

TECHNICAL DOCUMENT

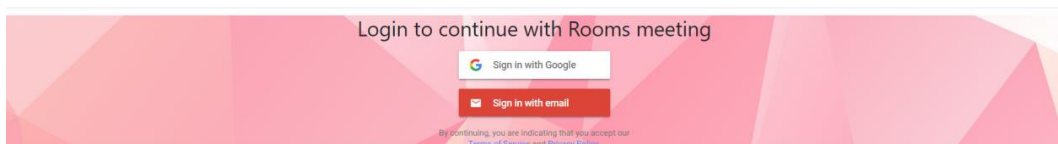
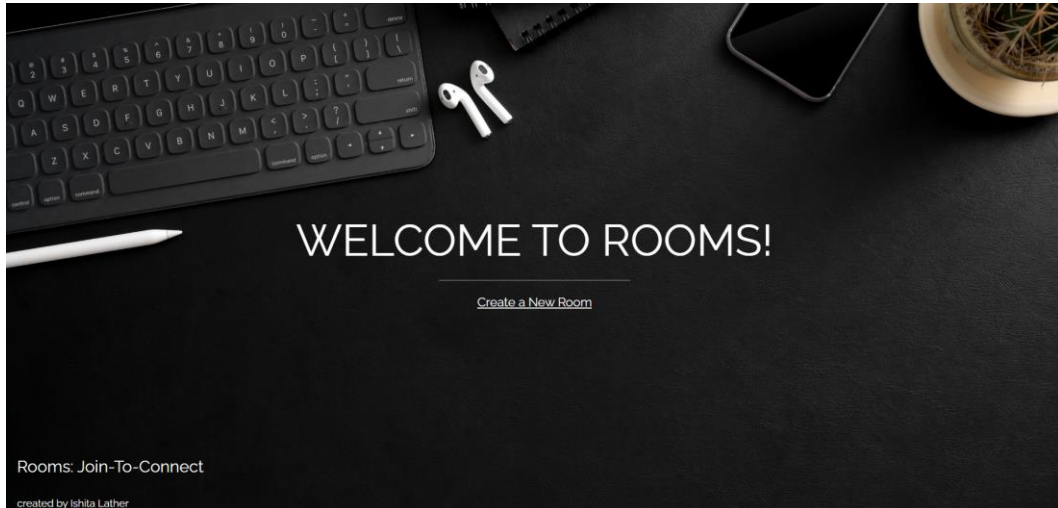
I would like thank Microsoft for giving me this opportunity and would like to express my deep and sincere gratitude to my mentors, Shriladha Balasubramanian and Srinivasan Rajagopalan. Their insights have been invaluable and their constant support and motivation has helped me complete this project successfully.

ROOMS: JOIN TO CONNECT

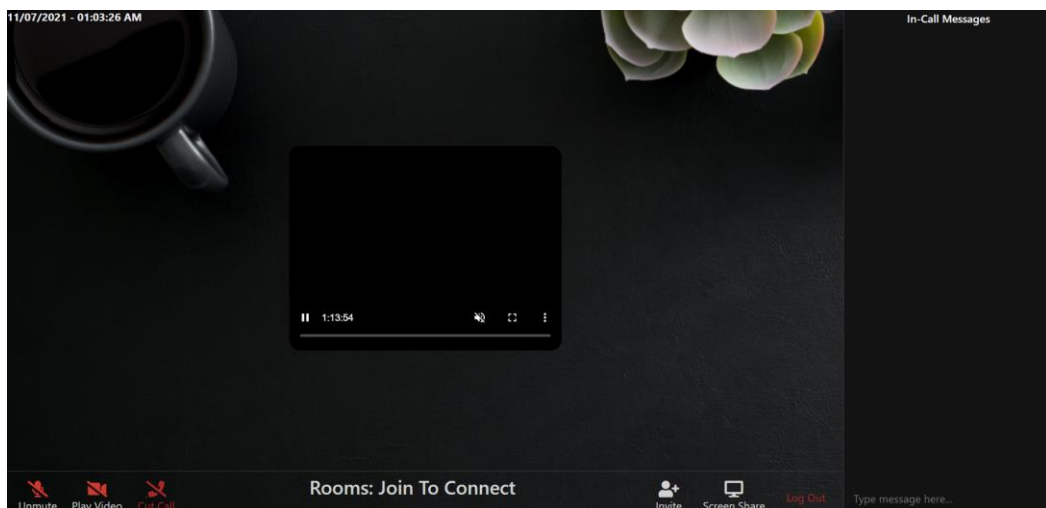
<https://rooms-join-to-connect.herokuapp.com/>

<https://github.com/ishita3008/video-conferencing-app.git>

Join Rooms to experience a seamless video call to connect with your friends, family and colleagues.



