A REPORT

ON

IMPLEMENTING MULTI SCREEN-SHARING FEATURE IN BIGBLUEBUTTON

BY

ANSH GUPTA 2020A7PS0116P

AT

Swecha, Gachibowli

A Practice School-I Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(July, 2022)

A REPORT

ON

IMPLEMENTING MULTI SCREEN-SHARING FEATURE IN BIGBLUEBUTTON

BY

ANSH GUPTA 2020A7PS0116P COMPUTER SCIENCE

Prepared in partial fulfilment of the
Practice School-I Course Nos.
BITS C221/BITS C231/BITS C241
AT

SWECHA, Gachibowli

A Practice School-I Station of

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

(July, 2022)

Acknowledgements

I am really grateful for this project opportunity and would sincerely thank our BITS Faculty coordinator Prof. Venkata Vamsi Krishna Venuganti for trusting me with this project. He has proved to be a guiding light in my entire journey.

Besides this, I would like to express my gratitude towards our project mentors from Swecha Mr. Shriharsha and Praveen for providing us with the necessary resources and guidance needed to fulfill this project. Their valuable insights have helped me improve and make this project a success.

I would also like to thank all my teammates for their support and dedication towards the project. The collaborative effort by everyone ensured the timelyprogress of the project.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI

(RAJASTHAN)

Practice School Division

Station: Swecha, Gachibowli **Centre:** Hyderabad

Duration: 8 weeks **Date of Start**: 30th May, 2022

Date of Submission: 21st July, 2022

Title of the Project: Implementing multi screen-sharing feature in BigBlueButton

Name of Student	ID	Discipline
Ansh Gupta	2020A7PS0116P	Computer Science

Name of Expert

Mr. Shriharsha Mopidevi

Mr. Praveen Gorla

Name of PS Faculty:

Prof. Venkata Vamsi Krishna Venuganti

Key Words: Open-source, BigBlueButton, multi screen-sharing

Project Areas: Web-development, Web-conferencing systems

Abstract: Web-conferencing systems are one of the most widely use web application even more after covid. BigBlueButton is an open-source web-conferencing system which is extensively used across various sectors. My task was to implement multi screen sharing feature in BigBlueButton. This report provides the wireframe, my progress and summary of my learning roadmap.

Table of Contents

Introduction

Swecha is an organization which works on various open-source projects and tweaks them for the benefit of the community. The project allotted to me in Swecha was, Development and addition of new features in BigBlueButton. BigBlueButton is an open-source web conferencing system for streamlined online learning. My team was assigned an issue from the BigBlueButton repository regarding the introduction of a multi-screen sharing feature in this application. So that multiple attendees could share their screen at the same time. Since BigBlueButton is one of the most widely used open source conferencing applications, having this feature could benefit a huge community.

Objectives

Proctoring

It makes proctoring much easier for the invigilators if students can share their screens at the same time. It also eliminates the need for proctoring software's, thus saving money for an organization and also eliminating privacy issues.

Online Learning

It would make online classes much more engaging for students. Imagine an online art class. Until now, the teacher has no way of identifying if the students are actually engaging in the class or not. But now, they'll have the option to monitor every student separately and appreciate their artwork. In a math class one student can learn from another student's way of solving a problem.

