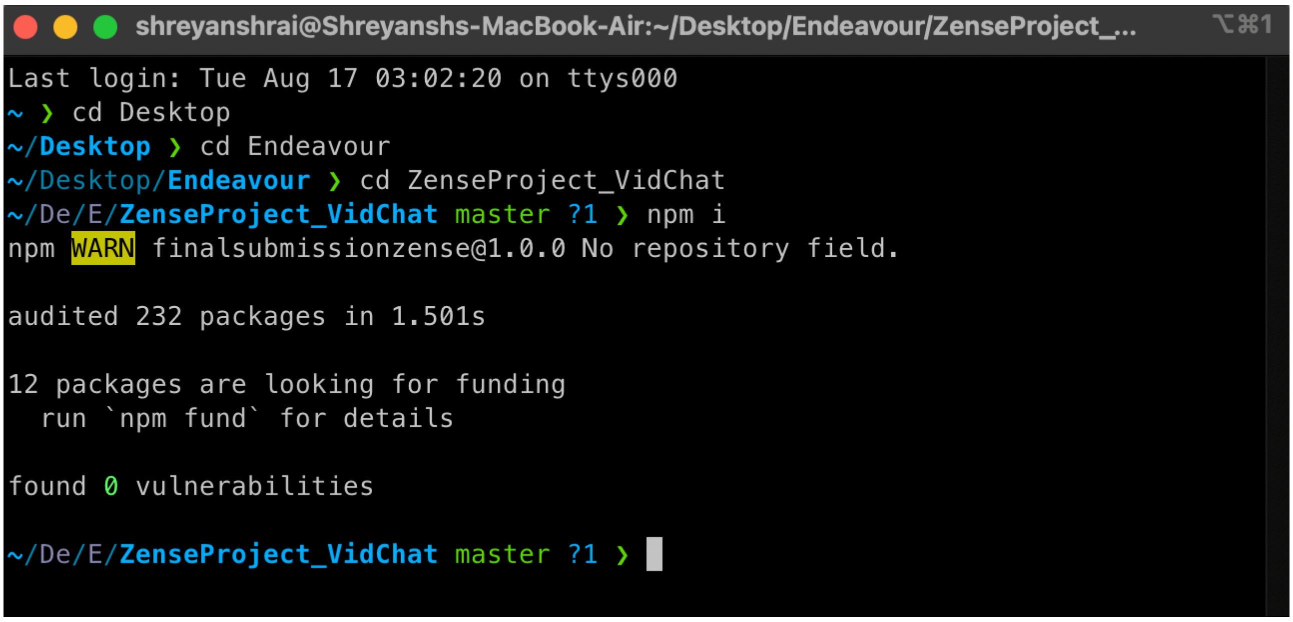
Zense Recruitment Project

Project Name: Glimpse
Created By : Shreyansh Rai IMT2020501

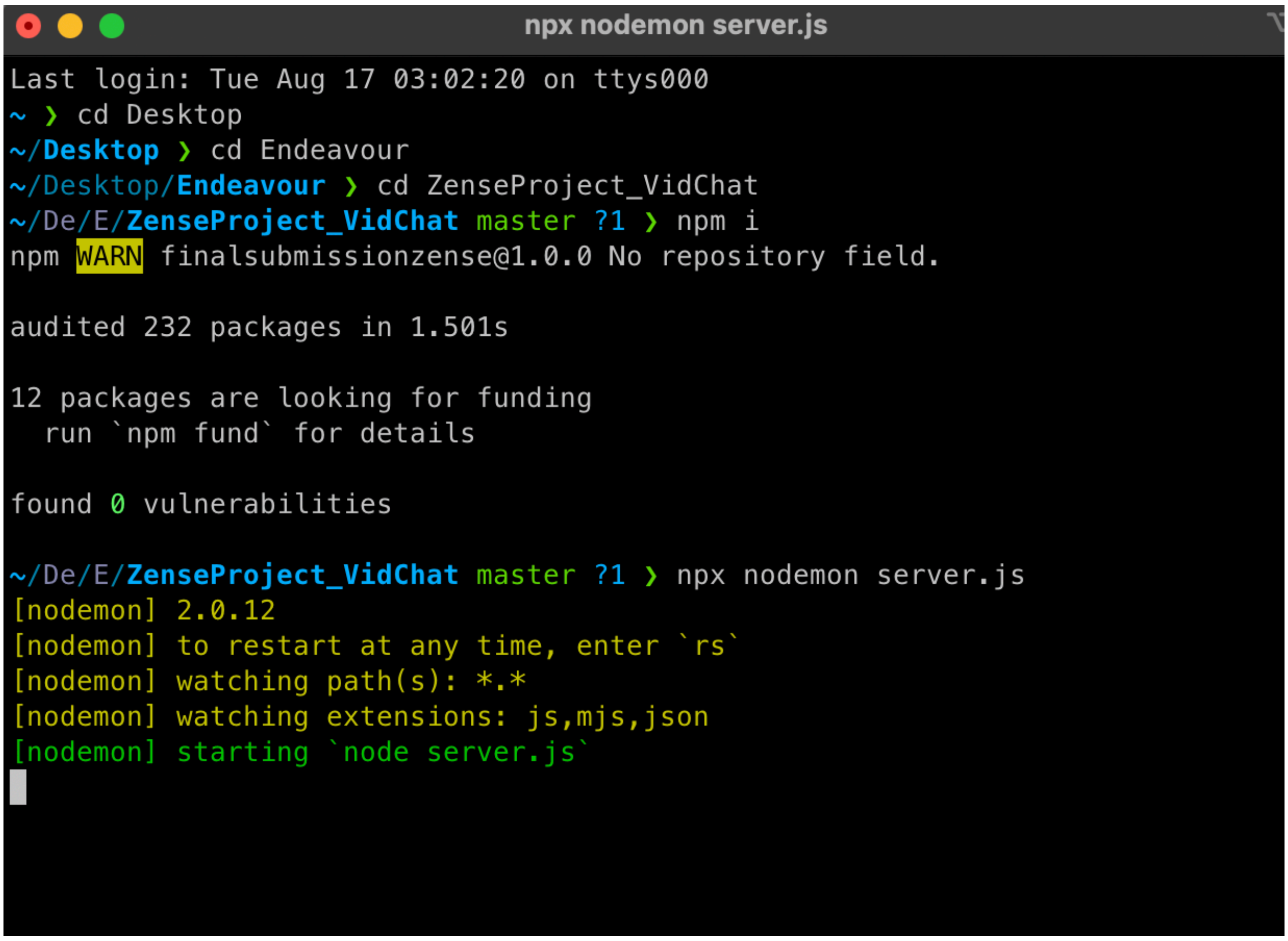
How to run in order to test it?

- Go to the repository https://github.com/Shreyansh-Rai/ZenseProject_VidChat
- Clone the project locally and go to the directory where the project is cloned
- Now run the command : npm i



```
shreyanshrai@Shreyanshs-MacBook-Air:~/Desktop/Endeavour/ZenseProject_...    ⌥⌘1
Last login: Tue Aug 17 03:02:20 on ttys000
~ ❯ cd Desktop
~/Desktop ❯ cd Endeavour
~/Desktop/Endeavour ❯ cd ZenseProject_VidChat
~/De/E/ZenseProject_VidChat master ?1 ❯ npm i
npm WARN finalsubmissionzense@1.0.0 No repository field.

audited 232 packages in 1.501s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

~/De/E/ZenseProject_VidChat master ?1 ❯
```

- Now run the command: npx nodemon server.js to start the backend locally
- the program runs on the port number 5000 so in your browser after the server has started running, visit localhost:5000 and the application will automatically assign a room to the user joining and the user will be redirected to it.
- To allow others to join the same room the link (in the address bar of the browser) has to be shared to them and they can also join the meeting using the link.



```
Last login: Tue Aug 17 03:02:20 on ttys000
~ ❯ cd Desktop
~/Desktop ❯ cd Endeavour
~/Desktop/Endeavour ❯ cd ZenseProject_VidChat
~/De/E/ZenseProject_VidChat master ?1 ❯ npm i
npm WARN finalsubmissionzense@1.0.0 No repository field.

audited 232 packages in 1.501s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

~/De/E/ZenseProject_VidChat master ?1 ❯ npx nodemon server.js
[nodemon] 2.0.12
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
```

Here is a demo link of one of the builds of this project that I have hosted on heroku and this can be joined directly by anyone globally.

I advise testing the application locally before going to heroku the reason will be explained at a later point in this report where I explain the challenges faced during the project's development.

