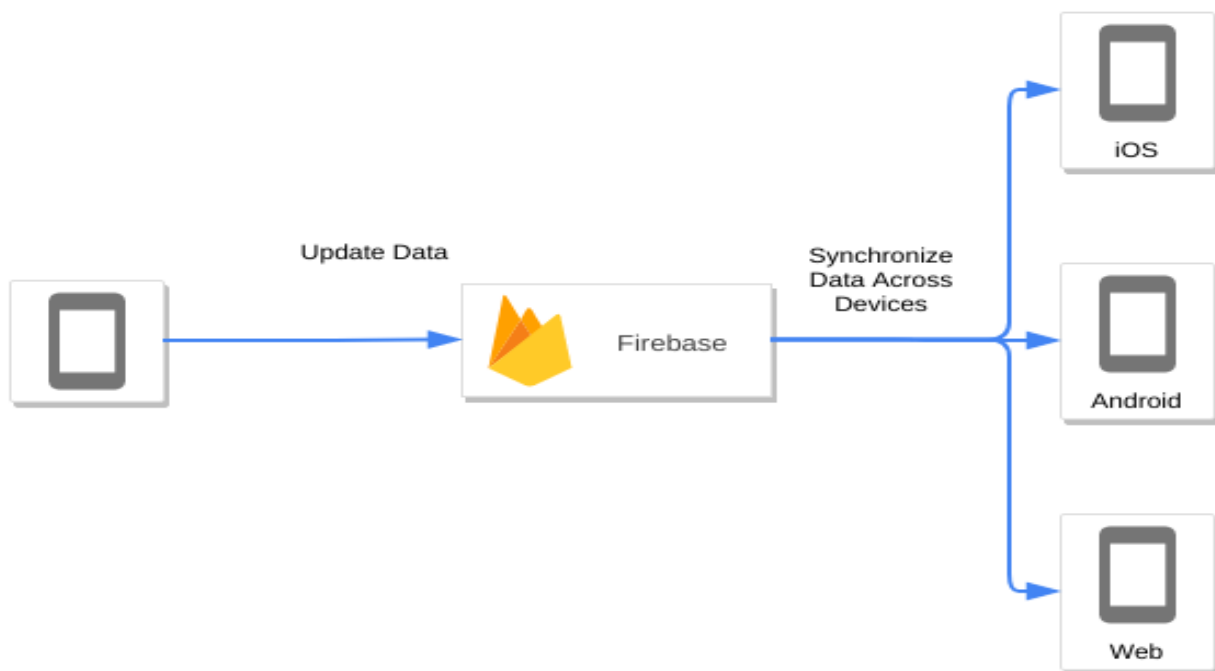
### 3.WebRTC:

WebRTC (Web Real Time Communication) is an open-source project that allows peer-to-peer, real-time communication between web browsers to stream live video, audio and data streams over a network. Modern desktop and mobile browsers such as Google Chrome, Mozilla Firefox, Safari, Opera and other Chromium-based browsers have already implemented this technology natively. This is good news, as users don't have to install a third-party plugin or an app to access the technology.



#### 4. Firebase:

Firebase is a live database (it has other very useful features aside from this). If you add data to your Firebase database, someone on your website doesn't need to refresh the page to see the new data. ... Firebase will allow us to send and receive messages live, which you need to get the video chat to work.



## 4. Methodology :

Daily is built on WebRTC, which is an open, secure standard for streaming media in browsers. There is no single webRTC technology — webRTC is a collection of standards, protocols, and APIs that make up an open-source framework which allows for real-time communications and data transfers.

Simply put, webRTC means that developers can integrate real-time communication into any website or app. From a product perspective, this means that webRTC can turn video calling into a built-in feature of any product.

Daily's API makes this as fast and flexible as possible.