Productive Meetings

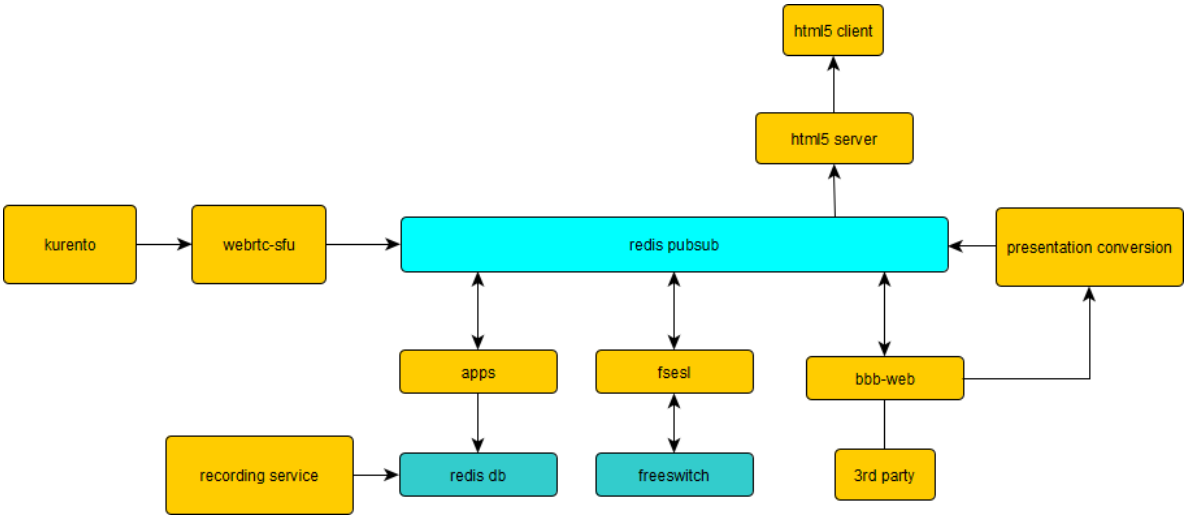
It would be very useful for meetings in general. If multiple users can share their screens at the same time, it makes a meeting more informative and productive. Also, attendees can visually represent their information. It would be a great tool for programmers to collaborate working on the same project. It would also be a nice streaming option for gamers imagining a chess tournament being streamed using the screen share option.

Work Checkpoints

Setting up Development Environment

I dual booted my Windows 11 system with Ubuntu 20.04 LTS for setting up the development environment in it. Following the official documentation I cloned the BigBlueButton repository on my system and installed all the necessary packages and frameworks required for it to work, resolving the errors that came along the way with the help of my teammates.

Decoding the Architecture



I thoroughly read and understood the application's architecture from the documentation. I figured out the necessary parts that we need to work on and summarized the tech stack that we will require to learn in order to implement the features that we thought of. Shared this knowledge with teammates and briefly explained to them the architecture.

As you can see from the diagram, HTML5 client is a web application that is accessible by users. The components that build the client are:

- ➔ ReactJS: For rendering the user-interface in an efficient manner
- ➔ WebRTC: For receiving/streaming audio and video

These connections are handled by web-server Nginx. The server is built upon:

- ➔ Meteor.js: For client and server communication
- ➔ MongoDB: keeps the state of the client consistent with the server.

BBB-web

BigBlueButton web application is a Java-based application written in Scala. It implements the BigBlueButton API and holds a copy of the meeting state.

Redis PubSub

Redis PubSub provides a communication channel between different applications running on the BigBlueButton server. When a meeting is recorded, all events are stored in Redis DB. When the meeting ends, the Recording Processor will take all the recorded events as well as the different raw (PDF, WAV, FLV) files for processing

FreeSWITCH

FreeSWITCH provides the voice conferencing capability in BigBlueButton. Users are able to join the voice conference through the headset. Users joining through Google Chrome or Mozilla Firefox are able to take advantage of higher quality audio by connecting using WebRTC.

Kurento Media Server (KMS)

Kurento Media Server is a media server that is responsible for the streaming of webcams, listen-only audio and screen sharing.