[Go back to home](#)



FEATURES:

- VIDEO AND AUDIO CONNECT
- USER AUTHENTICATION
- MUTE/UNMUTE
- VIDEO ON/OFF
- IN CALL MESSAGES
- SCREEN SHARE (WITH OPTION TO SHARE SCREEN, WINDOW, TAB)
- VIDEO CONTROLS
- DATE AND TIME DISPLAY
- INVITE MORE THAN 2 PEOPLE
- HOMEPAGE WITH A CREATE A NEW ROOM
- CUT CALL
- LOG OUT

BACKEND:

- NODE JS: It is a runtime environment for executing JavaScript, it is used for highly scalable and real-time apps, is great for prototyping and agile development.
- EXPRESS: It is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
- SOCKET.IO: It enables real-time, bi-directional communication between web clients and servers.
- UUID: It helps us create URLs with different room numbers inside of them so that we can have users to call in specific rooms.
- WEBRTC: WebRTC is a free and open-source project providing web browsers and mobile applications with real-time communication via simple application programming interfaces.
- PEER JS: PeerJS wraps the browser's WebRTC implementation to provide a complete, configurable, and easy-to-use peer-to-peer connection API.
- FIREBASE: It is a platform (backend as a service) developed by google for creating web and Mobile apps. It provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app.
- NODEMON: Nodemon is a tool that helps develop node. js based applications by automatically restarting the node application when file changes in the directory are detected.

FRONTEND:

- HTML: It provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript.
- CSS: is used to control presentation, formatting, and layout.
- JavaScript: is used to control the behaviour of different elements.
- EJS: It is a simple templating language that lets you generate HTML markup with plain JavaScript.
- BOOTSTRAP: It is a potent front-end framework used to create modern websites and web apps.
- FONT AWESOME: It is used to add font icons to your website. Font Awesome icons are created using scalable vectors, so you can use high quality icons that work well on any screen size.

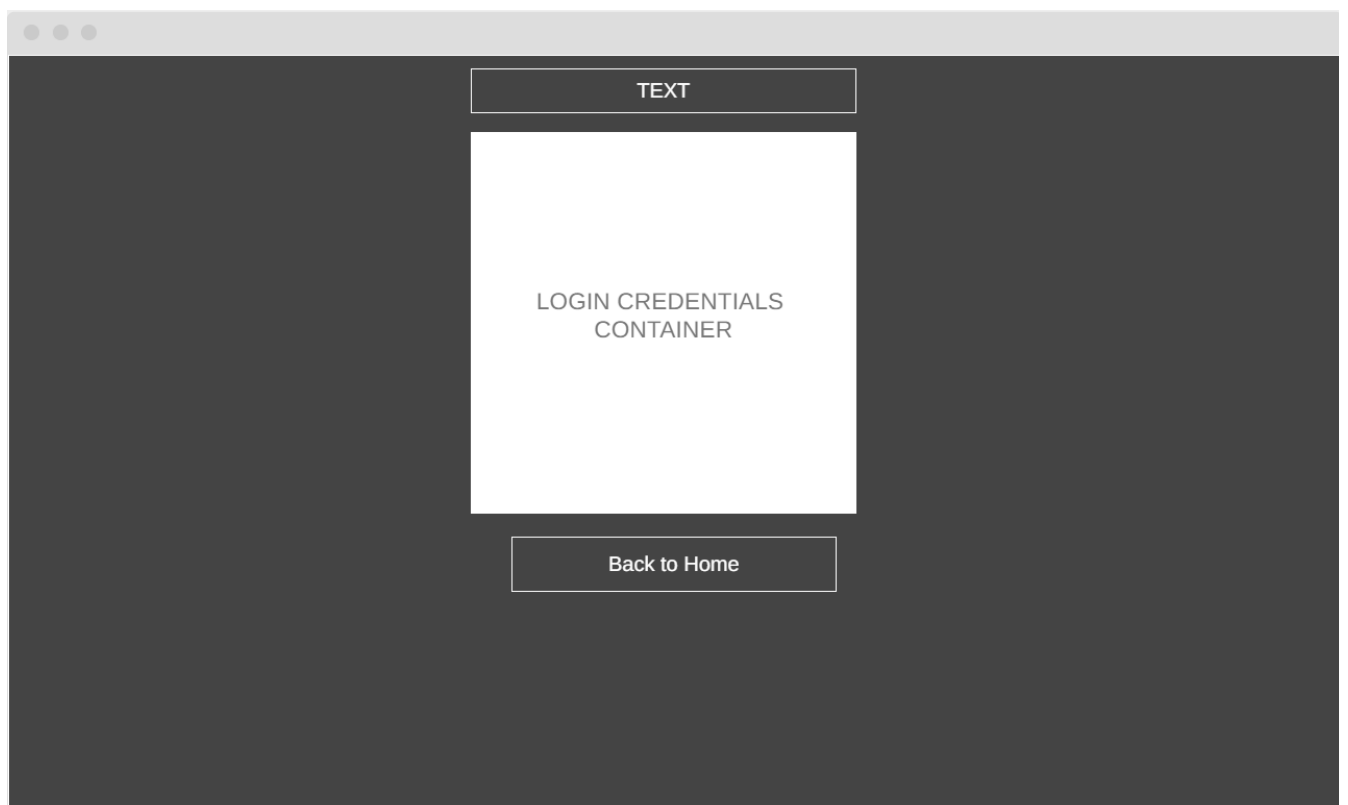
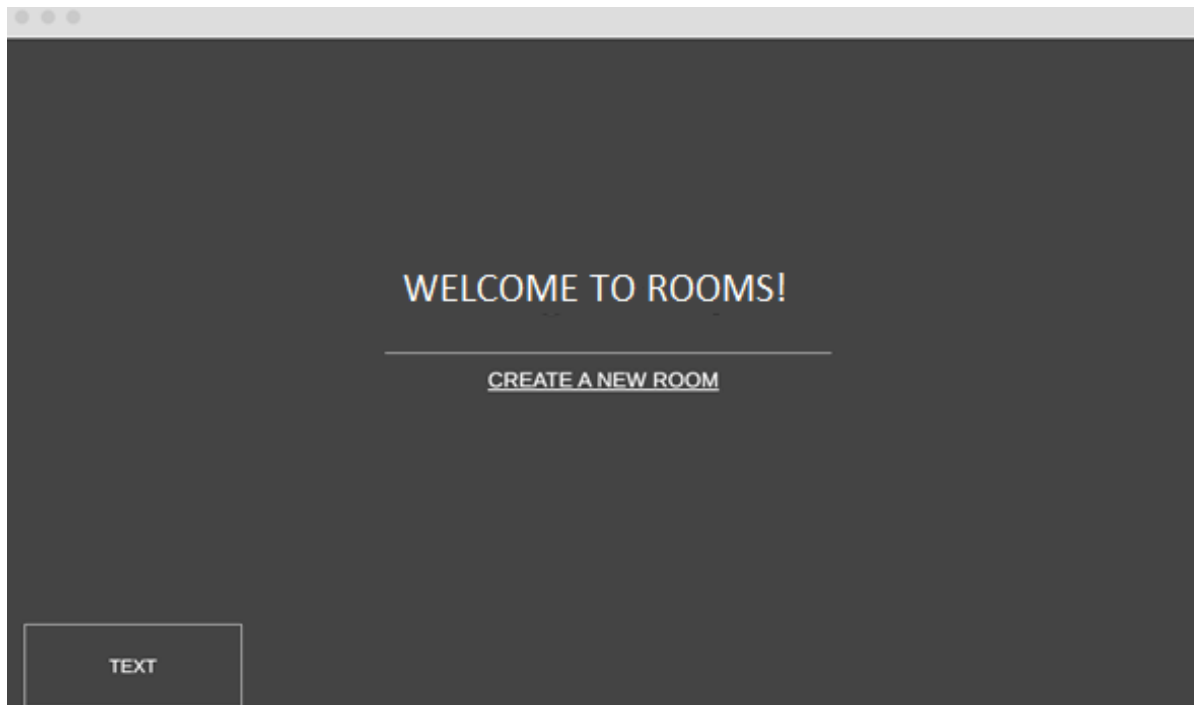
HOW TO RUN ON LOCAL MACHINE:

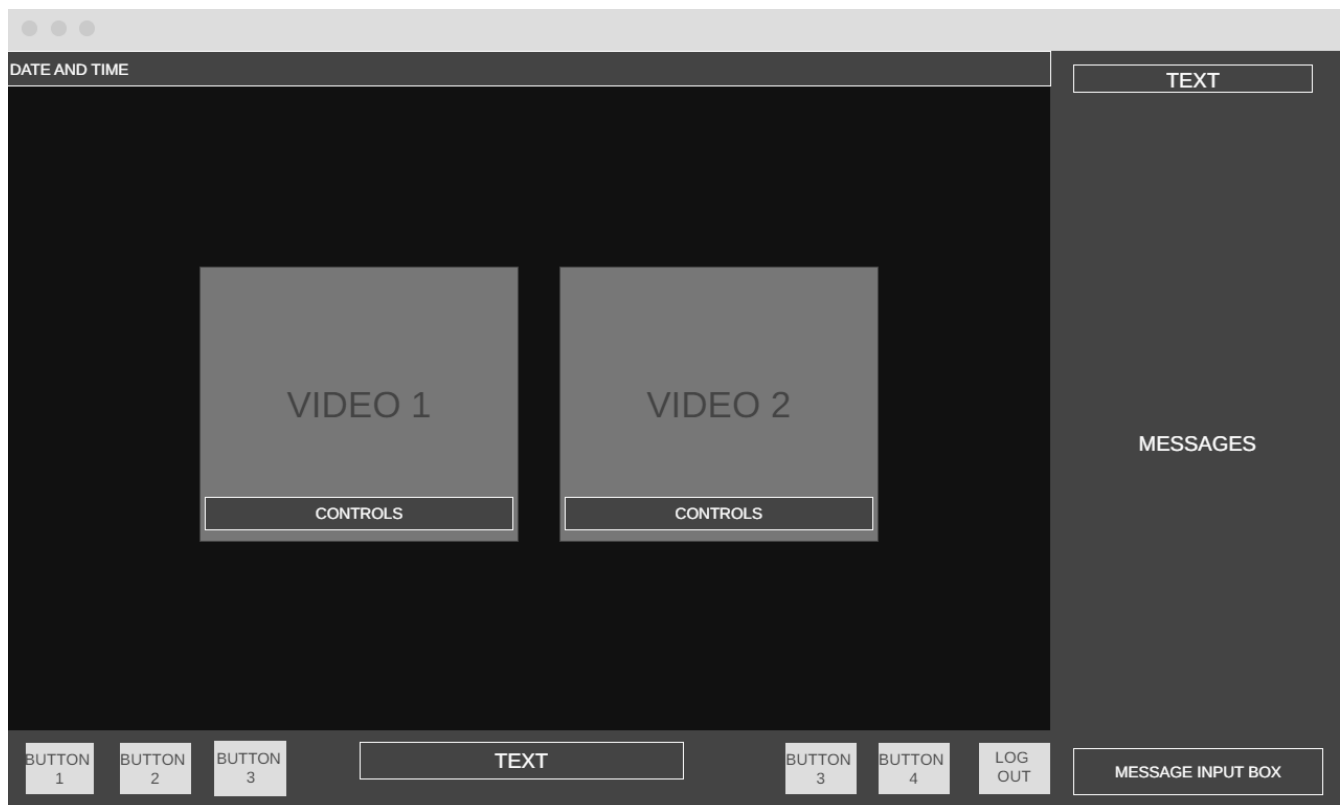
```
npm install
```

```
npm nodemon server.js
```

```
run on http://localhost:3010/
```