- **Chat** – Participants of a meeting can effectively communicate with each other using Zoom's integrated chat feature. They can choose to message all members of the meeting at the same time or individually message participants.
- **Encryption** – All Zoom meetings are embedded with end-to-end encryption software that prevents any unauthorized person from joining your meeting. Even the company itself cannot join a meeting without the host's permission.
- **Backend Development** – During the initial stages of building a video conferencing app it is essential to build a server to run your app on. It is always better to build your server from scratch instead of using ready-made technologies.
- **UI Design** – The user interface of your app creates the first impression of your service. Therefore, it is vital to have an appealing and intuitive UI.

Checking browser support & Making a request to the signaling server  
The first thing you have to do when trying to create the connection with the other user is to make sure that their browser or OS provides access to media devices.

**SetupRTC** function tests if the browser supports webRTC and has an access to needed media devices. If these conditions are satisfied, SetupRTC initializes PeerConnection and declares handlers.

**StartCall** verifies if WebRTC is supported and initialized on the client side and initiates the 'callRequest' using Socket.io library.

**TargetId** — user's identifier in the database. The request for connection was sent to it.

**EnableSelfVideo** enables user's camera, then takes and shows the video stream to the user through HTML5, for example. It also passes the stream to the object PeerConnection –  
\$rootScope.pc.addStream(stream).

## 5. Implementation:

Step1:

You'll learn how to build a simple video chat application using the WebRTC API in your browser and Cloud Firestore for signaling. The application is called FirebaseRTC and works as a simple example that will teach you the basics of building WebRTC enabled applications.

Before starting this codelab, make sure that you've installed:

- npm which typically comes with Node.js - Node LTS is recommended

### 2. Create and set up a Firebase project

Create a Firebase project

1. In the Firebase console, click Add project, then name the Firebase project FirebaseRTC.
1. Click Create project.

The application that you're going to build uses two Firebase services available on the web:

- Cloud Firestore to save structured data on the Cloud and get instant notification when the data is updated
- Firebase Hosting to host and serve your static assets

For this specific codelab, you've already configured Firebase Hosting in the project you'll be cloning. However, for Cloud Firestore, we'll walk you through the configuration and enabling of the services using the Firebase console.

Enable Cloud Firestore

The app uses Cloud Firestore to save the chat messages and receive new chat messages.

You'll need to enable Cloud Firestore:

1. In the Firebase console menu's Develop section, click Database.
2. Click Create database in the Cloud Firestore pane.
3. Select the Start in test mode option, then click Enable after reading the disclaimer about the security rules.

### **3. Get the sample code**

Clone the GitHub repository from the command line:

```
git clone https://github.com/webrtc/FirebaseRTC
```

The sample code should have been cloned into the FirebaseRTC directory. Make sure your command line is run from this directory from now on:

```
cd FirebaseRTC
```

### **4. Install the Firebase Command Line Interface**

The Firebase Command Line Interface (CLI) allows you to serve your web app locally and deploy your web app to Firebase Hosting.

1. Verify that the CLI has been installed correctly by running the following command: `sh firebase --version`

Make sure the version of the Firebase CLI is v6.7.1 or later.

1. Authorize the Firebase CLI by running the following command: `sh firebase login`

You've set up the web app template to pull your app's configuration for Firebase Hosting from your app's local directory and files. But to do this, you need to associate your app with your Firebase project.



1. Associate your app with your Firebase project by running the following command: `sh firebase use --add`
2. When prompted, select your Project ID, then give your Firebase project an alias.

An alias is useful if you have multiple environments (production, staging, etc). However, for this codelab, let's just use the alias of default.

1. Follow the remaining instructions in your command line.

## 6. Creating a new room

In this application, each video chat session is called a room. A user can create a new room by clicking a button in their application. This will generate an ID that the remote party can use to join the same room. The ID is used as the key in Cloud Firestore for each room.

Each room will contain the `RTCSessionDescriptions` for both the offer and the answer, as well as two separate collections with ICE candidates from each party.

