Video Wheels-Off Screenshots - Labs/Quickstart/ReactApp/Android

Credentials used in Lab:

Account SID - AC3daf41d39c3368308f467d37c077d33a
Auth Token - 9786b148fe6332fb30f5f54109e5b9e5

API Key - SKe55ee6121b76ebeb1ab3b56fcdf755a0
TWILIO_API_KEY_SECRET - hdHrMFdExdQ29H20YhkynYVGjz9aYtR9

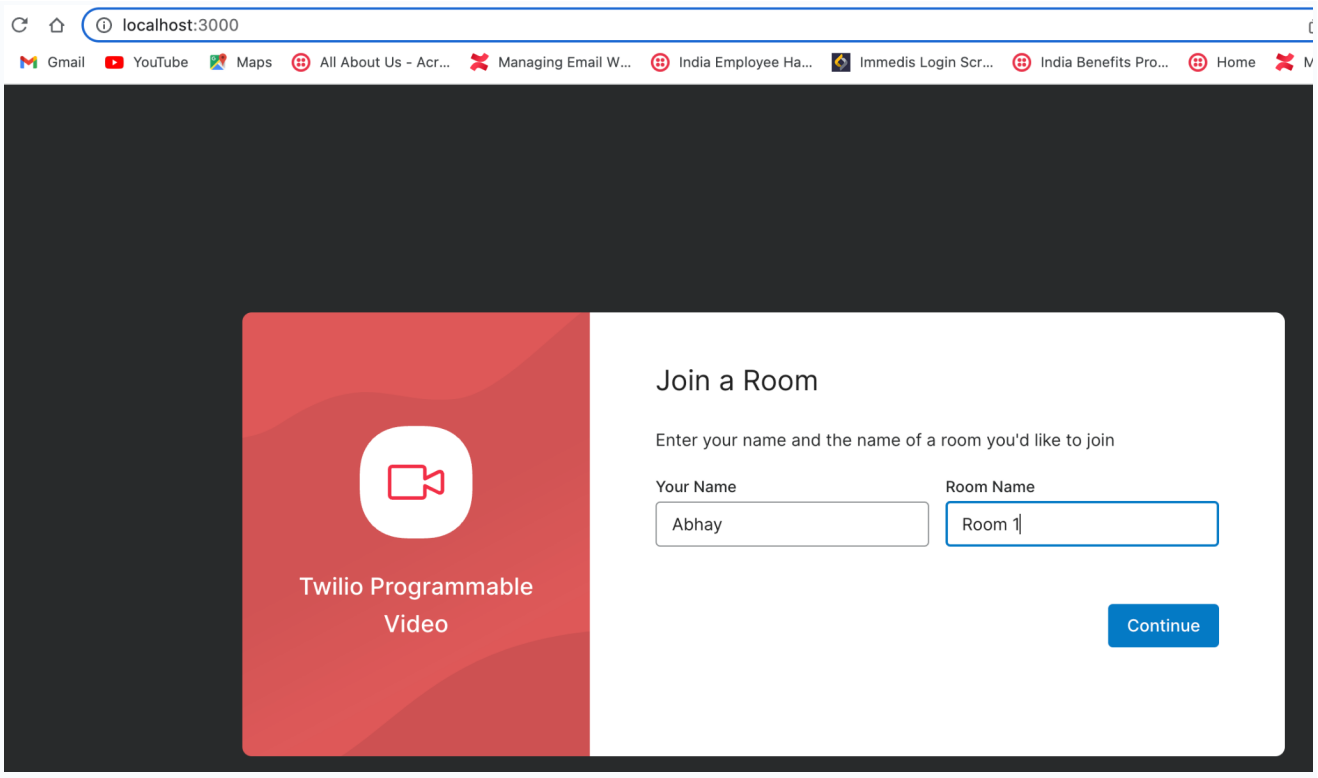
TWILIO_CONVERSATIONS_SERVICE_SID - ISf9b037ceebfc41ae8a84d6c370285428