WIREFRAMES:





HOW TO USE ROOMS:

- First up open Rooms: Join to Connect using <https://rooms-join-to-connect.herokuapp.com/> , and click on **Create a New Room** to be directed to a login page.
- Sign in using your email id and password or using your google account, if using for the first time, you'll be required to create a new account.
- Once logged in, you will enter the unique room just made for you and people you want to invite, you'll be able to see the **Date and Time** at the top of the page and can keep track of your meeting.
- By default, your audio and video will be switched off, you can choose to unmute and switch on your video using **"Mute/Unmute"** and **"Video On/Off"** buttons.
- Click on the **Invite button** to get the meeting link and share it with people you want to talk with.
- As soon as the person clicks on the link provided by you, they will be asked to enter in their name and will be redirected to your unique room and both of you will be **connected by audio and video channels**.
- Hovering over the videos, you can see **Video Controls** and can zoom in or mute the other person for yourself.
- While in the call, you and the person you are talking with can send and receive **In Call Messages**. To send a message, you have to type the message in the input container and press enter. Messages sent by you will be sent along with the name you entered in the prompt.
- You can also choose to share your screen using the **Screen Share** button on the right. You will be given the option to either share your:

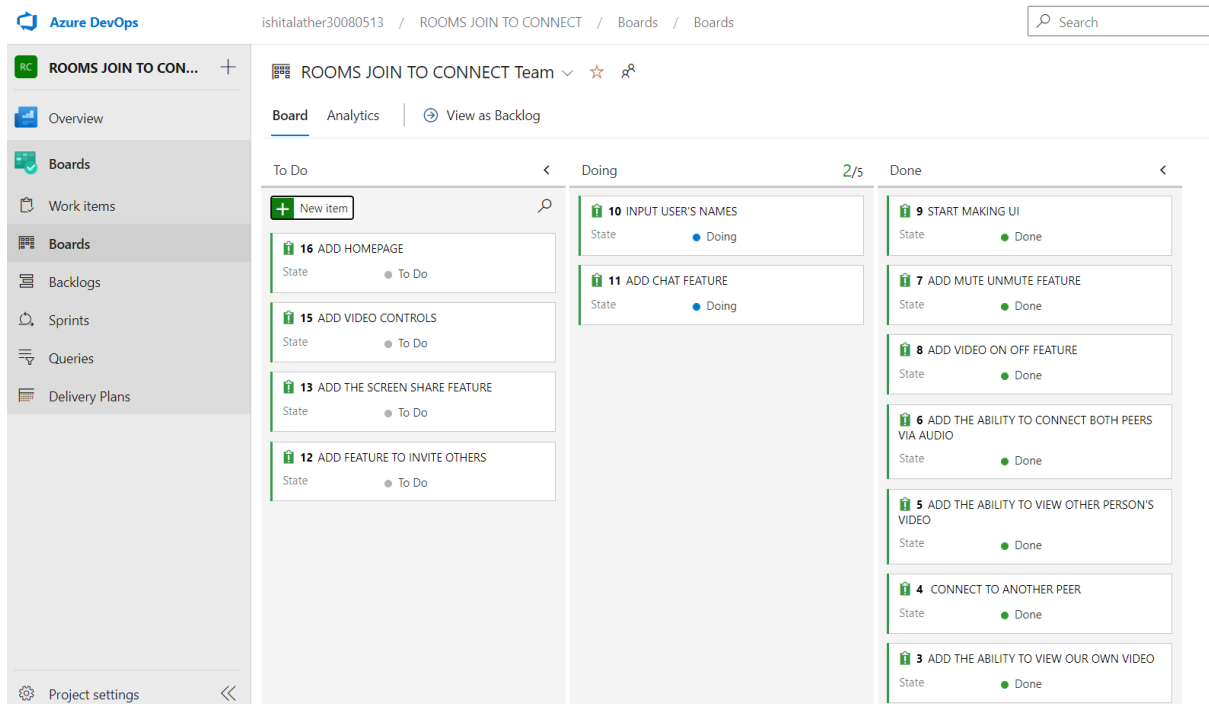
- ✓ entire screen
 - ✓ window
 - ✓ or just a single tab.
- As soon as you share your screen, for the other person, your camera video will be replaced by the screen stream. The other person can zoom in to watch your full screen using video controls.
 - Once you click on Stop Sharing displayed at the footer of your screen, your screen share will be stopped and your stream would be replaced by your camera video.
 - You can invite more than 2 persons as well in your call.
 - When you want to end the call, you can click on the **Logout** button to cut the call and log out and you will be redirected back to the login page, you will have the option to return back to the meeting by logging back in or you can simply choose to go to the home page.
 - If you close the window simply by the **End Call** button, you will still be logged in the website and can directly enter the meeting next time with the same credential.