Your first task is to implement the missing code for creating a new room with the initial offer from the caller. Open `public/app.js` and find the comment `// Add code for creating a room here` and add the following code:

The first line creates an `RTCSessionDescription` that will represent the offer from the caller. This is then set as the local description, and finally written to the new room object in Cloud Firestore.

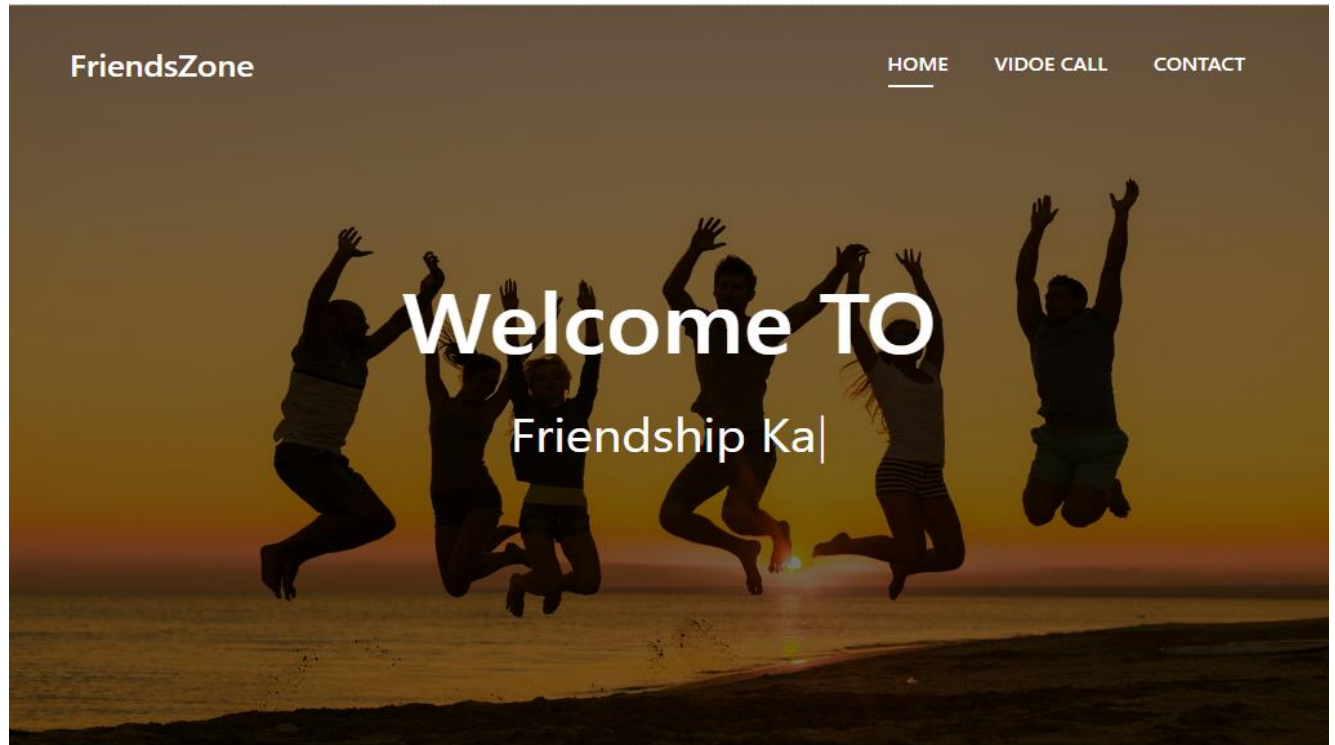
Next, we will listen for changes to the database and detect when an answer from the callee has been added.

