Internship at Felis Leo Ventures Private Limited

"Video Conferencing Application"

Submitted by:

Sreedhar K **PES1201700019**

Under the guidance of

Anirudh Jain

Product Head - Clique

Felis Leo Ventures

Internship duration : 3 months

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

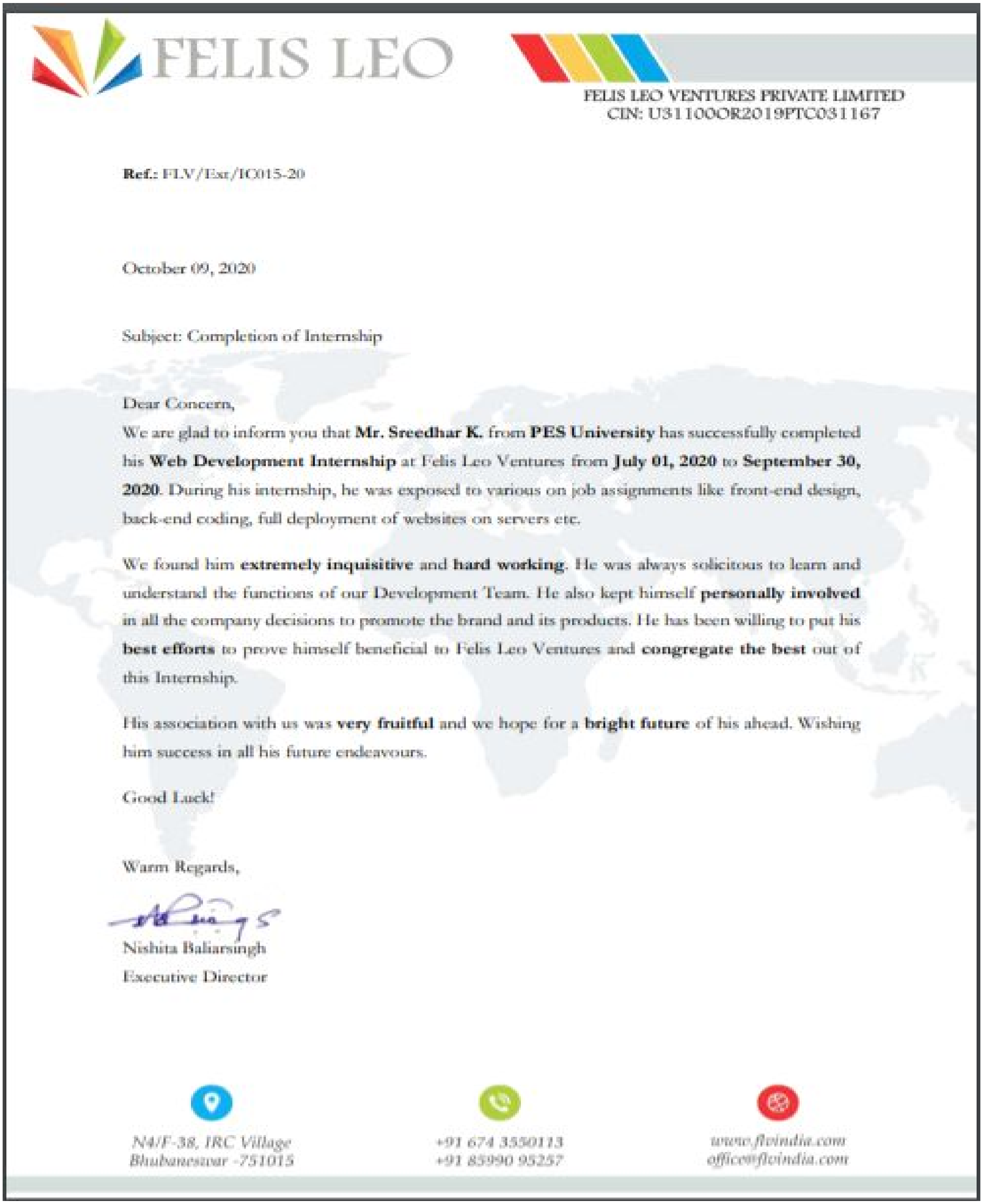
FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100ft Ring Road, Bengaluru – 560 085, Karnataka, India