AGILE METHODOLOGIES USED:

1. I categorized the whole app into **theme, epics, user stories and tasks** and thereafter planned the work flow into **4 sprints** and worked accordingly.
2. I kept on noting down the **bugs and issues** I encountered and the ones which I was able to resolve, jotted down under **closed issues** and the ones I am still working on figuring out, I have put them under **open issues**.
3. Further, I have kept the ideas and features I wish to implement in the **future under the backlog** table with proper segregation into epics, user stories and tasks.

The detailed agile approach I followed: https://1drv.ms/x/s!AuF_GqhwqqsxoHzTYqMtRk0upvIW?e=gceTC8

4. I also kept a track of the work To Do, the work Done and the work I was currently doing during all these 4 sprints using the **Scrum Board**.



5. Throughout the project, I followed the CI CD approach. With Continuous Integration Continuous Delivery, I kept on adding new deliverables to the project and continuously deployed the app, new features were added to the application but at no point the service was taken down.

FUTURE IMPROVEMENTS (BACKLOG):

1. REAL TIME OFFICE/SCHOOL FEELING

In this pandemic, we miss walking around our college/office building, meeting different people in the corridors. Now, we usually just talk to people who are in our team/class but in office/school we used to randomly meet people just because they were walking towards us or something like that

I want to add a feature which could give you real time office feeling. We can have a virtual tour of the campus using our cursor, going to different areas of the campus and meeting other people in the process just by luck because they were also coming that way, exactly what happens in real life and then as soon as two people are near each other their chat box opens, they can either chat or greet or even do a video call or just choose to move ahead.

This way one can roam around sitting in front of the computer screen and also get the office feeling and as soon as your class/meeting begins, you will be directed to it, in case someone loses track of time and keeps on roaming.

2. ADDING GAMES TO THE APP

I want to add certain games like Tic Tac Toe and Pictionary (with a random word generator) to the app.

3. ADDING A WHITEBOARD

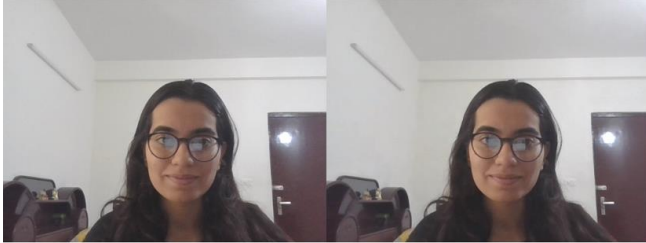
I wish to add a collaborative whiteboard in the app that helps remote teams ideate and collaborate online.

STEP WISE PROGRESS:

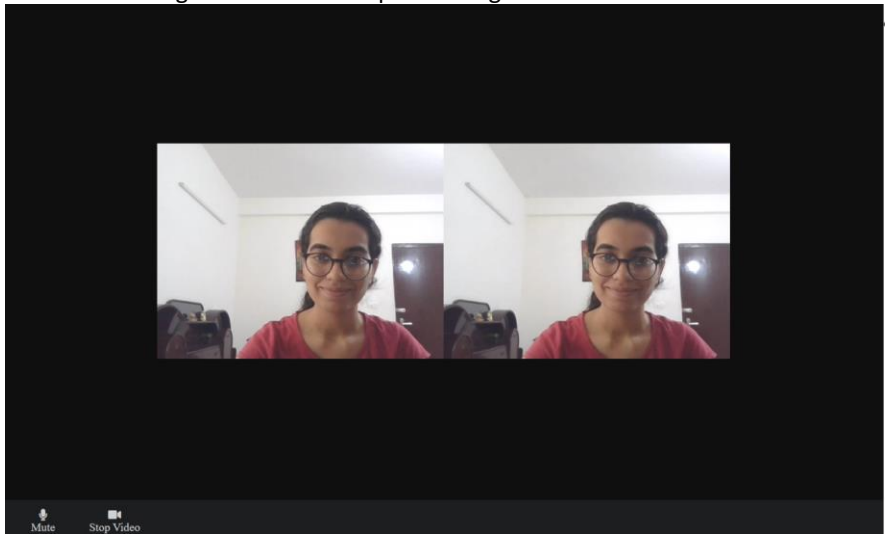
- Single Video Display



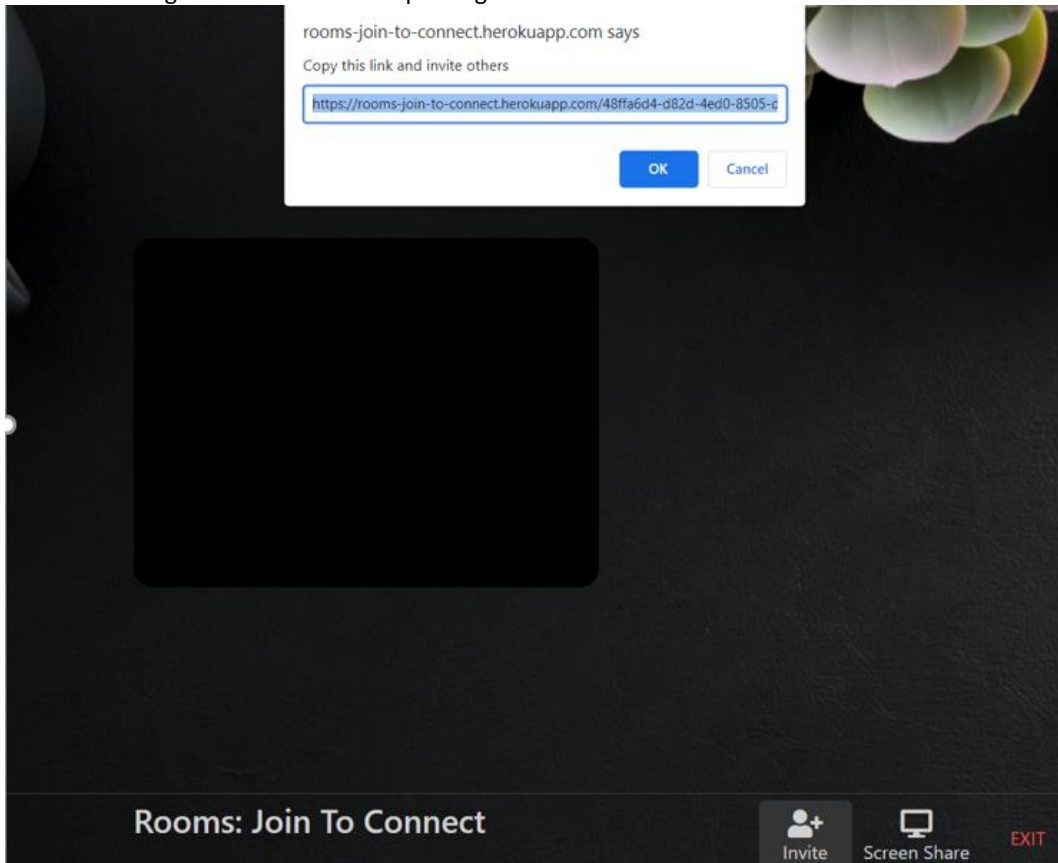
- Peer to Peer Connection established, both videos visible



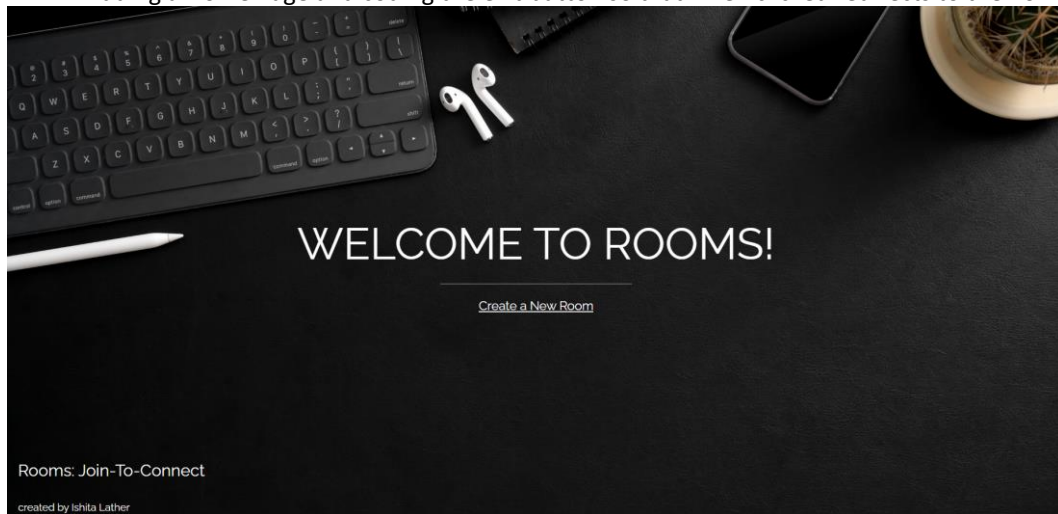
- Building a basic UI and implementing Mute Unmute and Video On Off



