

README

INSTRUCTIONS TO RUN :

1) clone the repository using

```
git clone https://github.com/Shreyansh-Rai/interviewme.git
```

2) change directory into interviewme/commsModule

```
run: npm install
```

```
run: npx nodemon server.js
```

3) change directory into .interviewme/interview

```
run: npm install
```

```
change directory into interviewme/interview/backend
```

```
run: npx nodemon server.js
```

4) change directory to interviewme/interview/src

```
run: npm start
```

5) In case of errors shown in step 4 (ie. a browser with the webpage does not open)

```
run: export NODE_OPTIONS=--openssl-legacy-provider
```

followed by

```
run: npm start
```

List of recommended browsers :

Although the npm start command will open a window in the browser that is set default on your machines the following is a list of browsers that we recommend you to open the localhost at :

1) Brave Browser

2) Mozilla Firefox

3) Safari

Although chrome works as well sometimes there is a cors issue while communicating between inter port processes as we have and so we do not advise using it to test the application.

LIST OF FEATURES:

AIM : The purpose of this application in short is to provide an interview experience such that minimal information about the interviewee is sent and shown to the interviewer unless the interviewee is accepted for the job post.

Now to implement this functionality we have made an application using mern stack that communicates with a server on the backend and also fetches and displays appropriate information to the interviewer.

We have implemented the functionality to list down meetings based on our database and also to join the meeting and get information about the user.

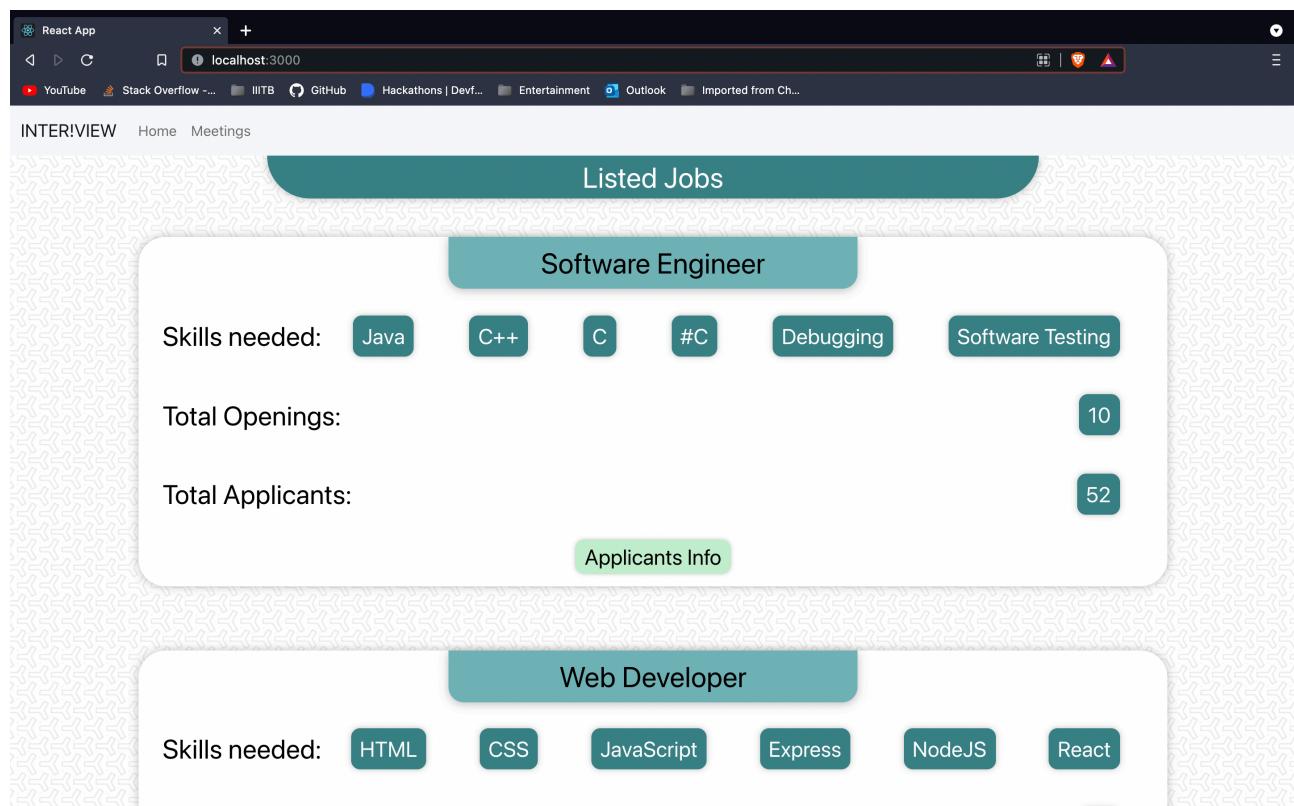
Before the user is accepted for the job no information about his name age or sex is revealed to the interviewer.