COMPANY CERTIFICATE



FELIS LEO

FELIS LEO VENTURES PRIVATE LIMITED
CIN: U31100OR2019PTC031167

Ref.: FLV/Ext/IC015-20

October 09, 2020

Subject: Completion of Internship

Dear Concern,

We are glad to inform you that **Mr. Sreedhar K.** from **PES University** has successfully completed his **Web Development Internship** at Felis Leo Ventures from **July 01, 2020** to **September 30, 2020**. During his internship, he was exposed to various on job assignments like front-end design, back-end coding, full deployment of websites on servers etc.

We found him **extremely inquisitive** and **hard working**. He was always solicitous to learn and understand the functions of our Development Team. He also kept himself **personally involved** in all the company decisions to promote the brand and its products. He has been willing to put his **best efforts** to prove himself beneficial to Felis Leo Ventures and **congregate the best** out of this Internship.

His association with us was **very fruitful** and we hope for a **bright future** of his ahead. Wishing him success in all his future endeavours.

Good Luck!

Warm Regards,

Nishita Baliarsingh
Executive Director

N4/F-38, IRC Village
Bhubaneswar -751015

+91 674 3550113
+91 85990 95257

www.flvindia.com
office@flvindia.com

DECLARATION

We hereby declare that the project entitled **“Video Conferencing Application”** has been carried out at **Felis Leo Ventures** by **Sreedhar K** under the guidance of <Mr. Anirudh Jain, Designation> and submitted in partial fulfillment of the credits for the degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester(7th semester(lockdown)) for **3 months**. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

SRN - PES1201700019

NAME - SREEDHAR K

SIGNATURE -

Sreedhar K

ACKNOWLEDGEMENT

I would like to express my gratitude to our guide Mr. Anirudh Jain, Product Head - Clique ,Felis Leo Ventures, for his continuous guidance, assistance and encouragement throughout the development of this project.

I am grateful to the internship coordinator,Prof. Preet Kanwal, Dept. of Computer Science and Engineering , PES University for organizing, managing and helping out with the entire process.

I take this opportunity to thank Dr. Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department.

I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

**I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof.
Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way.**

Finally, this internship could not have been completed without the continual support and encouragement I have received from my parents and my teammates of this internship.

1. ABSTRACT

Our project, titled “Video Conferencing Application” deals with integrating multiple users through a Zoom-like application. Some of the project features -

- Audio/Video Access (by asking permission to enable), and the users have the freedom to mute their video/audio
- Meeting Recording option, screen sharing option
- General settings (for audio, video etc.)
- Host has the right to permit users in
- Chat space, attendee list
- Option to enter full-screen mode
- Button to exit the meeting
- The application allows a maximum of 150 participants in one room.
- The interface only requires the room code/link for entering the video conference.

2. SCOPE

This project can be used as a video conferencing component for any larger projects in future. We have also uploaded the entire project-related code in github. We can also increase the number of participants in future (greater than 150).

3. TABLE OF CONTENTS

4. INTRODUCTION

Our project, titled “**Video Conferencing Application**” deals with integrating multiple users through a Zoom-like application. Some of the project features -

- Audio/Video Access (by asking permission to enable), and the users have the freedom to mute their video/audio
- Screen Sharing option
- Meeting Recording option
- General settings (for audio, video etc.)
- Host has the right to permit users in
- Chat space
- Attendee list
- Option to enter full-screen mode
- Button to exit the meeting
- The application allows a maximum of 150 participants in one room.
- The interface only requires the room code/link for entering the video conference.