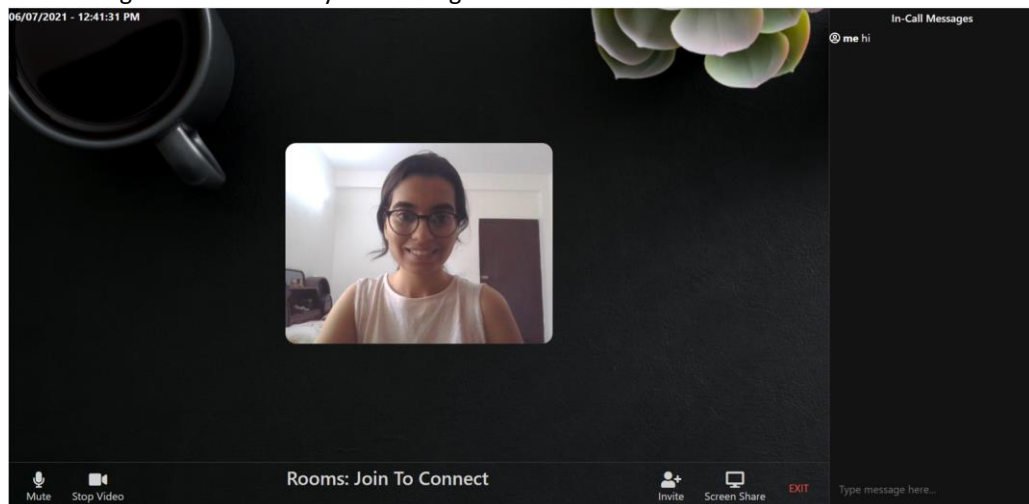
- Adding Invite Button and improving the UI



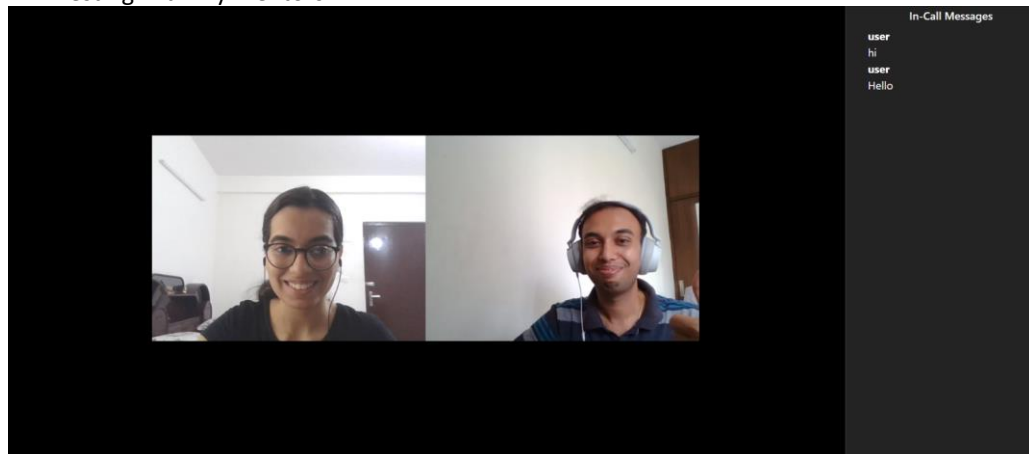
- Adding a Home Page and coding the exit button so that when clicked redirects to the homepage



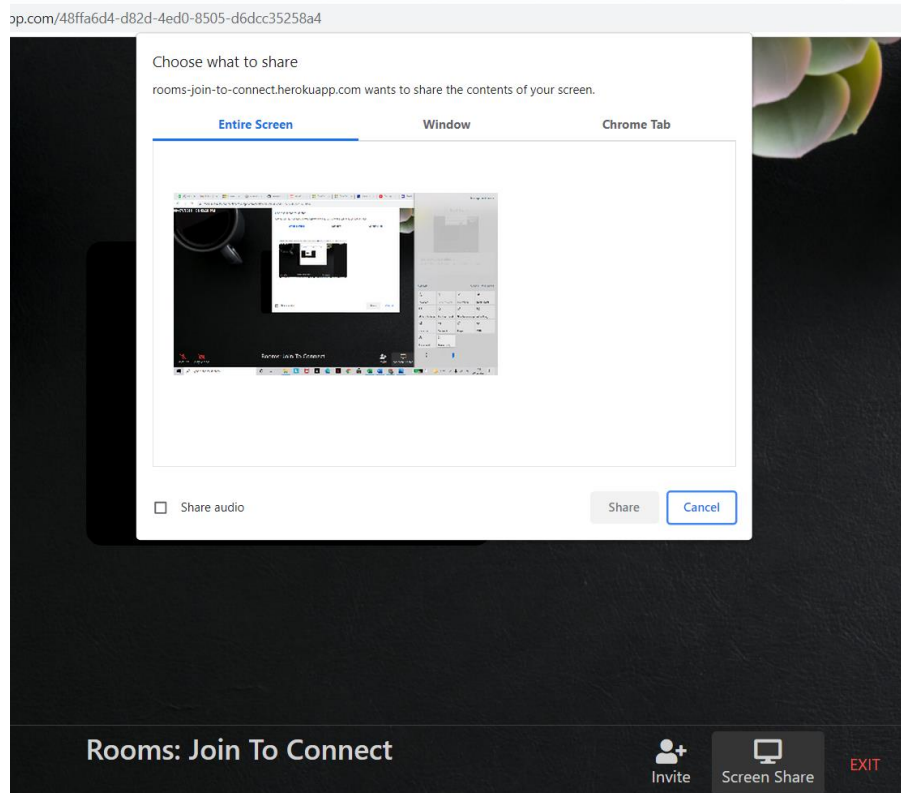
- Adding Chat Functionality and adding date and time