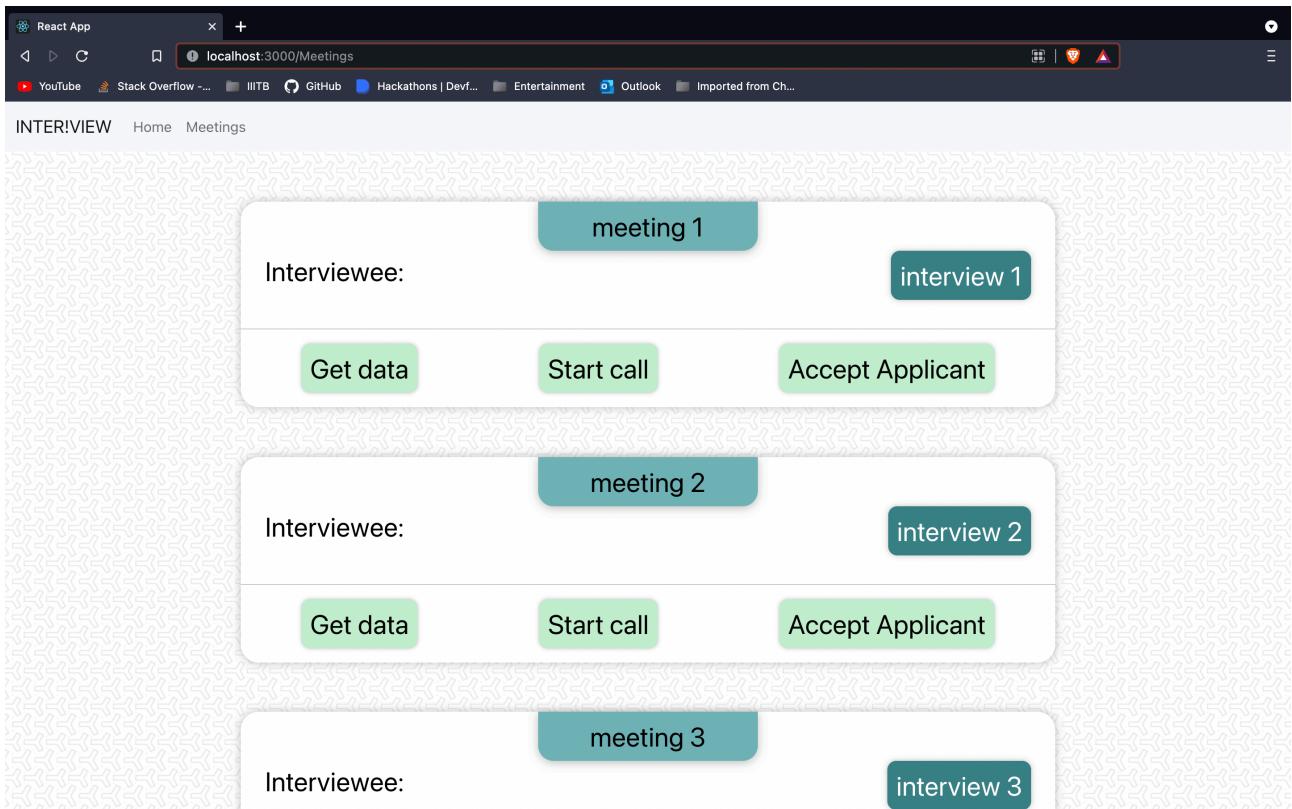
We have made a fully functional communication system for the interview and the interviewee and interviewer can communicate with each other via audio call or chat. Video is deliberately not enabled for them as it is proven to generate a bias in the interview and a completely anonymous interview is facilitated by the built in meeting system in our application.