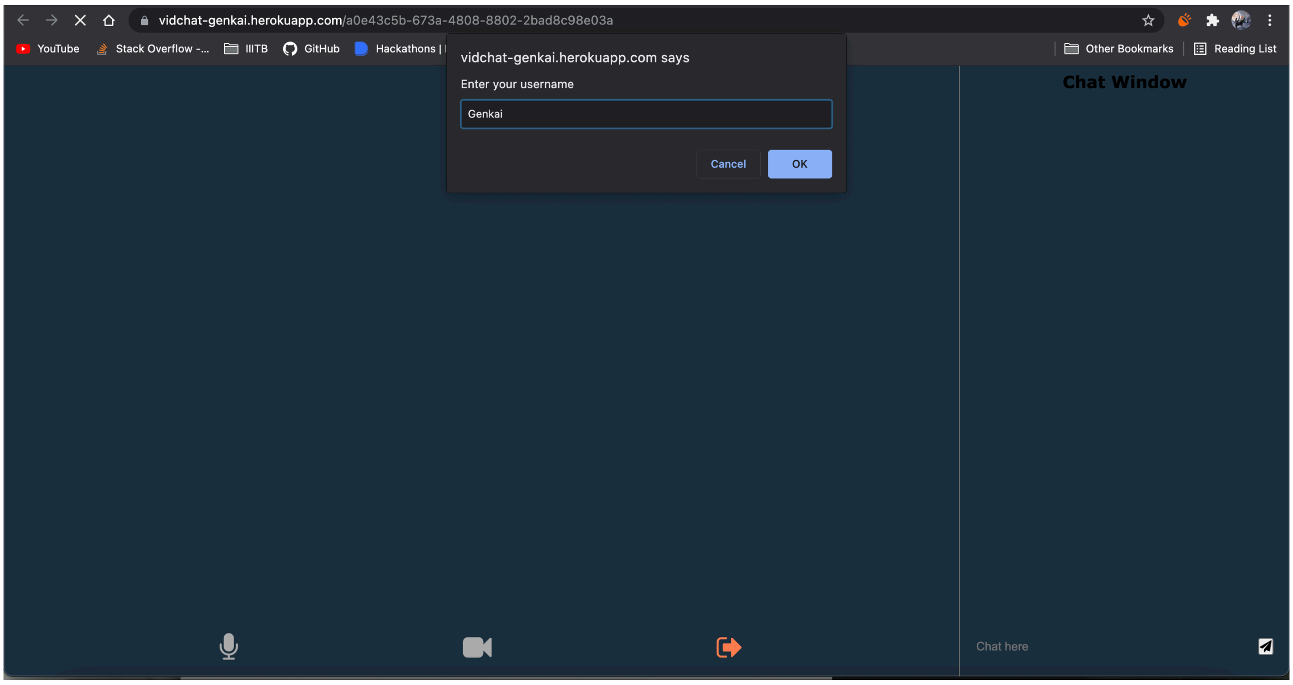
https://vidchat-genkai.herokuapp.com/

Joining the above link will then redirect you to a room and sharing that link with others will allow them to join you.

Project Description :

When chatting on most major video chatting solutions one must install an application and open it each time they wish to attend such a meeting but often times this is a slow and tedious process. Glimpse is a way to quickly share the link of a meeting to your friends in order to have a quick chat. There is a system for handling multiple rooms. The Web app has complete integrated chat functionality as well as the ability to mute and turn off camera. It is supported on MacOS, Windows, Linux and Android and works on most browsers without a hitch. The app can support multiple people at once and is peer to peer.

When a user joins the meeting via the link, they are prompted to enter the name they want to use for the duration of the meeting.