Ahoy! Demo Apps- REACT DEMO APP

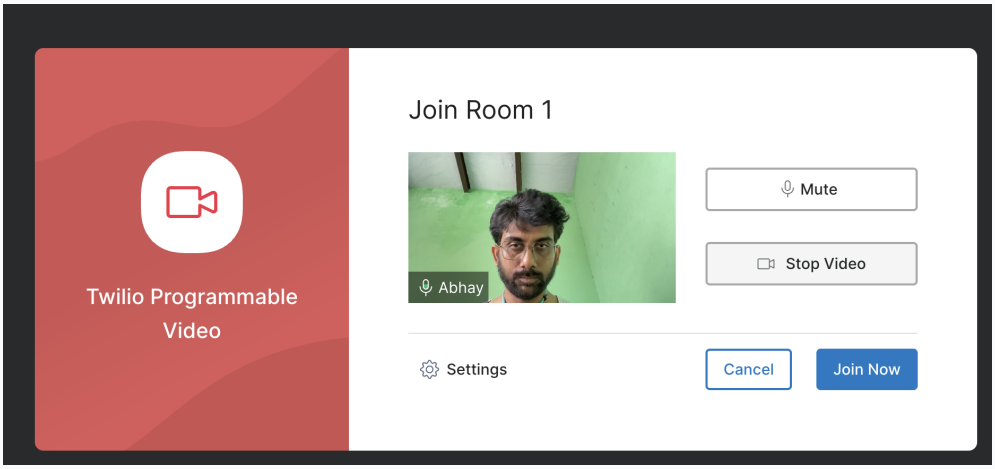
https://github.com/twilio/twilio-video-app-react JS SDK Ahoy! Demo App

Screen-shots of React Demo App:

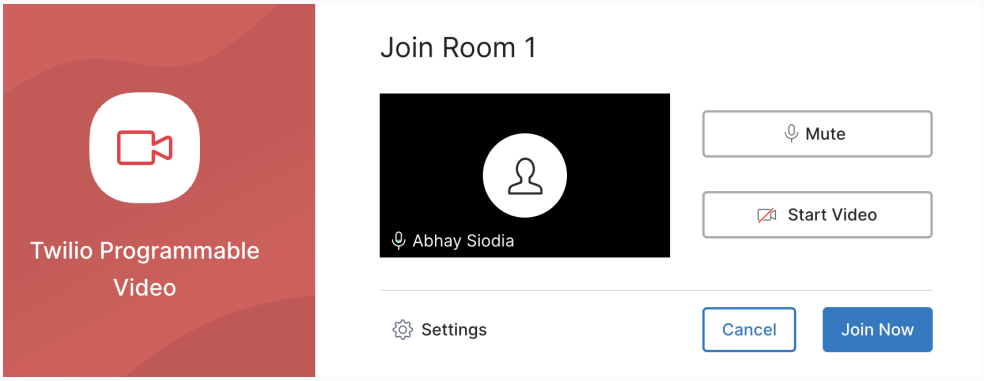
First Screenshot of App while adding entering in Room:



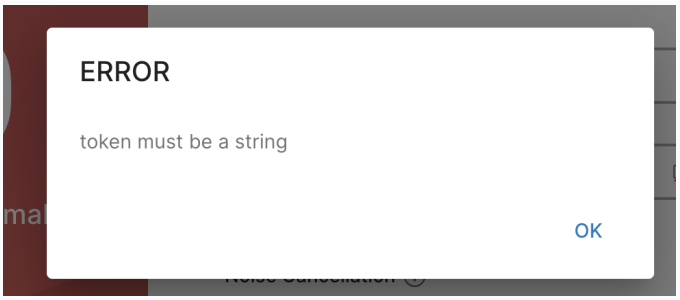
Screenshot #2 Participants joined Room 1.



