Microsoft Engage 2021

Mentored by : Swarnadeep Chatterjee & Abhilekh Malhotra

MICROSOFT CONNECT

Stay connected always.

Made by : Aishani Mitra
B. Tech Civil Engineering
Indian Institute of Technology, Guwahati

FEATURES IMPLEMENTED

- Peer to Peer video-calling feature
- User authentication using e-mail
- Normal in-call chatting feature
- Audio and video toggle option
- Screen-sharing feature
- Mobile responsive web-app

- BACKEND:

1. Server used : Express JS
2. Database used: MongoDB Atlas

- FRONTEND

1. HTML
2. CSS & SCSS

LIBRARIES USED

- Peer JS for WebRTC connection
- Socket IO for Socket connection
- UuidV4 for getting unique IDs
- Passport Authentication strategy for user authentication

SPRINTS

RESEARCH AND FINALISING ON OPEN SOURCE LIBRARY

I researched on Jitsi API, Twilio and WebRTC and finalized on using WebRTC since this would mean starting the project from scratch and not using existing pre-made features.

LOW LEVEL DESIGN

I first implemented a low-level design without any authentication. My LLD had the basic peer to peer calling feature, chat feature and the audio/video toggle feature.

HIGH LEVEL DESIGN

In my high-level design, I implemented more features like authentication using MongoDB as my database. I also made a few additional features like screen sharing and displaying the current time in the chat box when a message is sent.

WEEK 4

In week 4 I refined the UI, made my web-app mobile responsive and tried impkementing the chat feature. But that didn't come out too well!

SPRINT 1

IMPLEMENTATION

- Listed out pros and cons of using Twilio APIs, WebRTC or Jitsi API.
- Finalised on WebRTC.
- Made a rough plan of features to be implemented in my app.
- Looked up a little on Agile methodology of working.

FEEDBACK FROM SPRINT 1

- Need to learn more on NodeJS for using it in backend & MongoDB Atlas if that will be the final database chosen.
- Chalk out a simple UI plan for next sprint.
- Cleared out doubts related to working in sprints, its benefits and how it is used in the real world. Discussed on waterfall methodology too.

SPRINT 2

IMPLEMENTATION

- Made a basic version of my web app without any authentication.
- Used **NodeJS** for backend and **Express JS** as the server.
- Used **UUIDv4** to generate unique IDs for each meeting.
- Made a basic chat feature which doesn't display the name of the users nor displays time.
- Used **navigator.mediaDevices.getUserMedia** to get the users audio and video.
- Hosted a basic version on **Heroku** to test out.

FEEDBACK FROM SPRINT 2

- Try to implement authentication and use MongoDB Atlas as database.
- Currently the chat feature doesn't display name. Improve the chat feature, add current time and name of user too.
- Once the user disconnects the video of the person doesn't get removed. Fix that.
- App doesnt work on localhost. Try to find out the problem.

SPRINT 3

IMPLEMENTATION

- Implemented authentication. Used **Passport strategy** to check for already existing usernames and e-mails.
- Found the solution to the problem with the call feature working in local host. Solved this by adding a **setTimeout** in the **'user-connected' function** in socket code.
- Improved UI. Used **color palette** of **Microsoft Teams.**
- Added feature to remove person's video when he/she disconnects.
- Added chat feature with person's name and time displayed.

FEEDBACK FROM SPRINT 3

- The share link button gives the link but when I paste this link in the form, there's a 'Cannot GET' error. Fix this.
- Make the web-app mobile responsive.
- Add screen-share feature.
- The sign-up page redirects to the sign-up page again and the user has to go back to the login page. Change the routes a bit.

SPRINT 4

IMPLEMENTATION

- Fixed the link problem . The problem was that the share button was giving out the entire pathname. Silly mistake. Fixed this by **removing the trailing slash using slice.**
- Made app mobile responsive.
- Added screen-share feature using **navigator.mediaDevices.getDisplayMedia().**
- Made schema for conversations and tried storing messages in database.
- Changed routes so the sign-up page now **lands on to the login page incase of successful signup.**

FEEDBACK FROM SPRINT 4

- Couldn't store the real time changes in MongoDB. **Failed** to implement the **adapt feature**.

Thankyou!