## 7. Joining a room

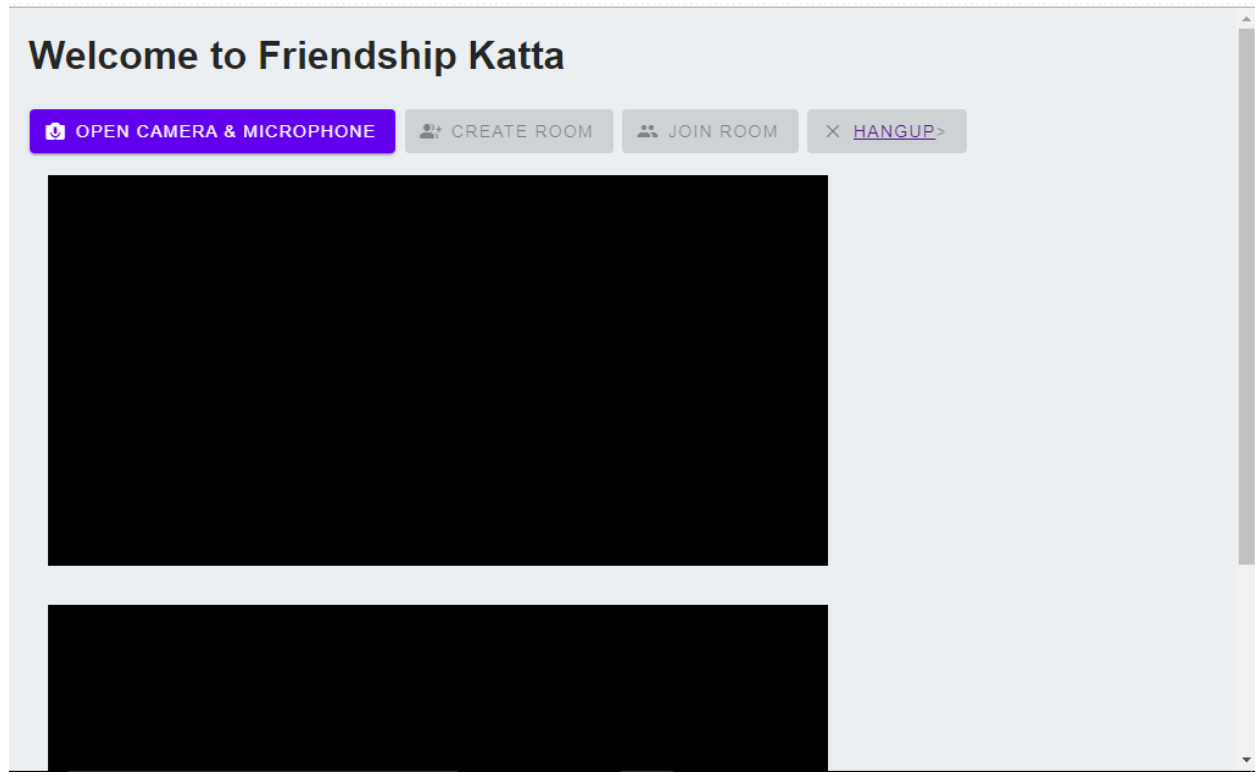
The next step is to implement the logic for joining an existing room. The user does this by clicking the Join room button and entering the ID for the room to join. Your task here is to implement the creation of the RTC Session Description for the answer and update the room in the database accordingly.

## 6.Screenshots:

### 1.Home page:



## 2.Video Call page:



### 3.Chat Page:

## Friendship Katta

Enter your Name

Join ➔

---

# Friendship Katta

sam  
hey good morning

|sam. Say somethings. 

---

## **7.Conclusion:**

To conclude, building a real-time video chat solution with WebRTC is easier when using a commercial platform. Going the open-source route is doable, but you'll need to host your own TURN servers to ensure a reliable connection for all your users. In any case, the building process can be time consuming, given the relative complexity of the sample apps we've looked at.

The key question to ask yourself is whether or not it's worth your time building your own custom real-time solution. Unless you plan on making WebRTC your core business, you'll probably need to first consult with a company that has experience dealing with the technology.