For the front-end of our Video Conferencing Application, we have used Bootstrap and basic Javascript, and for the back-end, we have used Node.js. The project that we have performed is currently a desktop-based application, meaning that it can work fine on all our desktops/ local servers. It would also be hosted on a web-server.

5. COMPANY BRIEF INTRODUCTION

Alpha Annexus, later renamed as Felis Leo Ventures, started off with a concept to provide complete marketing solutions to companies so that they do not have to bother about the marketing aspect. The company currently offers its client base continuous assistance regarding product marketing, including outdoor events, media promotions, sponsorship, website development, social media marketing etc.

6. PROJECT OVERVIEW

Project title - Video Conferencing Application

This is not a part of a bigger project, rather, a complete project on its own since it has the entire front-end and back-end up and running.

I mostly worked on the backend, wherein we had to enable multiple connections into a single video room with the help of the room id.

For this purpose, we made use of RTCMultiConnection.js(WebRTC JavaScript library for peer-to-peer applications (screen sharing, audio/video conferencing, file sharing, media streaming etc.)). We have used quite a lot of APIs from RTCMultiConnection.js (complete list of APIs can be found here).

We have also used Socket.IO in our project. Socket.IO enables real-time, bidirectional and event-based communication.It works on every platform, browser or device, focusing equally on reliability and speed.

RTCMultiConnection.js and Socket.IO are available as modules in NPM(Node Package Manager). So, in order to work with them, we first installed these 2 modules. These two modules are - socket.io and rtcmulticonnection .

7. PROJECT DESIGN DETAILS WITH TECHNOLOGIES USED

The technology predominantly used is Node.js. The design details of the project include the .html files and the styles we have used in them.

They are listed below (files are in github).

- audio-conferencing.html - Multi-user (many-to-many) audio-calling using mesh networking model.
- call-by-username.html - To Call Any Person By His UserName.
- camera-zoom.html - Used To Zoom Camera during WebRTC Live Streaming
- file-sharing.html - Used For Peer-to-Peer (private) file sharing. You can share/receive files from any platform/device e.g. desktop operating systems, Android, iOS etc.
- getStats.html - Using WebRTC getStats API to detect data sent/received, packets lost/success, ports/network, encryption and more.
- index.html - The main page of our project.
- one-to-one.html - Wherein maximum two users can join a room.

- ○ Scalable-Broadcast.html - Use peer-to-peer protocol to broadcast your video over 20+ users.
- ○ screen-sharing.html - Multi-user (one-to-many) screen sharing using star topology.
- ○ SSEConnection.html - This demo is using SSE (Server Sent Events) to set up WebRTC one-to-one connections.
- ○ text-chat-file-sharing.html - TextChat+FileSharing using RTCMultiConnection
- ○ translate-text-chat.html - TextChat Translation using RTCMultiConnection
- ○ video-and-screen-sharing.html - Multi-user (many-to-many) video chat using mesh networking model. We can add and remove screens anytime and multiple times!
- ○ video-broadcasting.html - Multi-user (one-to-many) video broadcasting using star topology.
- ○ video-conferencing.html - Multi-user (many-to-many) video chat using mesh networking model.
- ○ video-conferencing-chat-filesharing.html - Multi-user (many-to-many) video streaming + text chat + file sharing using mesh networking model.
- ○ vuejs-video-conferencing.html - Multi-user (many-to-many) video chat using mesh networking model.

For styling, we used Bootstrap, and jQuery for cross-browser compatibility.

8. CODING/IMPLEMENTATION DETAILS(MODULES USED TO ACHIEVE THE FUNCTIONALITY)

Here we would take a look at the backend functionality of our project.

