

INTERNSHIP REPORT

MEETUP

TABLE OF CONTENT

1	INTRODUCTION
2	ABSTRACT
3	PROBLEM STATEMENT
4	SYSTEM REQUIREMENTS
5	PROPOSED METHOD WITH ARCHITECTURE
6	IMPLEMENTATION
7	CONCLUSION
8	TEAM MEMBER DETAILS

INTRODUCTION

Meetup is a video chatting application that allows a user to video chat around the world with people having common interests, to see new meetups on time, and gets recommended meetups. This is a website that allows anyone to start a "meetup" in their area. A meetup, simply put, is a group of people that come together to socialize around a

common interest. The idea behind Meetup is to "use the internet to get in touch with people - and grow local communities.

ABSTRACT

This project deals with the '**Meetup web application**'. It helps you to create and join local social groups through video chat. It is a service used to organise online groups that host virtual events for people with similar interests.

PROBLEM STATEMENT

MEETUP-An App based on the common interests

Let's assume if you like Football or Cricket so much and you can find some unknown having the same interests.

You can send Hello to him/her and can chat as well. Fantastic. Let's Jump into the Project Ideas

. You can Select the Interest from the menu

. Find the person with the common interests and send him Hello request . You can get so much information about your interests. Chat with Friends

SYSTEM REQUIREMENTS

Hardware Requirements	<ul style="list-style-type: none">• System : windows home• Hard Disk : 40 GB• RAM : 512 Mb
Software Requirements	<ul style="list-style-type: none">• Operating system : Windows 9 or Windows 10Coding language : HTML,CSS,JSS• Documentation : VS code

PROPOSED METHOD WITH ARCHITECTURE

HTML CODE

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset='utf-8'>
5   <meta http-equiv='X-UA-Compatible' content='IE=edge'>
6   <title>Meetup</title>
7   <meta name='viewport' content='width=device-width, initial-scale=1'>
8   <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
9 </head>
10 <body>
11
12   <button id="join-btn">Join </button>
13
14   <div id="stream-wrapper">
15     <div id="video-streams"></div>
16
17     <div id="stream-controls">
18       <button id="leave-btn">Leave Stream</button>
19       <button id="mic-btn">Mic On</button>
20       <button id="camera-btn">Camera on</button>
21     </div>
22   </div>
23
24 </body>
25 <!-- <script src="https://download.agora.io/sdk/release/AgoraRTC_N.js"></script> -->
26 <script src="AgoraRTC_N-4.8.0.js"></script>
27 <script src="main.js"></script>
28 </html>

```

CSS CODE

```

1 body{
2   background: #3735cc;
3   background: -webkit-linear-gradient(to right, #2dc2ae, #0e7a9e, #ff9100);
4   background: linear-gradient(to right, #58afd4, #3692b1, #2dc0ff);
5 }
6
7 #join-btn{
8   position: fixed;
9   top: 50%;
10  left: 50%;
11  margin-top: -50px;
12  margin-left: -100px;
13  font-size: 18px;
14  padding: 20px 40px;
15 }
16
17 #video-streams{
18   display: grid;
19   grid-template-columns: repeat(auto-fit, minmax(500px, 1fr));
20   height: 90vh;
21   width: 1400px;
22   margin: 0 auto;
23 }
24
25 .video-container{
26   max-height: 100%;
27   border: 2px solid black;
28   background-color: #203449;
29 }

```

```

30
31 .video-player{
32   height: 100%;
33   width: 100%;
34 }
35
36 button{
37   border: none;
38   background-color: rgb(41, 26, 129);
39   color: #fff;
40   padding: 10px 20px;
41   font-size: 16px;
42   margin: 2px;
43   cursor: pointer;
44 }
45
46 #stream-controls{
47   display: none;
48   justify-content: center;
49   margin-top: 0.5em;
50 }
51
52 @media screen and (max-width: 1400px){
53   #video-streams{
54     grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
55     width: 95%;
56   }
57 }

```

JSS CODE