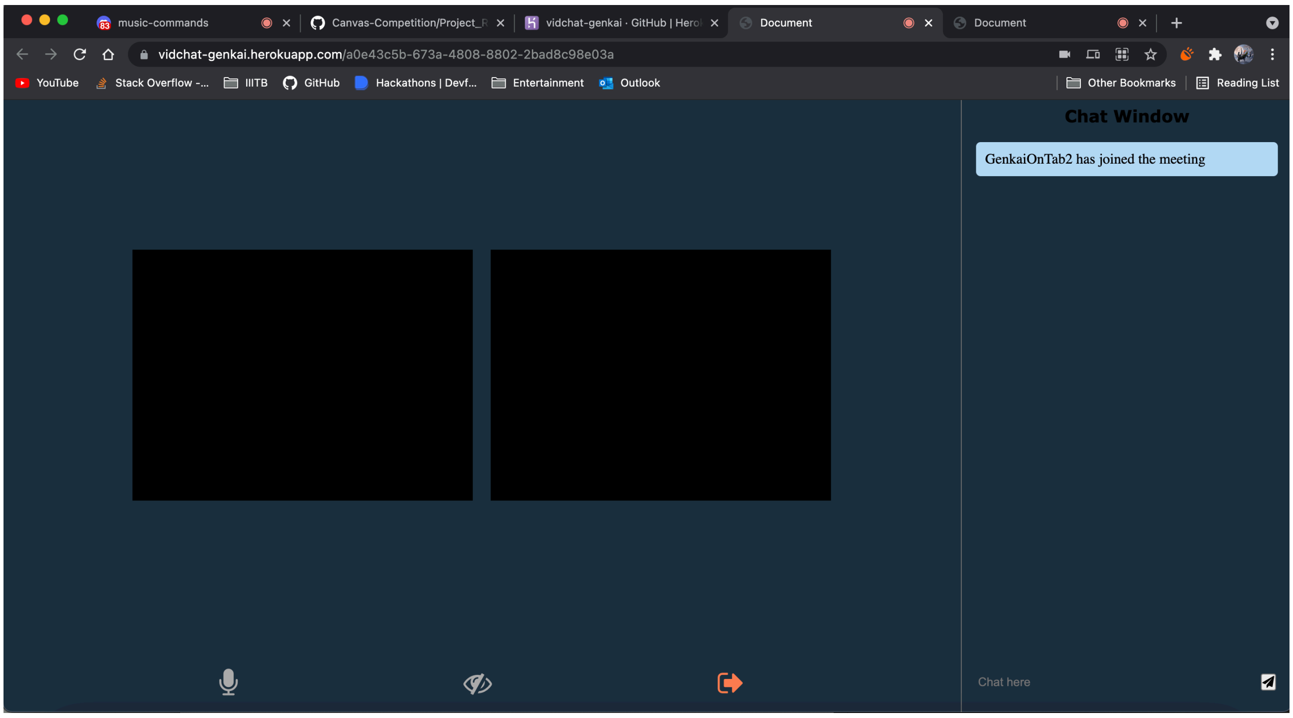


vidchat-genkai.herokuapp.com says

Enter your username

Genkai

All the users in the meeting are alerted to the new user joining

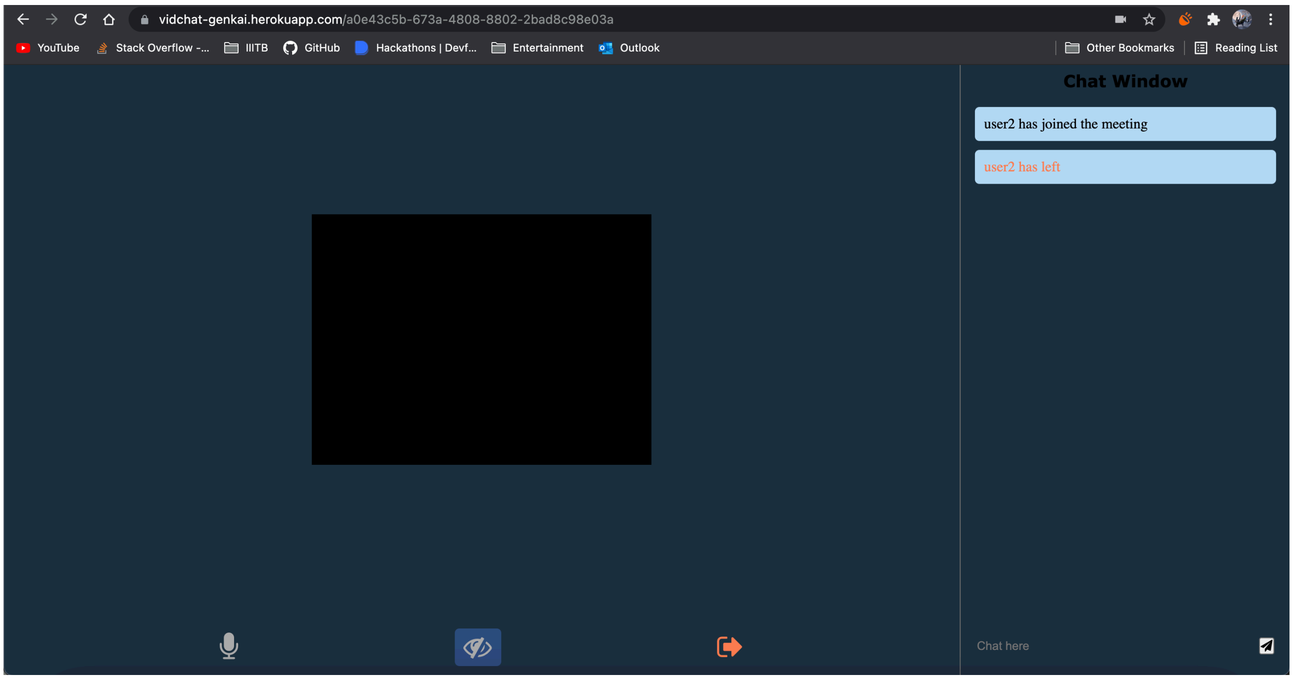


GenkaiOnTab2 has joined the meeting

(As You can see on the bottom there are controls to mute yourself and turn off your camera I have my camera turned off as indicated by the slashed eye icon)

The users can interact in the chat using the chat area provided to the right side of the screen

They can leave the meeting by clicking the leave icon in red and the room is alerted that the user has left the meeting.



user2 has joined the meeting

user2 has left

My Journey and details about implementation

So the basic idea was to make a video chatting application but at this point I should make it clear that it was something I had no idea about. Web development was something that had my curiosity but also a blind area for me. after several gruelling hours of learning the technologies necessary for this project, I started small trying to implement the concepts I learnt in basic projects.

After I learnt HTML and CSS I tried to create a basic website :

https://github.com/Shreyansh-Rai/PersonalWebsite https://shreyansh-rai.github.io/PersonalWebsite/

Above is the link to the place where it is hosted. I worked around with how animations and media controls work and optimised the site a little for usage on a mobile phone.

After this I learnt JavaScript and socket.io to make a basic chat application that I knew was the first step in actually making something useful you can check it out over here:

https://github.com/Shreyansh-Rai/ChatApp

here I have tried to implement a very basic chat application but there were several problems. For one it ran on the entire port and had no concept of a room.

Then after much research into the modules and frameworks that might be useful in making the video chatting application that I wanted to make, I decided upon Peer Js for my connections between the peers and the back end has node.js for the server and I am running it in addition to using Express for the requests and responses to the server (because http was a little cumbersome and Express was at least one other technology that I thought would perhaps be useful later on so why not!)

uuid was used to generate random rooms

What did I learn?

