Howdy – Video Conference Web Application

Abstract

With the need to shift to digital education and working from home, there's a huge demand on conference and collaboration technologies. Covid has highly affected our daily lives and education, work and socialization are one of the many things that were the most affected. There are many existing apps like Microsoft Teams, Zoom and Google Meet that suffice the need of collaboration. Howdy is one such collaboration application. It is not as sophisticated as any of the existing apps but just a simple implementation of video conference using WebRTC (P2P Topology) and Socket.io .

Introduction

This whole project has been done using Agile methodology – Scrum methodology, to be more precise. The main focus was put on 1-1 video call as that was the mandatory feature and the rest were added in the subsequent weeks.

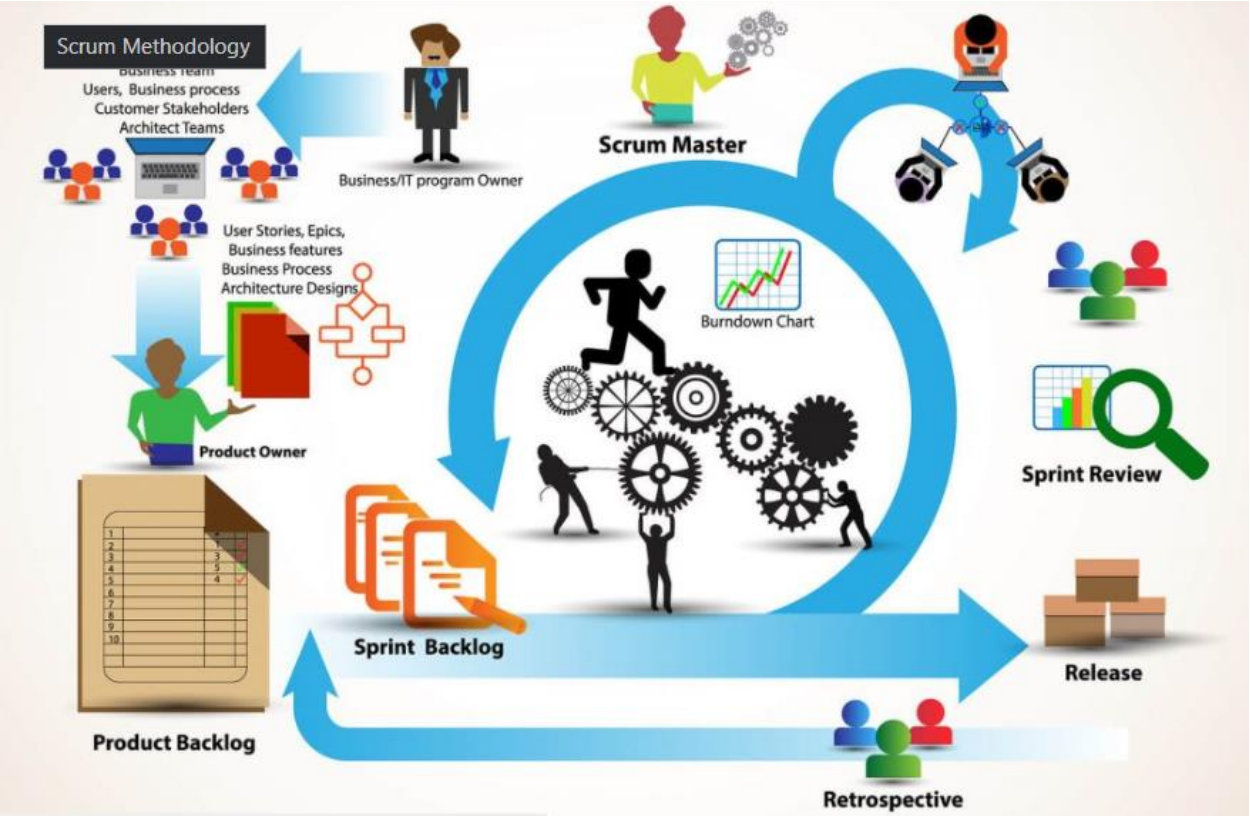
Agile Methodology

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholders. The most popular Agile frameworks are Scrum, Kanban, Extreme Programming (XP), and Adaptive Project Framework (APF).