Learning Theory and Collaboration Tools

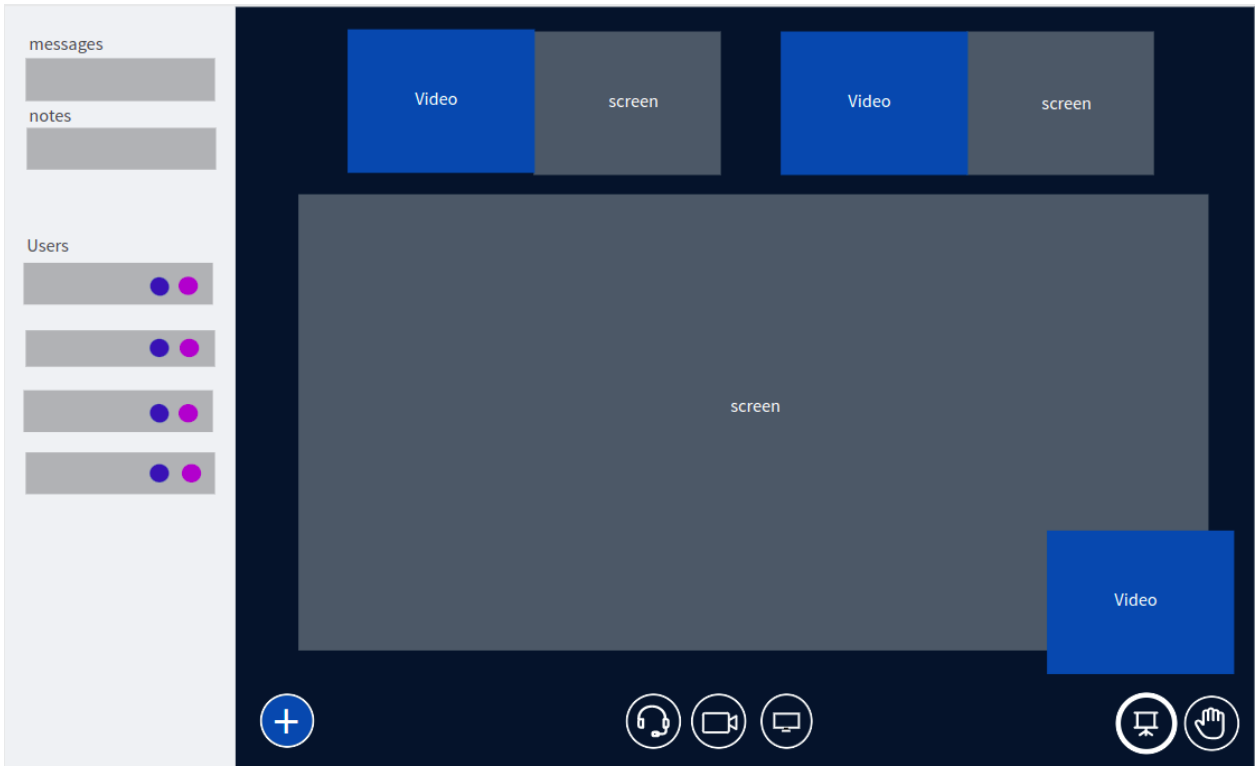
For the purpose of this project, I had to learn various software development concepts. I learned about the AGILE software development lifecycle. I practiced working with git and GitLab which is a version control system necessary for collaboration and tracking project history. Apart from this I also learned theory and various concepts such as Computer Networking and OSI model , WebRTC and Sockets , Docker and Containerization , Redis and databases , Nginx web server etc. through online tutorials and documentation. I also learned PenPot which is an open source designing and prototyping tool. I also learned the basics of UI/UX designing required for designing the prototype.