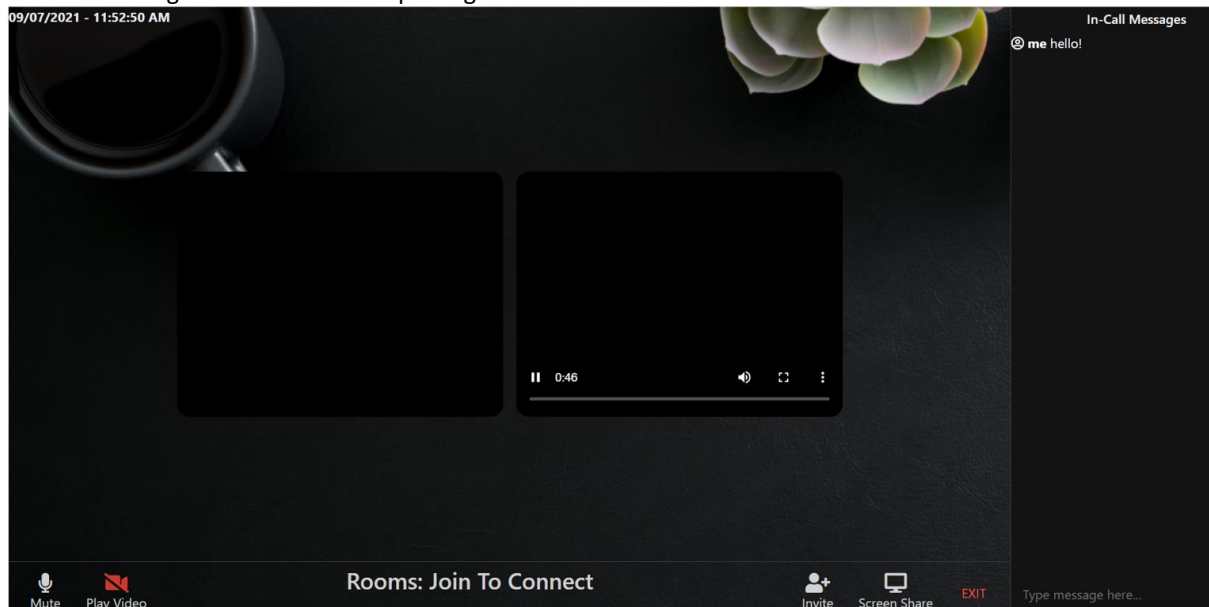
- Testing with my mentors



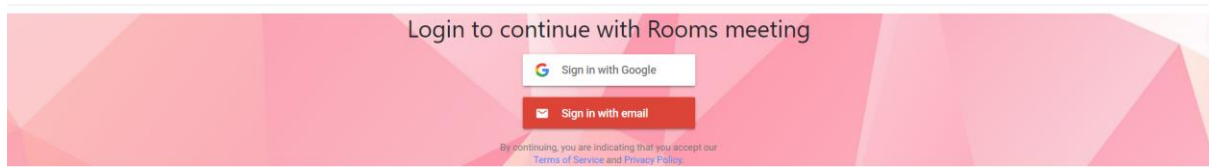
- Adding Screenshare



- Adding Video Controls and putting default audio video as off



- Adding User Authentication



[Go back to home](#)

REFERENCES/ SOURCES OF HELP:

- Online courses, documentations and tutorials for Node js, WebRTC, Sockets.io, Peer js, Firebase etc helped me gain the basic knowledge required for the implementation of the project.
- Various workshops and AMA sessions, 1:1 and group sessions with the mentors multiplied my learnings.
- Stack Overflow, solutions to most bugs and issues can be found on this website, all you need to do is a little research.
- Visual Studio Code, by the ease with which it links with git and github it proved really important while continuously committing and deploying.
- W3Schools, is a website to easily learn HTML, CSS and Javascript.