Scrum

Scrum is an agile development methodology used in the development of Software based on an iterative and incremental process. Scrum is adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer throughout the development of the project. The primary objective of Scrum is to satisfy the customer's need through an environment of transparency in communication, collective responsibility and continuous

progress. The development starts from a general idea of what needs to be built, elaborating a list of characteristics ordered by priority (product backlog) that the owner of the product wants to obtain.



Technologies Used

WebRTC

Web Real-Time Communication aka WebRTC is a collection of Web APIs that allow developers to build audio, video, and generic data streaming applications over peer to peer connections within web browsers.

How does WebRTC work?

Under the hood WebRTC is just a collection of JavaScript APIs. Before WebRTC, most audio/video streaming applications were implemented using a client-server architecture. Where the browser is the client and it connects to a server of sorts to request data. The server then responds with a data stream.

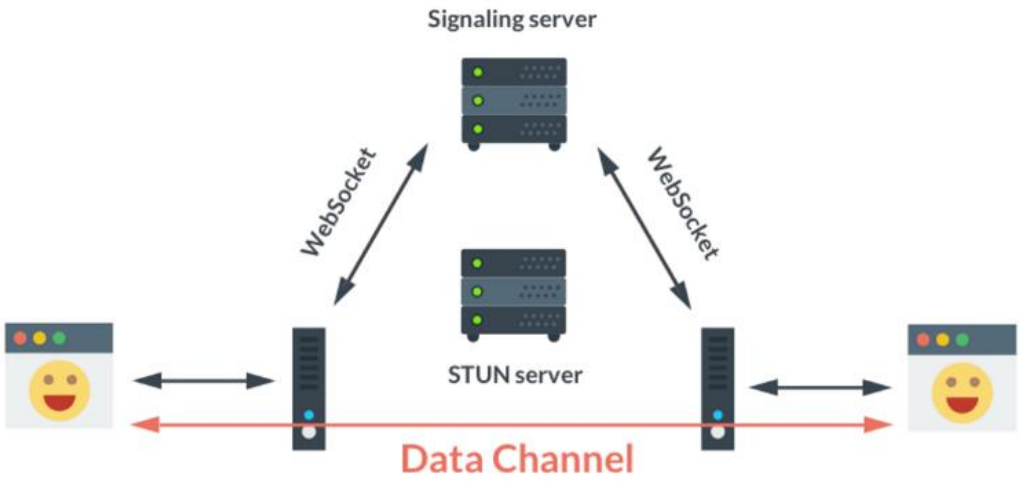
WebRTC sends data across browsers

It uses NAT traversal mechanisms for browsers to reach one another

Sometimes browsers need to connect to a relay server.

Separates media and signaling

Peer to Peer is the default approach, you can add servers to store and forward media data between peers.