- For using socket.io, we used socket.io.js .
- To link RTCMultiConnection.js script file,

❏

```
<!--html-->
<script src="/dist/RTCMultiConnection.min.js"></script>
```

❏

```
<!-- Heroku Link -->
<script src="http://rtcmulticonnection.herokuapp.com/dist/RTCMultiConnection.min.js"></script>
```

❏

<!-- or specific version-->
<script src="https://github.com/muaz-khan/RTCMultiConnection/releases/download/3.4.3/RTCMultiConnection.js"></script>

❏

<!-- or CDN-->
<script src="https://rawgit.com/muaz-khan/RTCMultiConnection/master/dist/RTCMultiConnection.min.js"></script>

- If you're sharing files, you also need to link:

  ❏ <!--html-->
  <script src="/dev/FileBufferReader.js"></script>

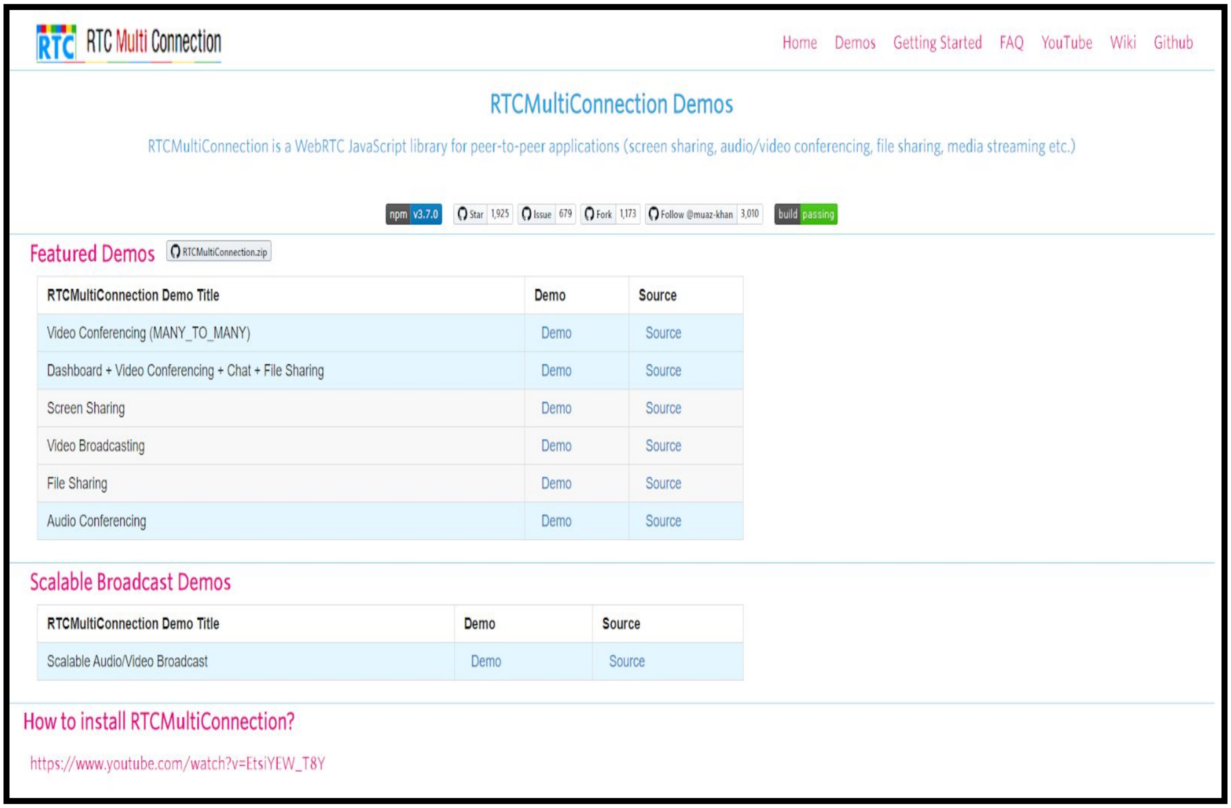
  ❏ <!-- or CDN -->
  <script src="https://cdn.webrtc-experiment.com:443/FileBufferReader.js"></script>