To ensure no cheating goes on the side of the interviewee we have implemented a Machine Learning based proctoring system in order to detect if there are more than one people present in the interviewee's frame and an alert is issued to both the interviewee and interviewer about the same.

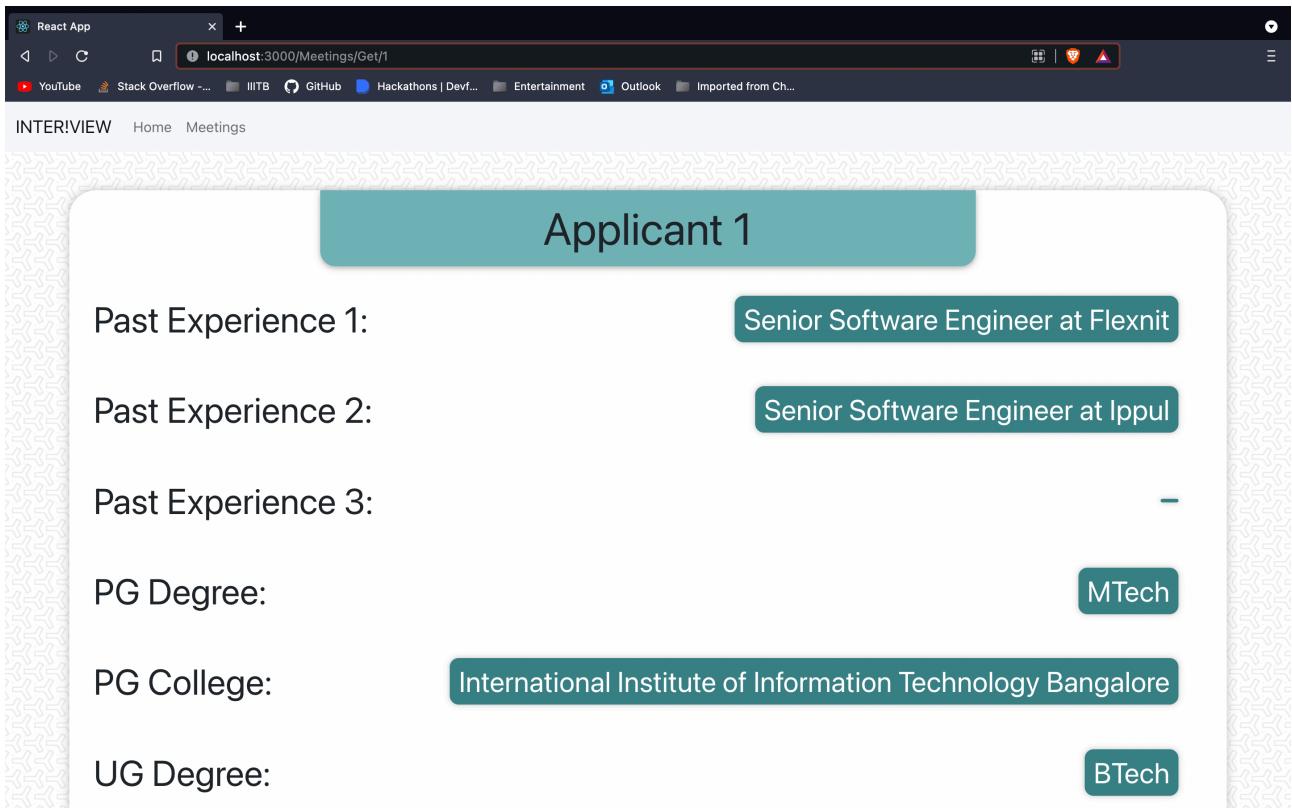
A lot more needs to be said about the project and a ppt will be soon uploaded detailing the same.