```

const APP_ID = "698b2373d582428893fe5fda6c90ea2d"
const TOKEN = "00669b2373d582428893fe5fda6c90ea2dEADruvFTNr3gZDgm258dHfH+ADuJVWg4fJgTfEhAasvR57awAAAAAEACu7KxLvsXVYQEAQC/3dVh"
const CHANNEL = "Meeting"

const client = AgoraRTC.createClient({mode: 'rtc', codec: 'vp8'})

let localTracks = []
let remoteUsers = {}

let joinAndDisplayLocalStream = async () => {
  client.on('user-published', handleUserJoined)
  client.on('user-left', handleUserLeft)

  let UID = await client.join(APP_ID, CHANNEL, TOKEN, null)
  localTracks = await AgoraRTC.createMicrophoneAndCameraTracks()

  let player = `<div class="video-container" id="user-container-${UID}">
    <div class="video-player" id="user-${UID}"></div>
  </div>`
  document.getElementById('video-streams').insertAdjacentHTML('beforeend', player)

  localTracks[1].play('user-${UID}')

  await client.publish([localTracks[0], localTracks[1]])
}

let joinStream = async () => {
  await joinAndDisplayLocalStream()
  document.getElementById('join-btn').style.display = 'none'
  document.getElementById('stream-controls').style.display = 'flex'
}

```

```

let handleUserJoined = async (user, mediaType) => {
  remoteUsers[user.uid] = user
  await client.subscribe(user, mediaType)

  if (mediaType === 'video'){
    let player = document.getElementById(`user-container-${user.uid}`)
    if (player != null){
      player.remove()
    }

    player = `<div class="video-container" id="user-container-${user.uid}">
      <div class="video-player" id="user-${user.uid}"></div>
    </div>`
    document.getElementById('video-streams').insertAdjacentHTML('beforeend', player)

    user.videoTrack.play('user-${user.uid}')
  }

  if (mediaType === 'audio'){
    user.audioTrack.play()
  }
}

let handleUserLeft = async (user) => {
  delete remoteUsers[user.uid]
  document.getElementById(`user-container-${user.uid}`).remove()
}

let leaveAndRemoveLocalStream = async () => {
  for(let i = 0; localTracks.length > i; i++){
    localTracks[i].stop()
    localTracks[i].close()
  }
}

```

```

69     await client.leave()
70     document.getElementById('join-btn').style.display = 'block'
71     document.getElementById('stream-controls').style.display = 'none'
72     document.getElementById('video-streams').innerHTML = ''
73 }
74
75 let toggleMic = async (e) => {
76     if (localTracks[0].muted){
77         await localTracks[0].setMuted(false)
78         e.target.innerText = 'Mic on'
79         e.target.style.backgroundColor = 'cadetblue'
80     }else{
81         await localTracks[0].setMuted(true)
82         e.target.innerText = 'Mic off'
83         e.target.style.backgroundColor = '#EE4B2B'
84     }
85 }
86
87 let toggleCamera = async (e) => {
88     if(localTracks[1].muted){
89         await localTracks[1].setMuted(false)
90         e.target.innerText = 'Camera on'
91         e.target.style.backgroundColor = 'cadetblue'
92     }else{
93         await localTracks[1].setMuted(true)
94         e.target.innerText = 'Camera off'
95         e.target.style.backgroundColor = '#EE4B2B'
96     }
97 }
98
99 document.getElementById('join-btn').addEventListener('click', joinStream)
100 document.getElementById('leave-btn').addEventListener('click', leaveAndRemoveLocalStream)
101 document.getElementById('mic-btn').addEventListener('click', toggleMic)
102 document.getElementById('camera-btn').addEventListener('click', toggleCamera)

```

IMPLEMENTATION

This app can be used for the following:

- tech meet-up groups
- social meet-up groups
- fitness meet-up groups
- hiking meet-up groups

FEATURES

1. Explore your interests
2. Search intended groups
3. Start a group
4. Stay connected

CONCLUSION

The entire project has been developed and deployed as per the requirements stated by exposys data labs, it is found to be bug free as per the testing standards that are implemented. The system at present does not take care of the money payment methods. The system needs more elaborative technicality for its inception and evolution.

The app is weaved around one simple idea **“When we get together and do things that matter to us we are at our best”** and that's what the app does.

This app can be used for the following:

- tech meet-up groups
- social meet-up groups
- fitness meet-up groups
- hiking meet-up groups

TEAM MEMBER DETAILS

NAME	EMAIL
YUKTHA.N	Yuktha163@gmail.com
SAKSHI NAGARAJARAO JADHAV	Sakshijadhav2001k@gmail.com