- Modules used(all from npm) :
  - rtcmulticonnection-server -
    const RTCMultiConnectionServer = require('rtcmulticonnection-server');
  - path - const path = require('path');
  - url - const url = require('url');
  - http - var httpServer = require('http');
  - socket.io -
    const ioServer = require('socket.io');

9. PROJECT RESULTS/LEARNING OUTCOMES

This is the index page of our project.

index.html



RTCMultiConnection Demos

RTCMultiConnection is a WebRTC JavaScript library for peer-to-peer applications (screen sharing, audio/video conferencing, file sharing, media streaming etc.)

Featured Demos

RTCMultiConnection Demo Title	Demo	Source
Video Conferencing (MANY_TO_MANY)	Demo	Source
Dashboard + Video Conferencing + Chat + File Sharing	Demo	Source
Screen Sharing	Demo	Source
Video Broadcasting	Demo	Source
File Sharing	Demo	Source
Audio Conferencing	Demo	Source

Scalable Broadcast Demos

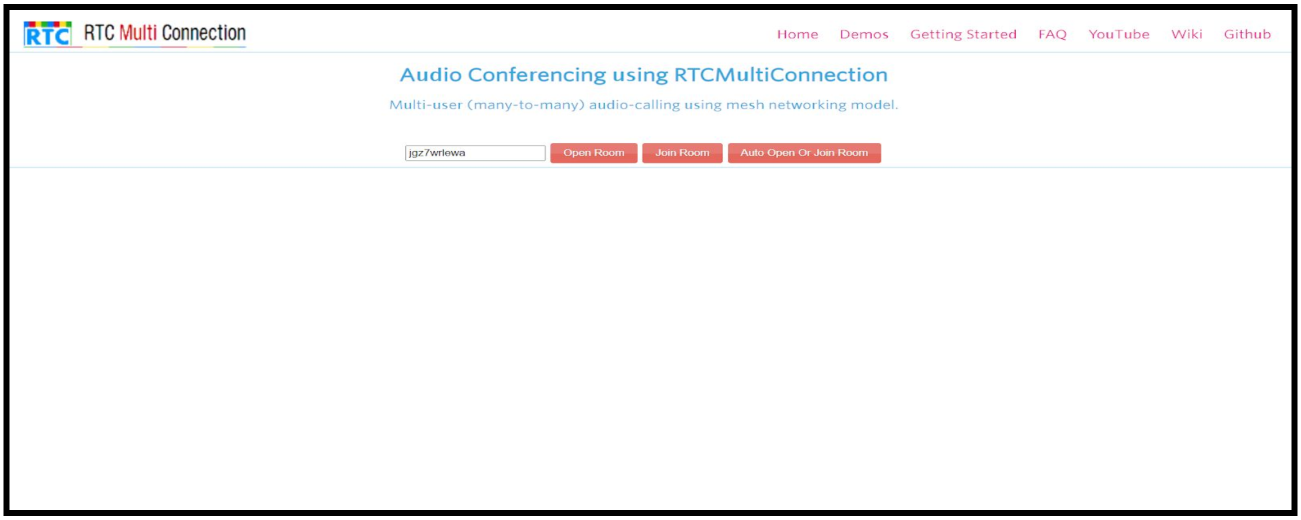
RTCMultiConnection Demo Title	Demo	Source
Scalable Audio/Video Broadcast	Demo	Source

How to install RTCMultiConnection?

https://www.youtube.com/watch?v=EtsiYEW_T8Y

Most of the below pages have a similar layout(UI), only the backend implementation is different depending on their functions.

