DETAILED REPORT

PROJECT TITLE

AUDIO AND VIDEO CHAT WEB APPLICATION USING WEBRTC

2019202031
MOHAMED AZARUDEEN S
MCA REGULAR (6th Semester)

Abstract

Audio Video Chat Web Application using webRTC(Web Real-Time Communication) enables audio video communication between two or more users. The users connect to our system through the front end. They make use of this web application via web browsers. These browsers must support WebRTC (Google Chrome, Opera and Firefox). WebRTC comprises getUserMedia, RTCPeerConnection and RTCDataChannel APIs. Through these APIs the caller can place a call to the callee. Additionally, there is a back-end which contains data about all the users who have registered with our system. Once users login into the system, users will have two choices, it is either create an audio call or create a video call. In order to create an audio call, the browser will request access to the user's microphone; while if it is a video call, the browser will request access to both the user's microphone and camera. If the request to access is block, the communication will not be able to setup. After the two ways communication was established, users can perform audio chat only if audio call is chosen while video chat can see the people through camera and chat through microphone. Screen sharing and call recording options are included in this application.

Introduction(Domain)

This web application establishes a live connection between two or more than two users at different geographic locations, using computer networks for the transmission of audio and video data.

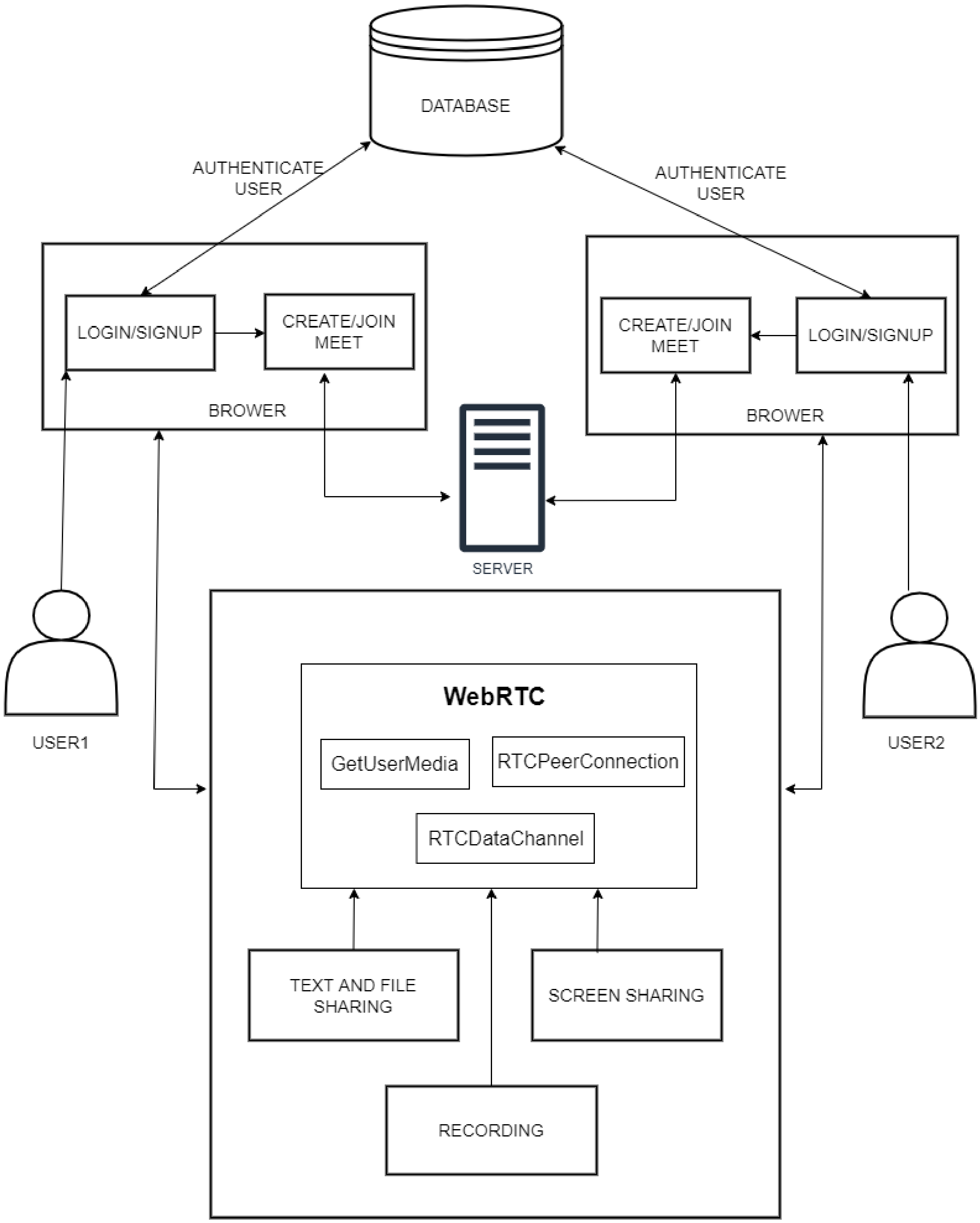
Problem statement

In existing applications, If we want to make a video call, we have to install some video call applications like Zoom, Webex. Need high internet bandwidth connection for an online meet. Large size of recorded video file.

Objective

This web application enables voice and video communication between two and more than two users. This is a secure and reliable application. The use of WebRTC ensures that we send the data over secure data channels thus preventing anyone from gaining access to the data being transmitted. This is a browser based application. Need not install any plug-in or software.

Architecture Diagram



Architecture Explanation