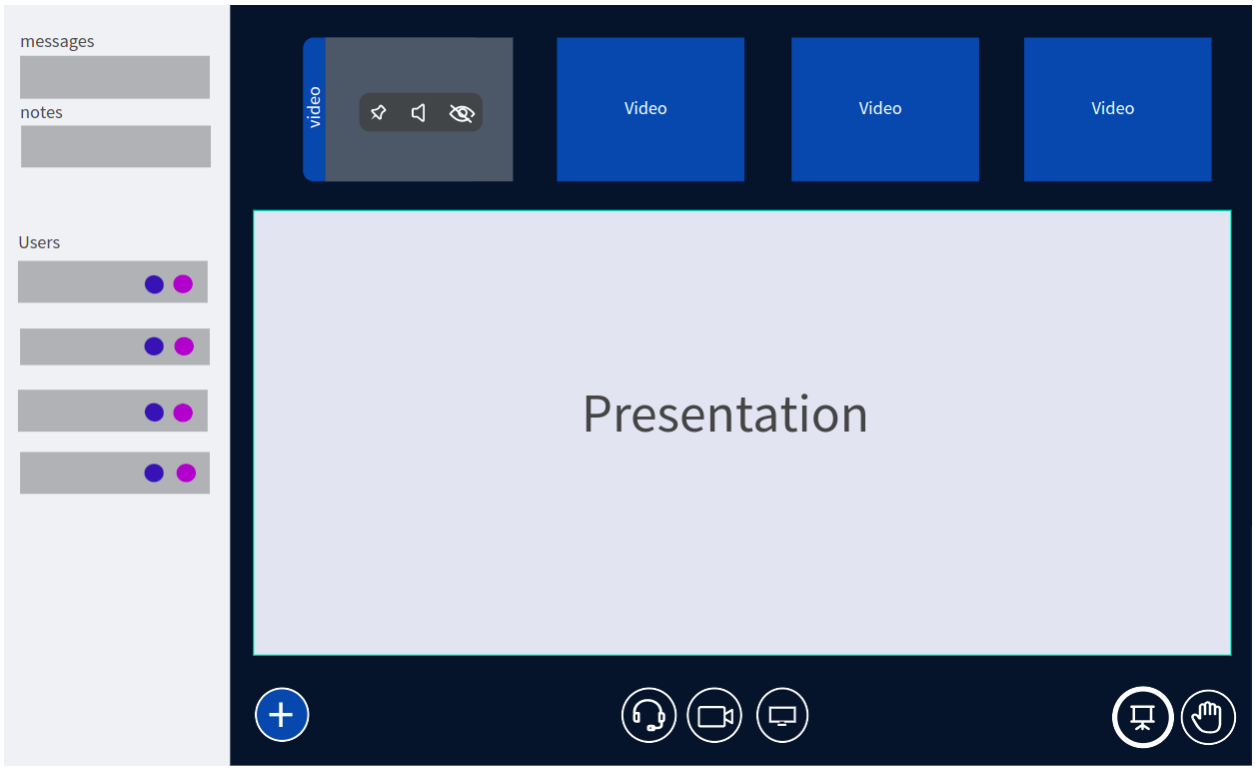
Designing and Prototyping

Using PenPot, an open source designing and prototyping tool, I created the wireframe and prototype of all the use cases of the feature required. A flowchart of user interaction is made. The features planned to be accommodated in the beta release are formulated and planned of. Below are a few snapshots from the wireframe I made.

Fig 1. Upon Pinning the shared screen, it becomes the main screen for a user.

Fig 2. Upon hiding the video of a user and just seeing the screen.


messages
notes
Users
Video
screen
Video
screen
screen
Video


messages
notes
Users
video
Video
Video
Video
Presentation

Learning the Tech Stack

I summarized the tech stack used in the application following the architecture for the team for an easy learning roadmap.

Tech Stack used:

1. HTML5, CSS and Javascript
2. ReactJS as a frontend framework
3. Akka apps using JAVA and Scala
4. Meteor.js in ECMA2015 for communication between client and server.
5. MongoDb and Redis databases for state management and recording
6. FreeSwitch application for handling audio
7. WebRTC for sending/receiving audio and video files
8. BigBlueButton comes with its own front-end called Greenlight

Source: https://docs.bigbluebutton.org/2.4/architecture.html

Roadmap I created for learning the relevant part for our project:

1. Git and Gitlab for collaboration and version control
2. PenPot for designing and prototyping.
3. Javascript HTML5 and CSS for basic Front-End development
4. React js for Front-End development.
5. Nodejs Mongodb and express for backend development.
6. Meteor js to interact with frontend and backend.
7. Kurento Media Server (KMS) and webRTC stfu - responsible for streaming of webcams, listen-only audio, and screen-sharing.