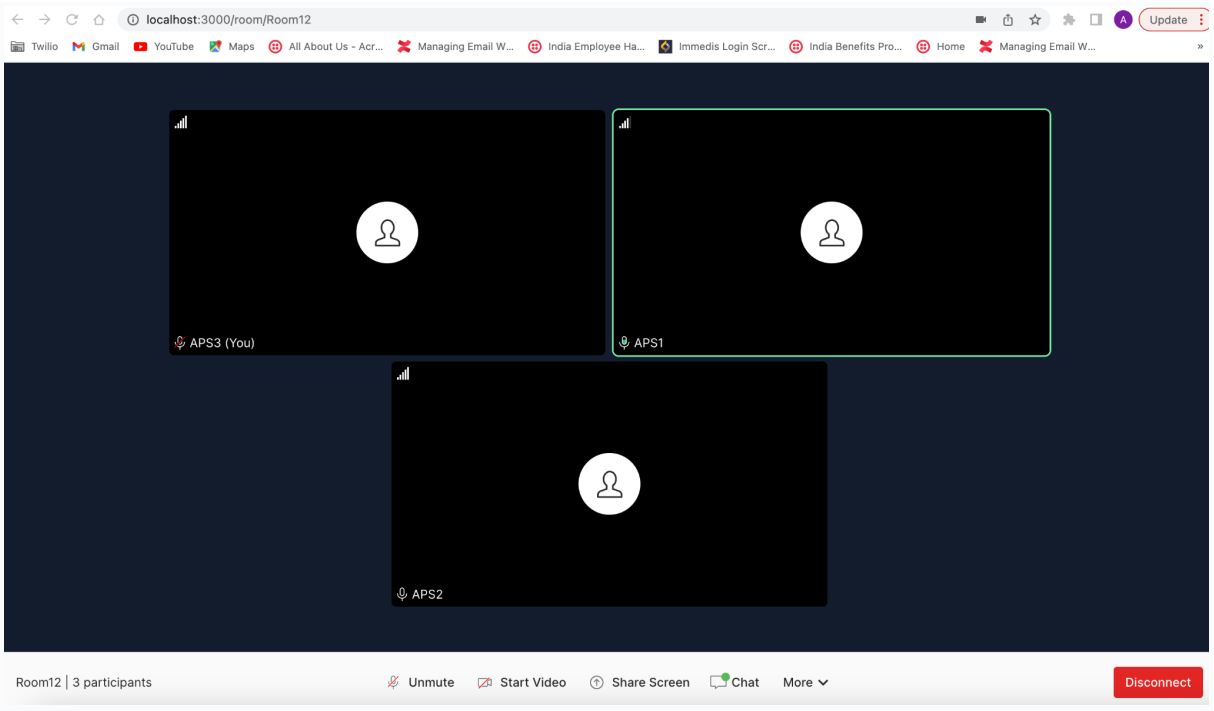
Screenshot #3 Video turned off.



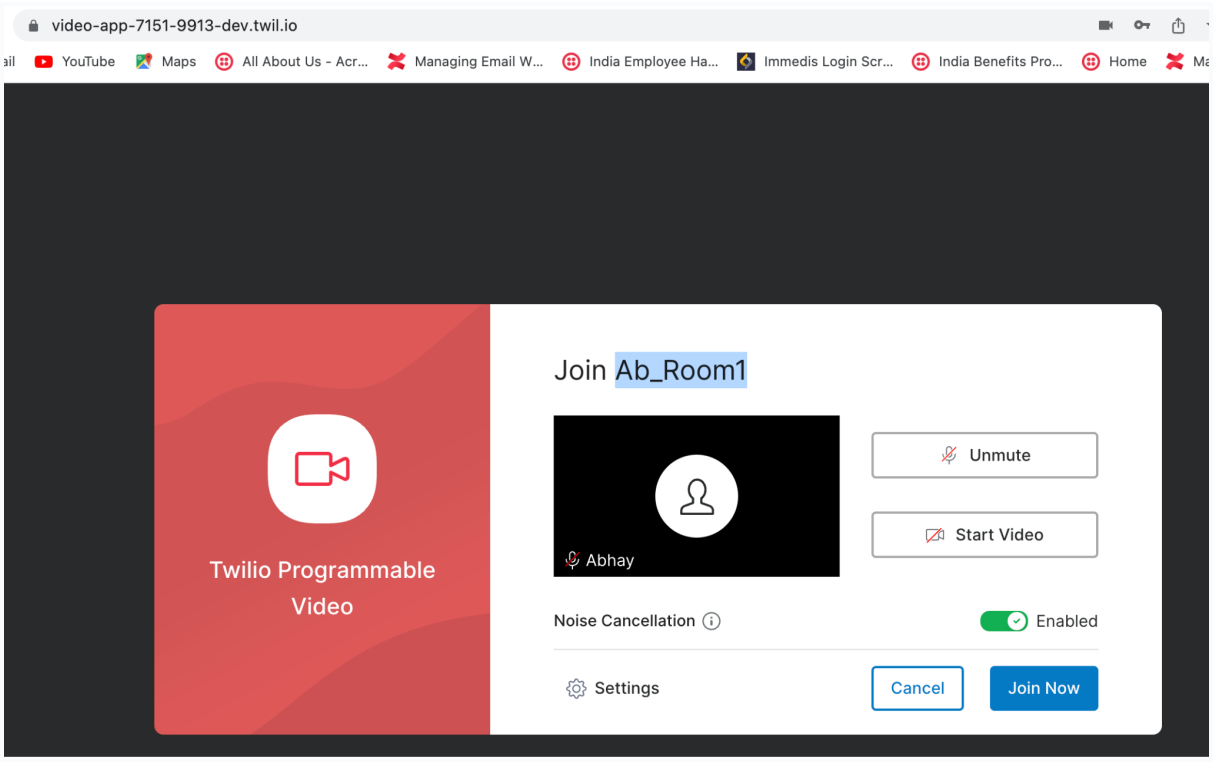
When same user in same room:



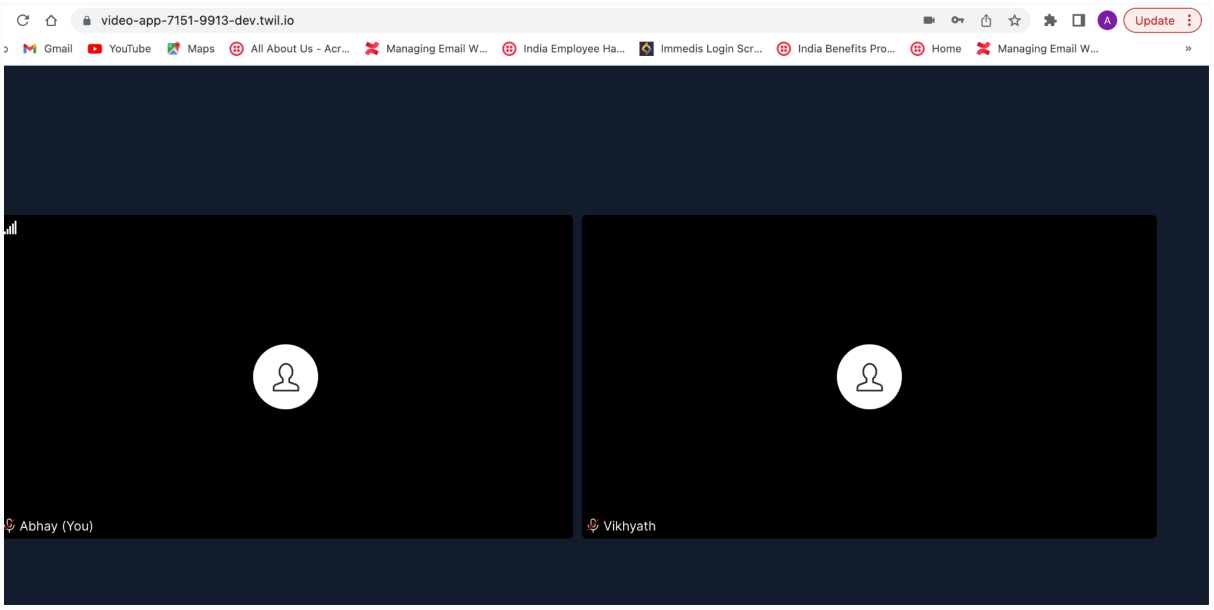
Screenshot #4 Multiple Participants joined Room:



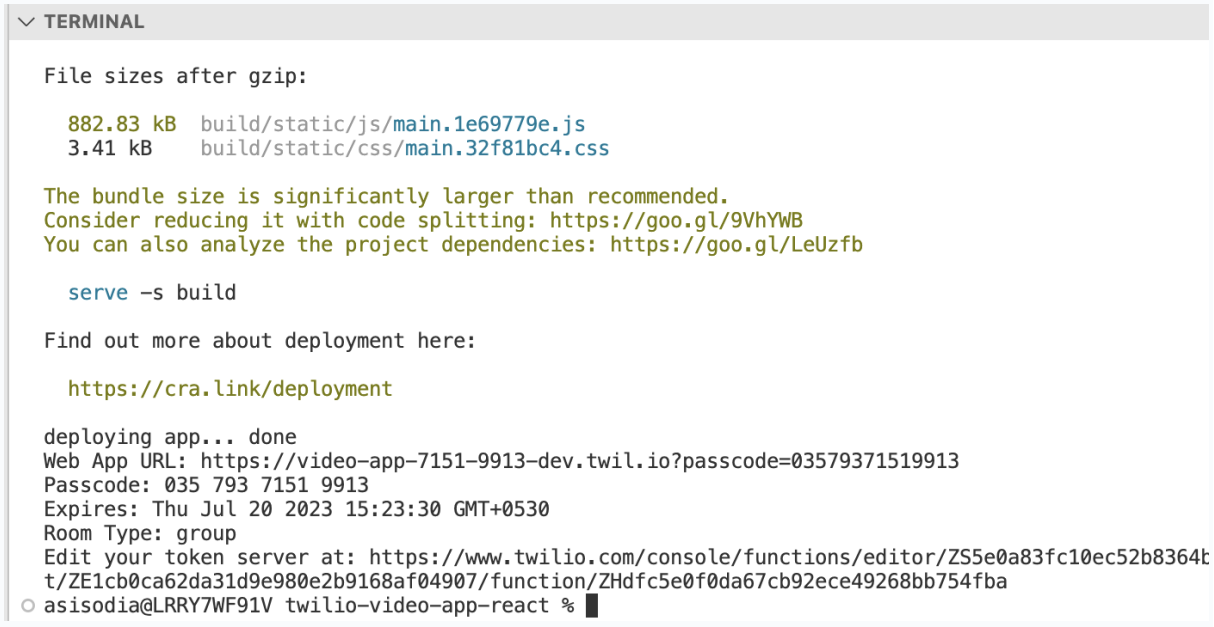
Screenshot #5 Test done in different machine, Joined VS through Public URL:



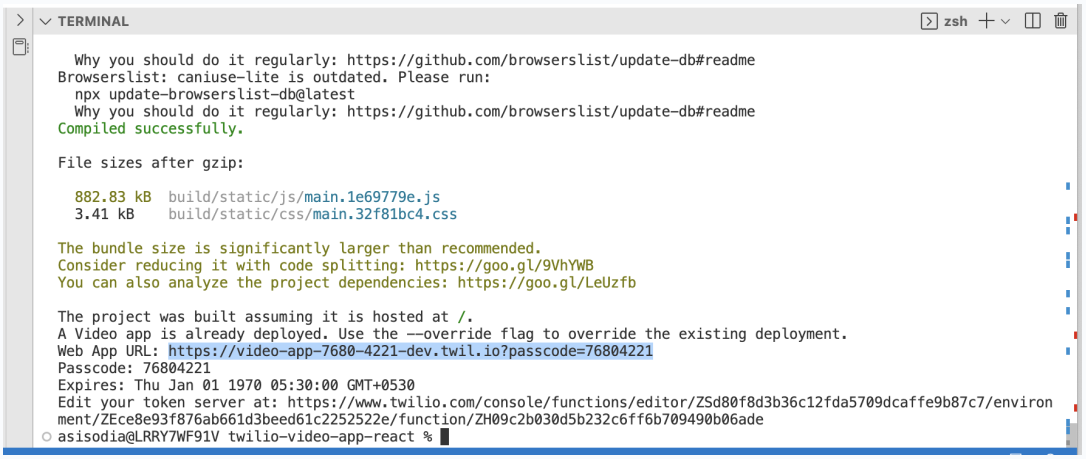
Screenshot #6 Me and VS, joined same Room:



Screenshot #7, Terminal screenshot:

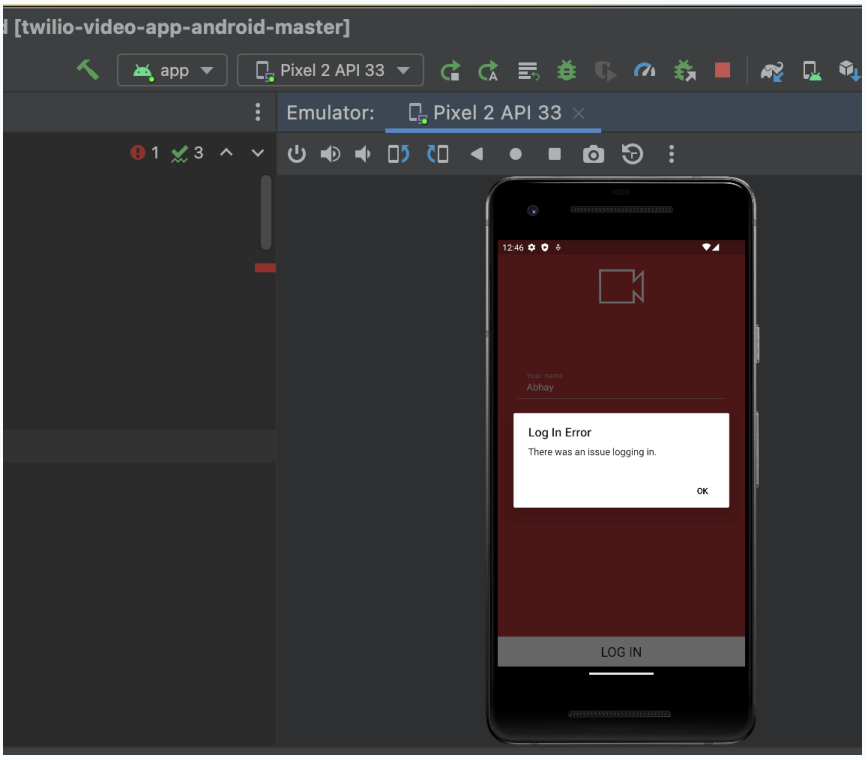


Screenshot #8



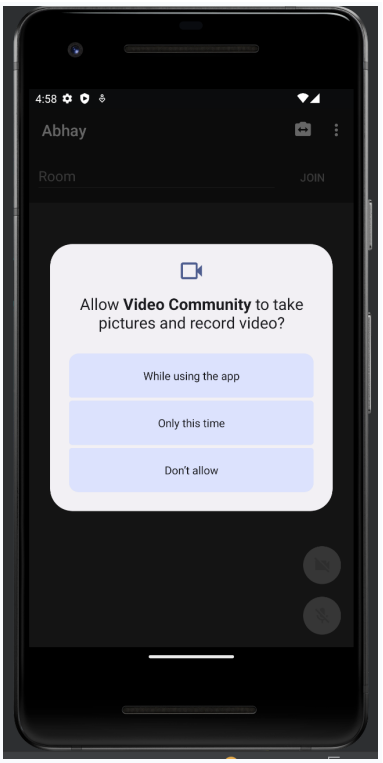
Android SDK Ahoy! Demo App
Below are the screenshots of Android app

Screenshot #1 Lab started using Android studio:



After generating passcode, i am able to login:

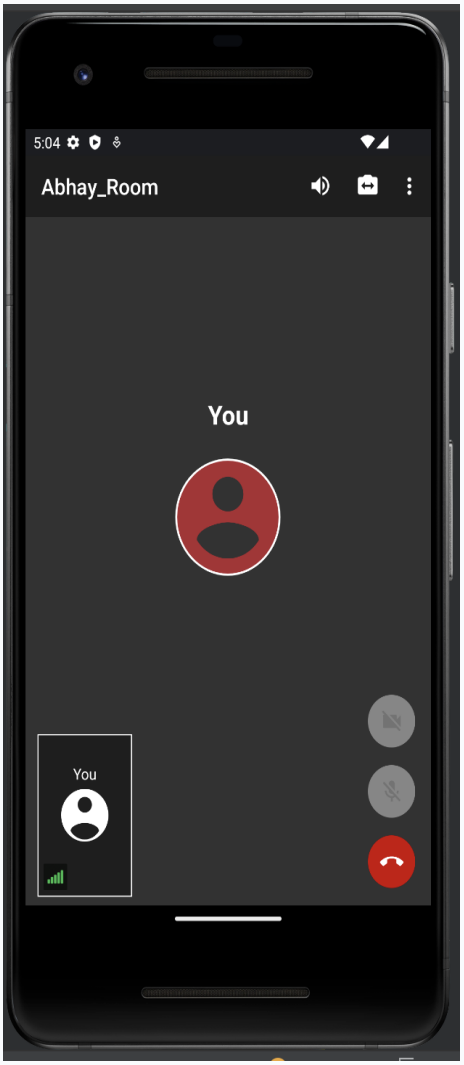
Screenshot #2 After credentials, asked for few permissions::