This web application is run on a browser. Once the user's authentication is successfully completed, then the user can create or join a meeting through the browser. The authentication will be done by the Database. If the user creates or joins a meeting, then the server establishes the connection to other users. Once the server establishes the connection all users can communicate with each other using WebRTC. The three APIs that comprise WebRTC - GetUserMedia, RTCDataChannel, and RTCPeerConnection. GetUserMedia manages the selection of input user devices in case a user has multiple cameras or microphones on his device. RTCPeerConnection is an API for making WebRTC calls to stream video and audio, and exchange data. RTCPeerConnection provides methods to connect to a remote peer, maintain and monitor the connection. RTCDataChannel is responsible for the exchange of all real-time data that is not audiovisual. That means text-based chats, peer-to-peer file sharing.

List of Modules

1. User Login/Signup
2. Create a Meeting
3. Join a Meeting
4. Audio, Video Chat
5. Text Chat
6. File Sharing
7. Screen Sharing
8. Meet Recording

Brief Description of Modules

1. **User Login/Signup**
   Users can register his details and login with his credential(Username and Password). Using Firebase database to store user information and authenticate users.
2. **Create a Meeting**
   Users can create a new meeting with a custom meeting code and then the meeting link/code will be generated.

3. **Join a Meeting**
   Users can join the meeting using the meeting link/code.
4. **Audio, Video Chat**
   Users can create either an audio or a video call. In order to create an audio call, the browser will request access to the user's microphone while if it is a video call, the browser will request access to both the user's microphone and camera. If the request to access is block, the communication will not be able to setup. After the two ways communication was established, users can perform audio chat only if audio call is chosen while video chat can see the people through camera and chat through microphone.
5. **Text Chat**
   After connecting a call users can also chat with text messages to each other.
6. **File Sharing**
   Users can share files while they are in a call. Files such as pdf, docs, images. Other Participants can download those files.

7. **Screen Sharing**
   Screen sharing allows users to display their screen with other meeting participants. Users can share either the entire window screen or a particular screen.
8. **Meet Recording**
   Meeting organizer/host can record the meeting for the persons who missed the online meeting and future decisions.

References

- https://webrtc.org/getting-started/overview
- https://nodejs.org/en/docs/guides/
- https://firebase.google.com/docs/web/setup
- https://www.researchgate.net/publication/298711298_P2P_audio_and_video_calling_application_using_WebRTC
- https://ieeexplore.ieee.org/document/9119656