First off I never knew anything about Web Development and the process of starting was really intimidating. Every time I learnt a new thing I just realised that there was some other thing that I needed to learn before starting and I was itching to actually do something because I did not want to get stuck in tutorial hell and end up doing nothing.

I learned HTML, CSS, Javascript, Node JS, Express, Peer JS and more importantly how to use it all together.

But Most importantly I realised that most of this was just a matter of how the individual modules were implemented and there is only one way to eventually utilise them in an application, that is to read there documentation!

Although I might surely eventually forget some of what I learned in the last (nearly) 20 days I am confident that I will be able to get back to speed after looking at the code and reading the documentation of the various components of this program.

Also I got better at debugging effectively.
What I mean by this is that there was a bug in the code such that when a person clicked on the leave button in the meeting their peer was getting destroyed as required but the last frame of the peer that left could be seen by the other people so that was an issue and so a senior advised me to remove the divs entirely so the next thing I did was as I added the video elements to the screen I added ids corresponding to the peers that sent that stream (which took a lot of digging around the documentation to find out how to access the peer id on call) and then I generated a userleft event from the server and broadcasted it to the room and removed the video of the peer id that was destroyed.

<u>Challenges with Heroku</u>

While the process for leaving the meeting is handled carefully as mentioned above there is another event that was proving to be difficult to handle, that is a user getting disconnected, in such an event the socket emits a disconnect event and although removing the disconnected users in a similar way to when the users leave by hitting the leave button, the problem is, that on local host for some reason this works perfectly but on the hosted version of the app, it works with varying rates of success. So while it does mostly work sometimes there is an error. Due to this The code pushed onto heroku does not have the implementation of a user getting suddenly disconnected, but the code on the github repo does.