



MEETUP

Presented by: Yuktha.N & Sakshi Jadhav

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.






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

```
# main.css > {} @media screen and (max-width:1400px)
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: #203A49;
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

JS main.js > client

```
1  const APP_ID = "69bb2373d582420893fe5fda6c90ea2d"
2  const TOKEN = "00669bb2373d582420893fe5fda6c90ea2dIADru+FbNr1gZDgm2SNWlHT+A0ujVYVg4fjgtMDwWaenvR57mw4AAAAEACu7KwLvSXVYQEAAQC/JdVh"
3  const CHANNEL = "Meetup"
4
5  const client = AgoraRTC.createClient({mode:'rtc', codec:'vp8'})
6
7  let localTracks = []
8  let remoteUsers = {}
9
10 let joinAndDisplayLocalStream = async () => {
11
12     client.on('user-published', handleUserJoined)
13
14     client.on('user-left', handleUserLeft)
15
16     let UID = await client.join(APP_ID, CHANNEL, TOKEN, null)
17
18     localTracks = await AgoraRTC.createMicrophoneAndCameraTracks()
19
20     let player = `<div class="video-container" id="user-container-${UID}">
21         <div class="video-player" id="user-${UID}"></div>
22     </div>`
23     document.getElementById('video-streams').insertAdjacentHTML('beforeend', player)
24
25     localTracks[1].play(`user-${UID}`)
26
27     await client.publish([localTracks[0], localTracks[1]])
28 }
29
30 let joinStream = async () => {
31     await joinAndDisplayLocalStream()
32     document.getElementById('join-btn').style.display = 'none'
33     document.getElementById('stream-controls').style.display = 'flex'
34 }
```

```
36 let handleUserJoined = async (user, mediaType) => {
37   remoteUsers[user.uid] = user
38   await client.subscribe(user, mediaType)
39
40   if (mediaType === 'video'){
41     let player = document.getElementById(`user-container-${user.uid}`)
42     if (player != null){
43       player.remove()
44     }
45
46     player = `
```

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




THANK YOU