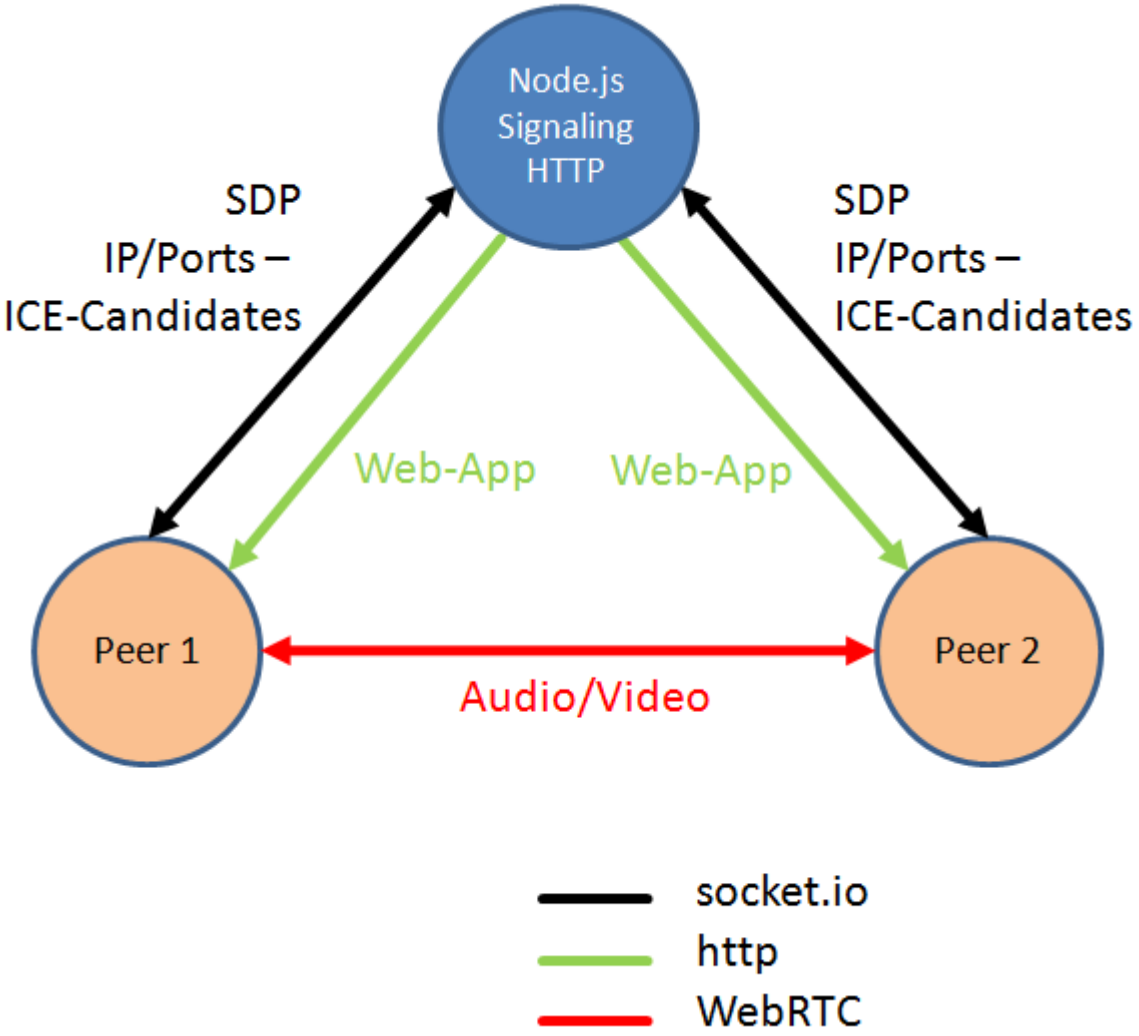
Sockets

WebRTC needs a signalling server for the peers to be able to send across Session Description Protocols (SDP). In this project, I've used sockets for signalling.

Socket.IO is a JavaScript library for realtime web applications. It enables realtime, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js. Both components have a nearly identical API. Like Node.js, it is event-driven.

Socket.IO primarily uses the WebSocket protocol with polling as a fallback option, while providing the same interface. Although it can be used as simply a wrapper for WebSocket, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O.