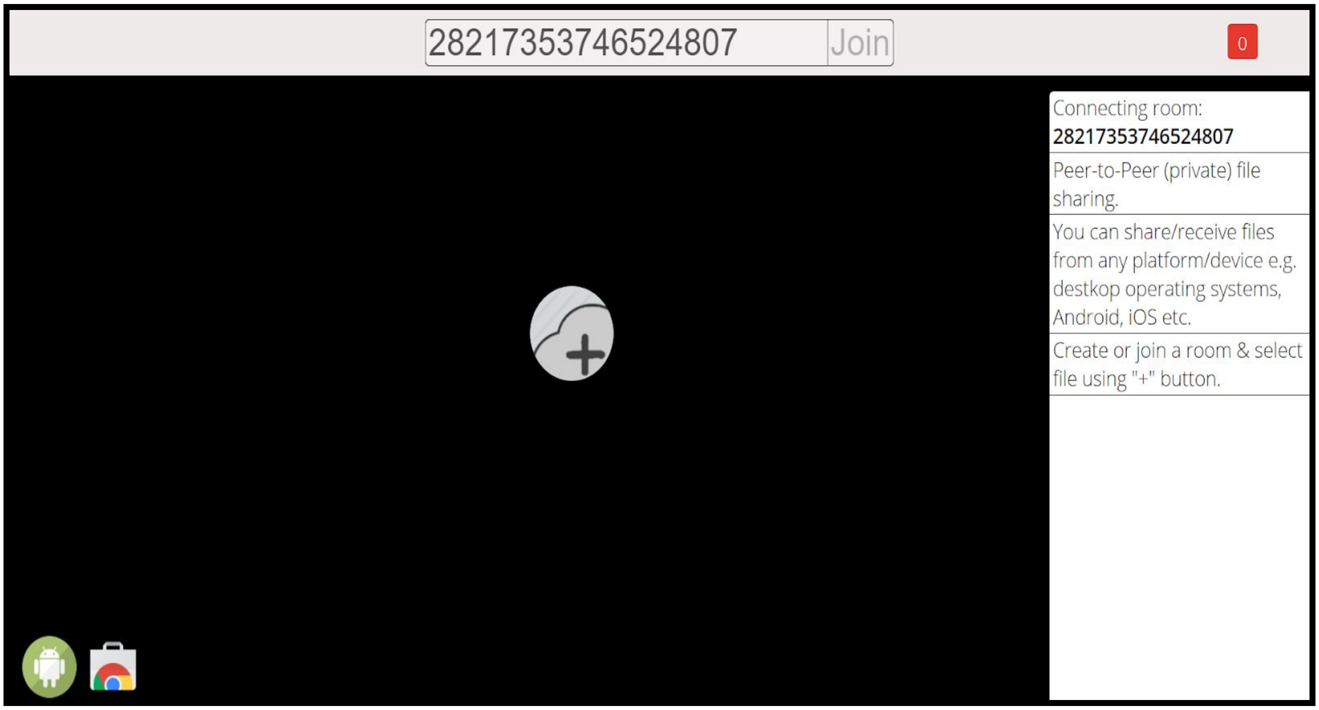
audio-conferencing.html



Audio Conferencing using RTCMultiConnection

Multi-user (many-to-many) audio-calling using mesh networking model.