Screenshot #3 Abhay joined as first participant in Room:

Screenshot #5 Two participant joined same Room(Abhay_Room):

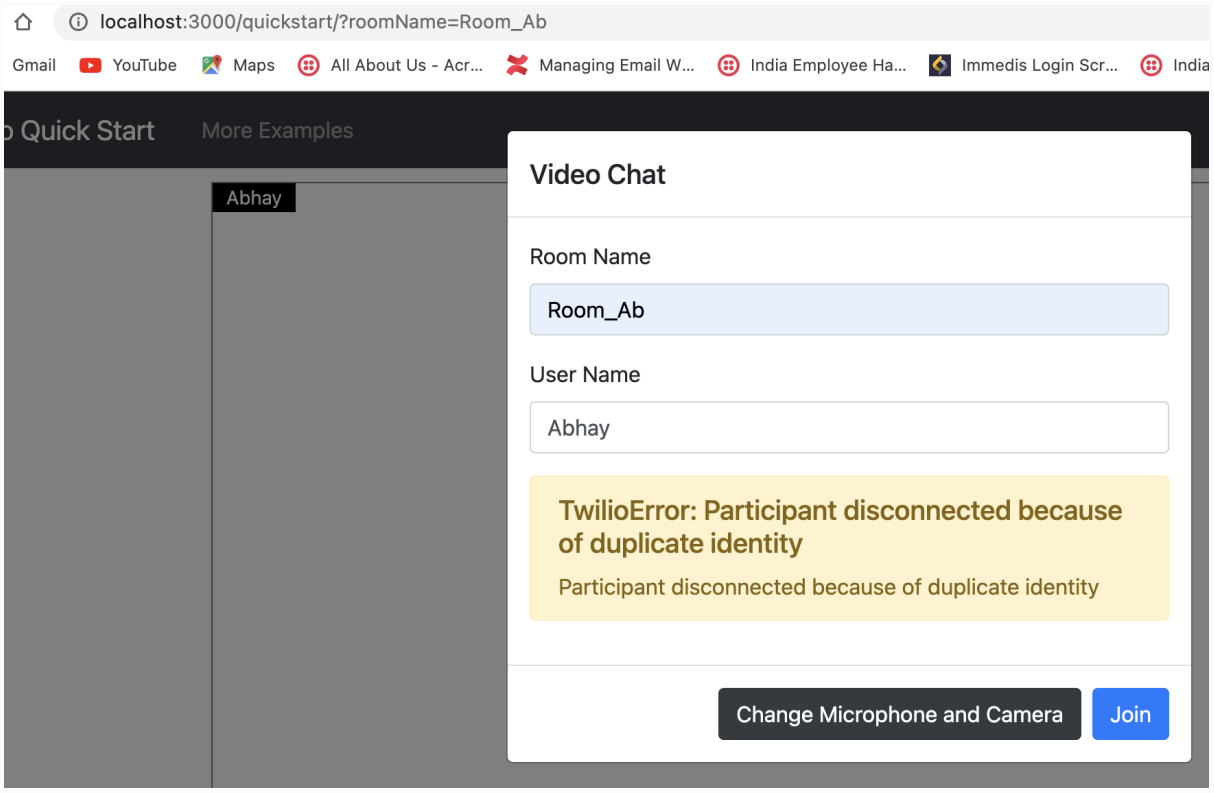
Screenshot #6 Two participant joined same Room(Abhay_Room):