SCREENSHOTS :





Above is the meetings page. On clicking Get Data



(notice how there is no other information that reveals the person's gender or name or age)

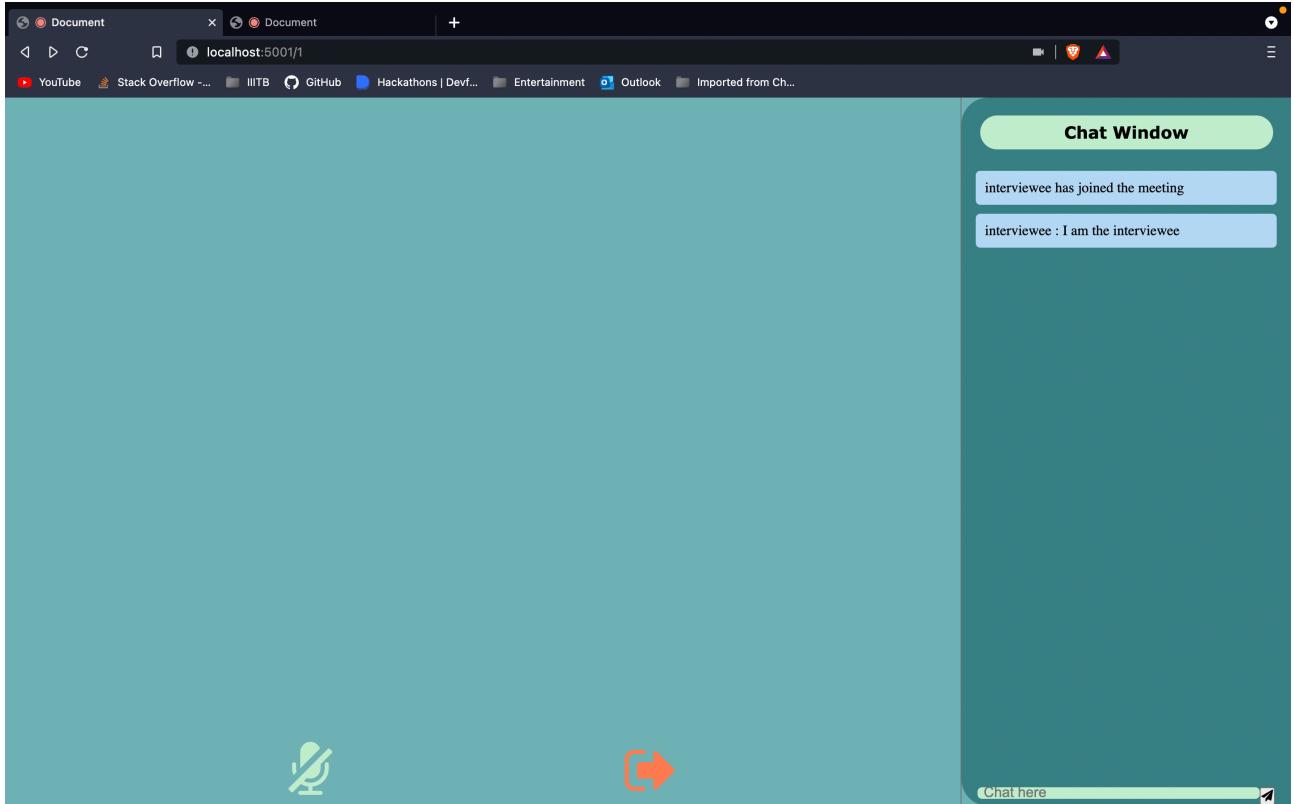
A screenshot of a web browser window titled "React App". The address bar shows "localhost:3000/Meetings/Get/1". The page content displays "Past Experience 2" at the top. Below it, there is a list of educational details:

- PG Degree: MTech
- PG College: International Institute of Information Technology Bangalore
- UG Degree: BTech
- UG College: International Institute of Information Technology Bangalore
- Higher secondary school: Essar International School

Now we look at the meeting portal :