file-sharing.html



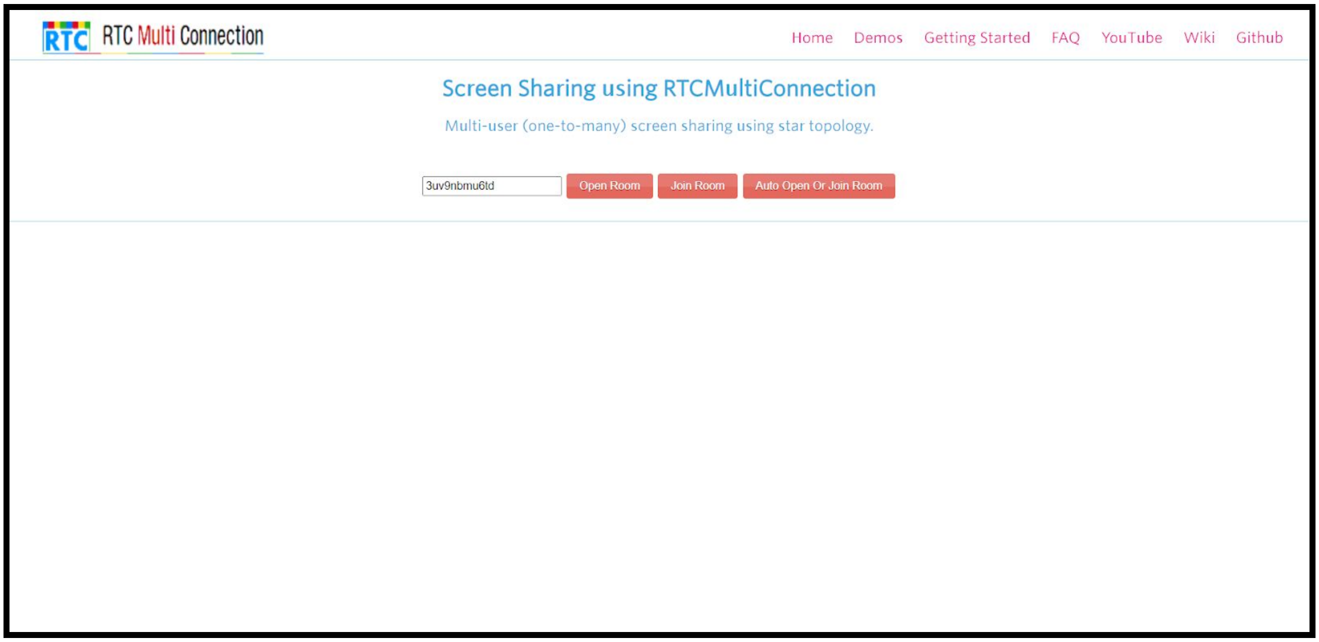
Connecting room: **28217353746524807**

Peer-to-Peer (private) file sharing.

You can share/receive files from any platform/device e.g. destkop operating systems, Android, iOS etc.

Create or join a room & select file using "+" button.