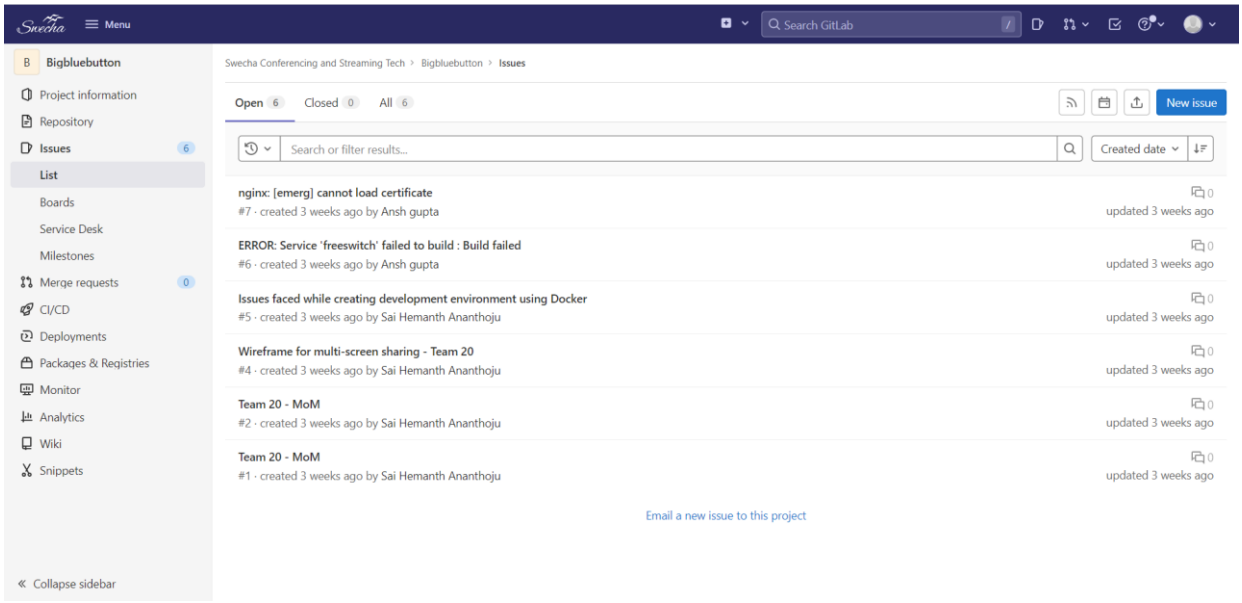
BBB Server Installation

I learned and installed Docker for Linux. Docker is an open-source platform for developing, shipping and running applications in the form of docker containers. We used the BigBlueButton's official docker image to setup the server client on our system. For doing this I learned the basics of docker and containerization along with all the necessary commands. We were able to run the BBB API on our local environment and performed API testing using AI mate.

Raised issues on GitLab

During the server installation and setting up of development environment. Most of them the team members solved by collaborating with each on the in standup meets. Persistent bugs and updates were posted on GitLab as issues so that the mentors could tract our progress and help us with persistent bugs.

Fig 1. Screenshot of issues raised on GitLab repository of Swecha.



Learning Outcomes

Theory

Learned Backend and other relevant concepts such as

- Computer Networking and OSI model
- WebRTC and Sockets
- Docker and Containerization

- Redis , Nginx
- Git and GitLab etc

Prototyping

I made the design and the workflow of the app features. Learned PenPot, an open source designing tool for making wireframes and flowchart for the use cases and user stories.

Small demo: https://design.penpot.app/#/workspace/65098030-f2fa-11ec-b774-f940e3befd53/6eb145a0-f2fa-11ec-b774-f940e3befd53?page-id=31964660-f3cd-11ec-bc51-533cd82cdb21

Front-End

Learned various technologies necessary to work on BigBlueButton. One such library was Reactjs. Since React is a very extensive library and takes months to master generally. The only way to learn it for me was to practice it by making mini projects along the way using React JS.

Small Demo: https://ansh-25.github.io/Quote-generator/

Key Learnings from This project:

1. React Routing
2. API data fetching using Axios
3. Tailwind CSS
4. React Hooks such as useState and useEffect.
5. React functional components and JSX
6. Application of VCS in a real project
7. Deployment and hosting using Gh-Pages

Some responsive websites I made along the way to practice HTML , CSS and Vanilla JS:

1. https://ansh-25.github.io/Frontend_projects/Krunch/
2. https://ansh-25.github.io/Frontend_projects/Edie/
3. https://ansh-25.github.io/Frontend_projects/Interior/
4. https://ansh-25.github.io/Frontend_projects/Slick/

Conclusion

The project was a hand-on project. Learned and made several web development projects along the way. Followed AGILE SDLC and experienced software development in a corporate setup. I learned various frame works and libraries in this project. Got to know about open-source community and contributed to one. The internship experience at Swecha was very fruitful and enlightening.

References

1. https://swecha.org/
2. https://reactjs.org/
3. https://www.w3schools.com/
4. https://www.kurento.org/
5. https://bigbluebutton.org/
6. https://code.swecha.org/
7. https://docs.bigbluebutton.org/2.4/architecture.html
8. https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API

Glossary

BigBlueButton: An open-source web conferencing system.

Open-source software: Open-source software is computer software that is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software and its source code to anyone and for any purpose.

Web Development: Web development is the work involved in developing a website for the Internet or an intranet.

React JS: React is a free and open-source front-end JavaScript library for building user interfaces based on UI components.