Implementation Design



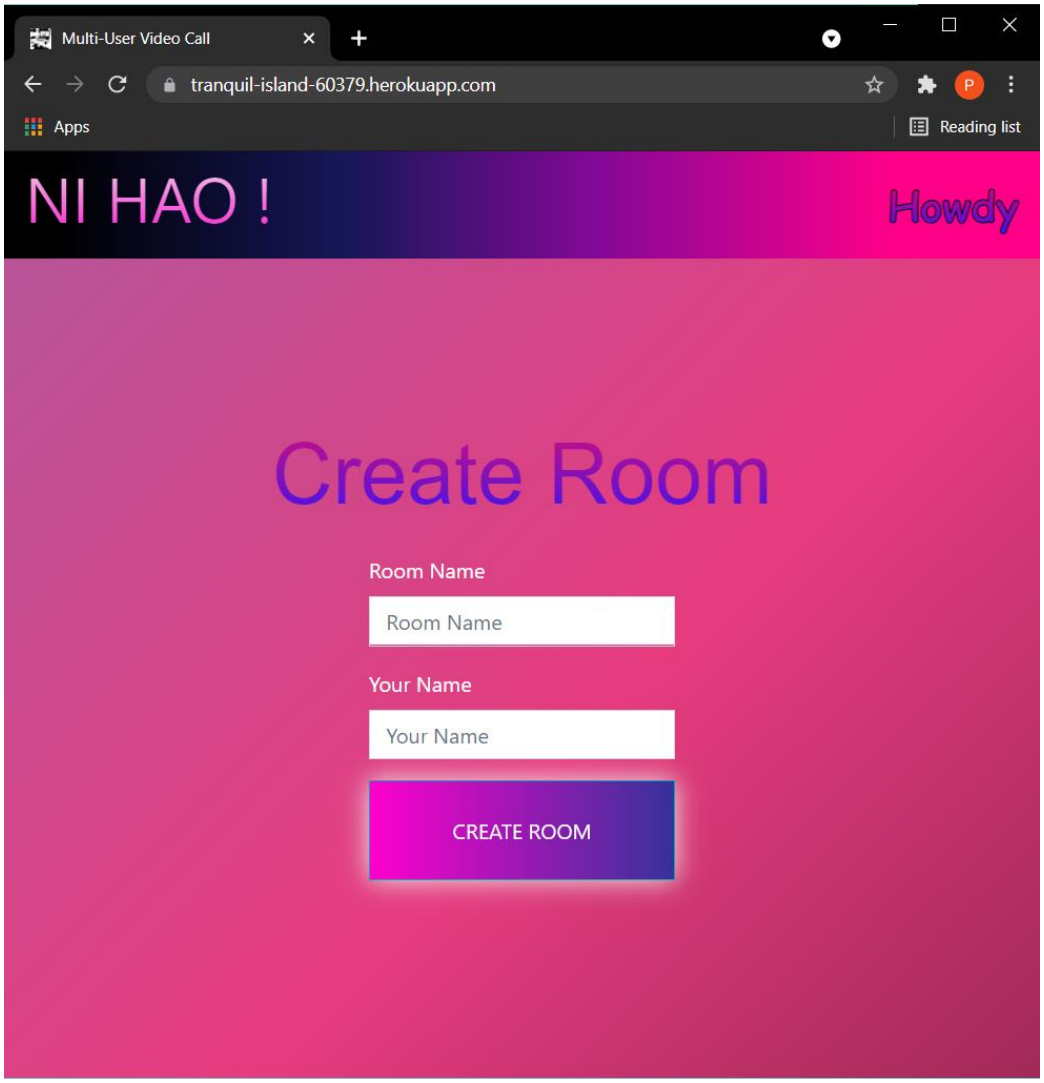
Features of Howdy

- Creating Rooms and sharing unique Room Id
- 1-1 Video Conference
- Group Video Conference
- Toggle Video (Enable/Disable)
- Toggle Audio (Mute/Unmute)
- Chat (1-1 / Group)
- Screen Sharing
- Screen Record
- Video Record

- Chat Notifications
- Mute Individual Participants
- Remote Stream Expand
- Pop-out Local Stream
- Leave Room

User-guide

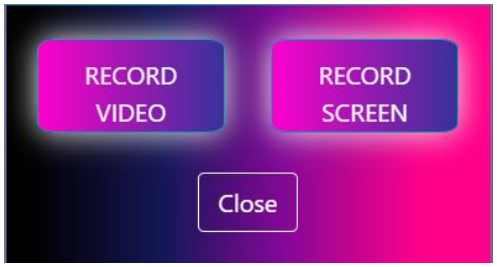
- One can use https://tranquil-island-60379.herokuapp.com/ to use the web app which has been hosted on Heroku Cloud platform.
- The link takes you to the following page