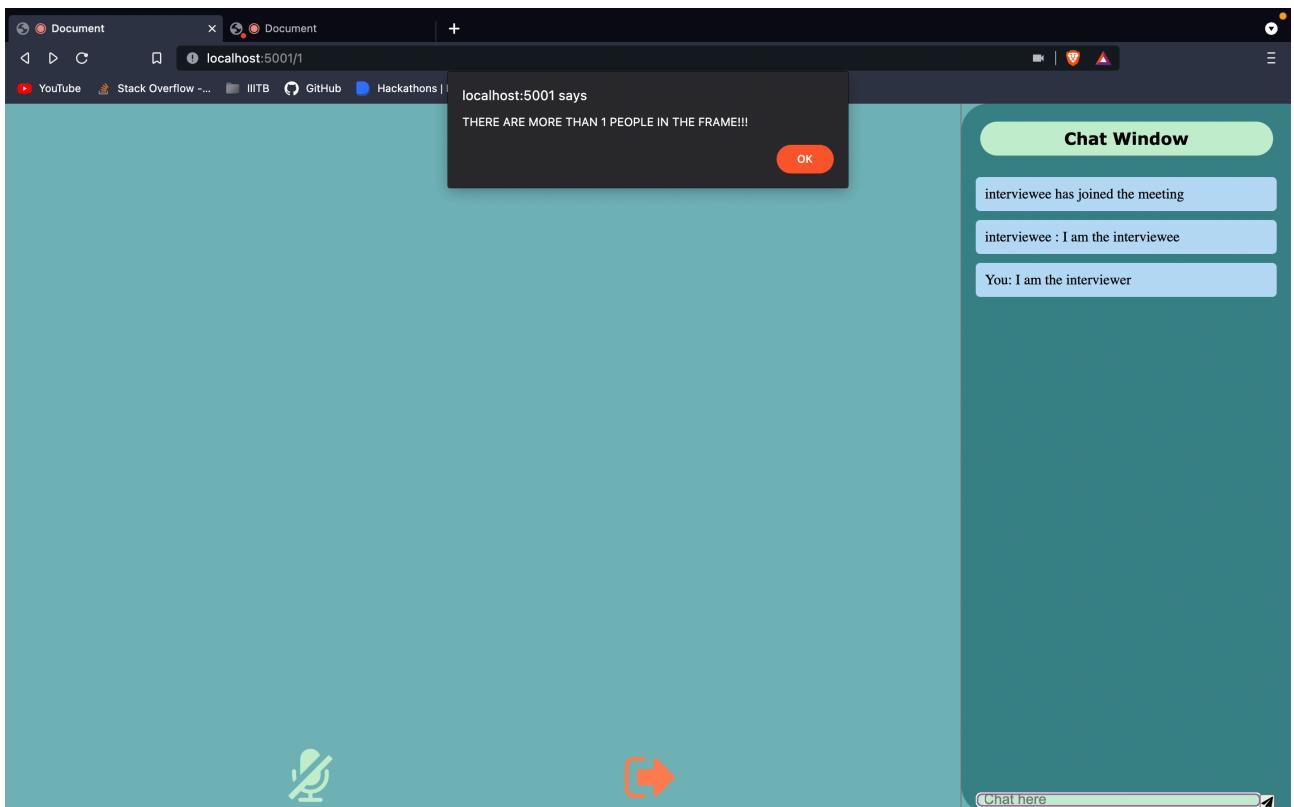
A screenshot of a web browser window titled "Document". The address bar shows "localhost:5001/". The page content features a video call interface with a "Chat Window" on the right side. The chat window has a green header labeled "Chat Window" and a message "You: I am the interviewee". At the bottom of the screen, there are icons for microphone and camera, along with a red "Call" button.

Here the chatting functionality works along with audio chat.



An example of messages sent to and from the interviewer

Below is our proctoring system that informs the interviewer when more than 2 people are present in the frame.



Accept Applicant is a representative button that shows us what happens when the interviewee is granted a job and his CV is shown to the interviewer. At this point the portal's job is done and we make a small change to the DB.

The screenshot shows a web application interface titled "React App" with a tab labeled "Document". The URL in the address bar is "localhost:3000/Meetings/Accept/1". Below the header, there is a large teal button with the text "Applicant 1". The main content area displays the following information:

- Name : Sarthak Harne
- Gender : Male
- Age : 28
- Past Experience 1: Senior Software Engineer at Flexnit
- Past Experience 2: Senior Software Engineer at Ippul
- Past Experience 3: -

A horizontal ellipsis (...)