screen-sharing.html

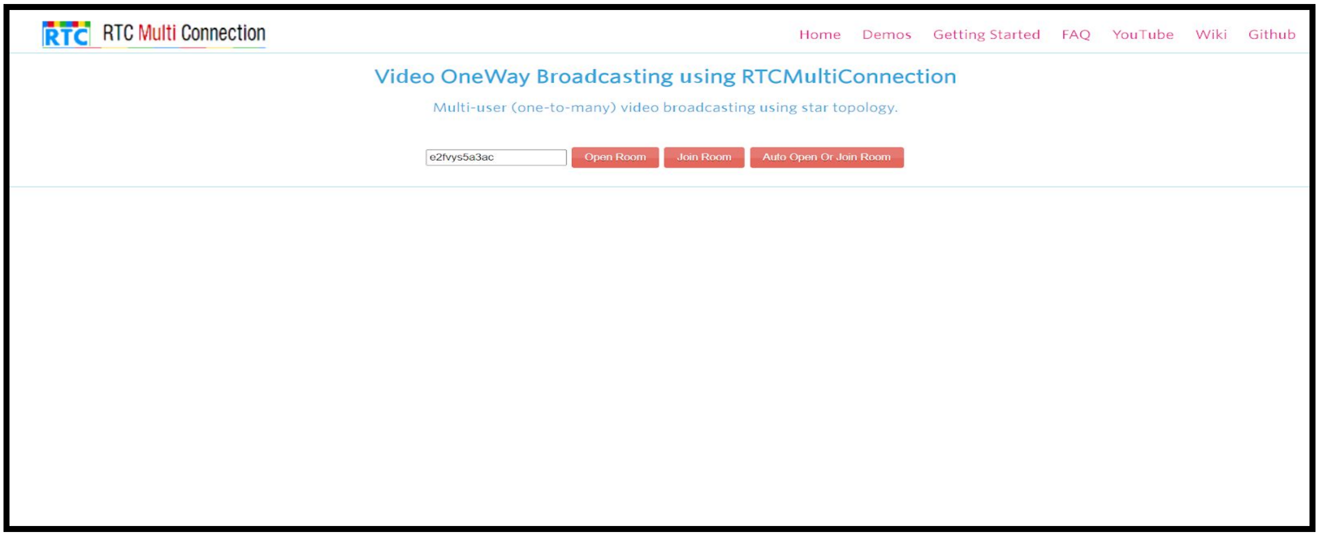


Screen Sharing using RTCMultiConnection

Multi-user (one-to-many) screen sharing using star topology.

3uv9nbmu6td

video-broadcasting.html

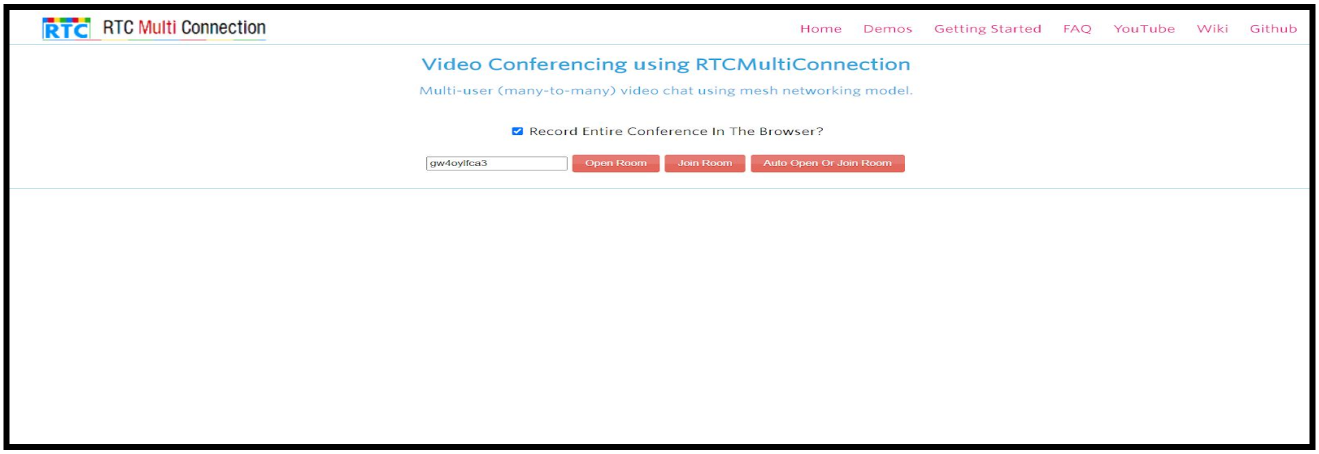


Video OneWay Broadcasting using RTCMultiConnection

Multi-user (one-to-many) video broadcasting using star topology.

e2fvys5a3ac

video-conferencing.html



Video Conferencing using RTCMultiConnection

Multi-user (many-to-many) video chat using mesh networking model.

☑ Record Entire Conference In The Browser?

gw4oylfca3

Throughout this project, I have learnt a lot about how to set up multiple connections using Node.js, how to design even the front-end for this project, and also how to use various npm modules.

10. CONCLUSION

This application that we have developed so far is a desktop-based application. We can as well deploy it on the server to make it accessible to everyone. We tried to keep the front-end minimal , and the majority of our focus was on the back-end.

Overall, we got a very wonderful experience working on a project which most of us were very unfamiliar with, but gradually supported each other, and learnt a lot of new concepts that could be of great use to us in future projects as well. Our guide and also my teammates have been extremely supportive during the course of this internship.

11. REFERENCES

NPM

RTCMultiConnection

Github

Socket.IO

cdnjs

Bootstrap

jQuery

THANK YOU