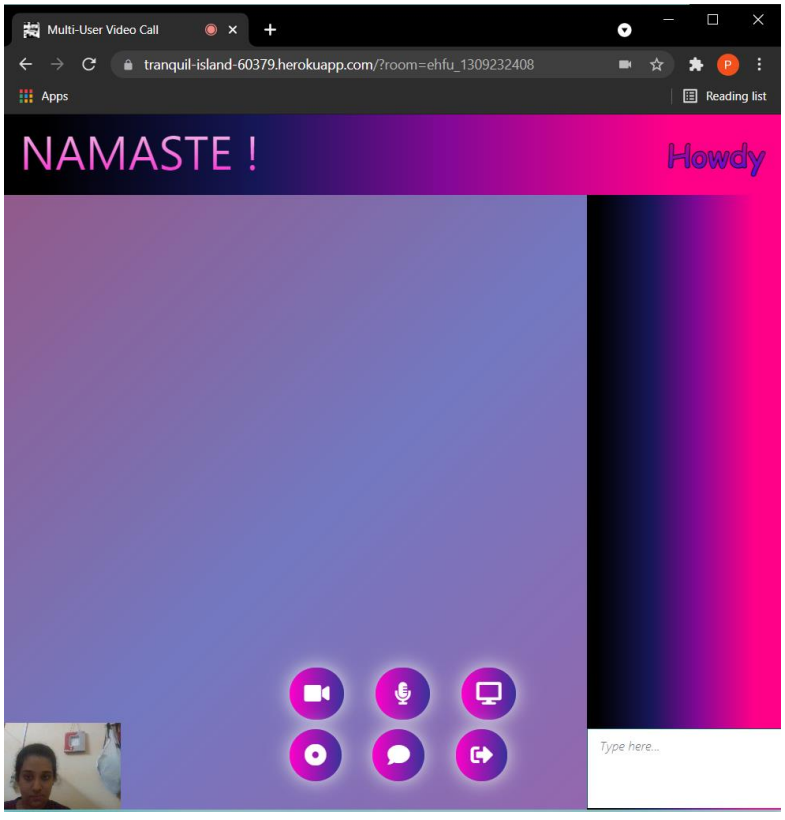
- You need to enter the credentials and click on “Create Room”
- Enter the room using the “Enter Room” button.
- Once in the room, share the Unique Room Id which is the URL of the page you are currently in to whoever you to join the meeting.

- You’ll have audio  and video  toggle buttons to enable and disable your streams.

- You can share your screen using the screen share  button
- You can record both the screen and also yourself using the record button  , clicking on which you get the options “record video” and “record screen”



- You can chat using the chat button  which opens the chat window



- Every time there's a new message, you get a notification badge over the chat icon 

.
- You can leave the room using the Leave button  , which directs you back to the Create Room page.

Limitations

- As I'm using WebRTC Peer to Peer Topology, it gets a little difficult for the application to accommodate more than 4-5 participants. It puts a huge burden on the bandwidth as well as the CPU. So, it is recommended to only have just 4-5 participants.
- Right now, this app works well only on Desktops and Tablets(Landscape Orientation).
- Cannot send Images in chat

Future Scope

- Scalability, by using Selective Forwarding Unit (SFU)
- Responsiveness
- Cross-platform
- Images and File Sharing

Hosting and Deployment Details

App is hosted on Heroku - https://tranquil-island-60379.herokuapp.com/

GitHub Repository - https://github.com/pujya-kvs/Howdy---Video-Conference-Web-Application-

References

- https://webrtc.org/
- https://levelup.gitconnected.com/building-a-video-chat-app-with-node-js-socket-io-webrtc-26f46b213017
- https://webrtc.org/getting-started/firebase-rtc-codelab
- https://youtu.be/FExZvpVvYxA