5:04
Abhay_Room
You
You

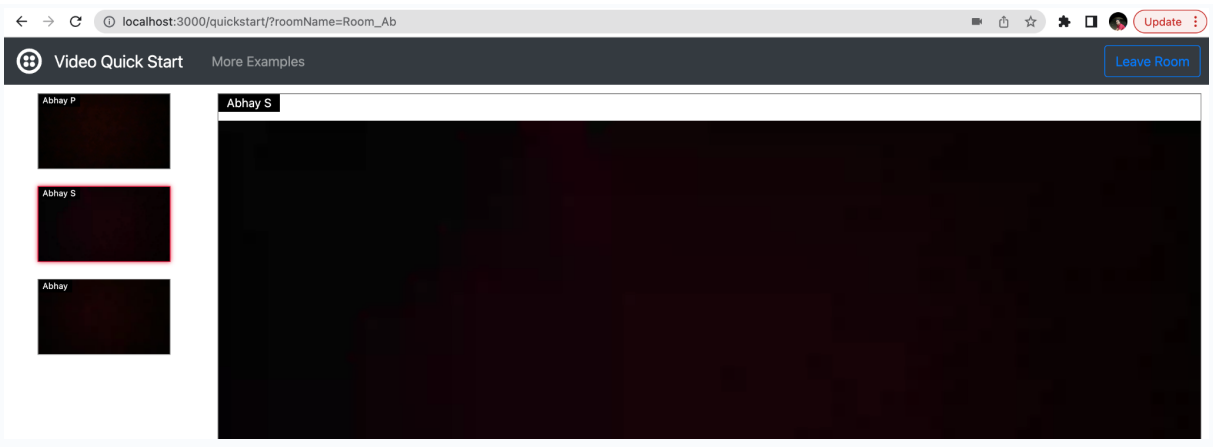
JS SDK Quickstart

Below are the screenshots of JS Quickstart Lab:

Screenshot # Duplicate participant joined same room:



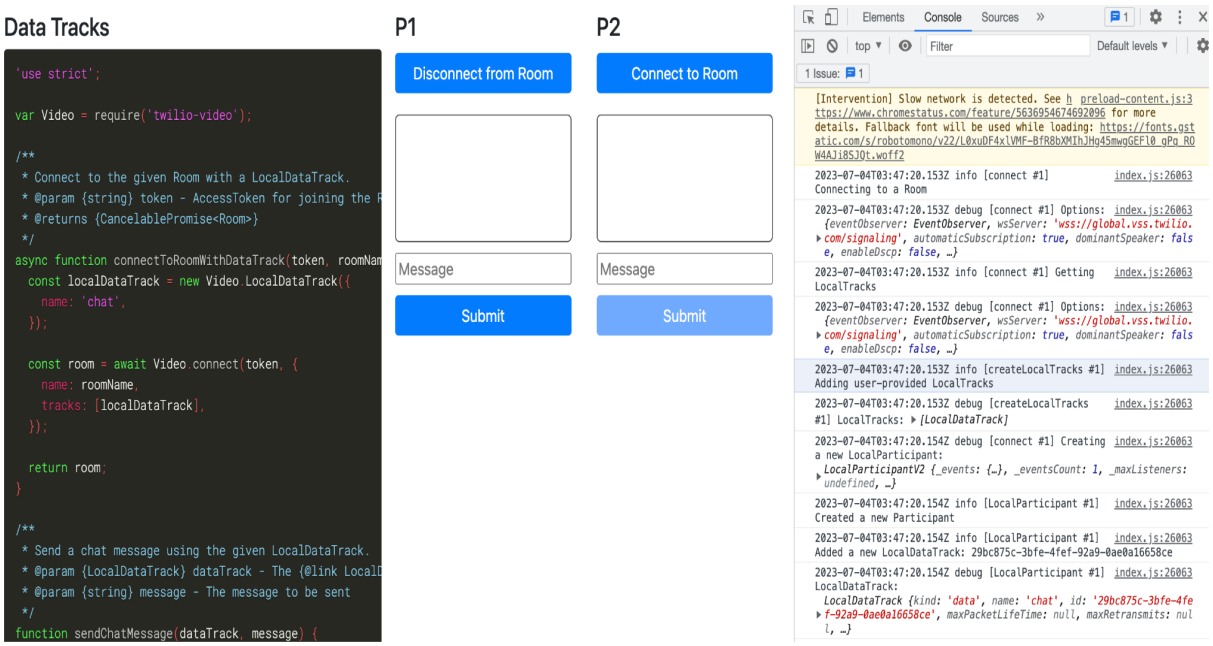
Screenshot # Three Participant joined same Room:



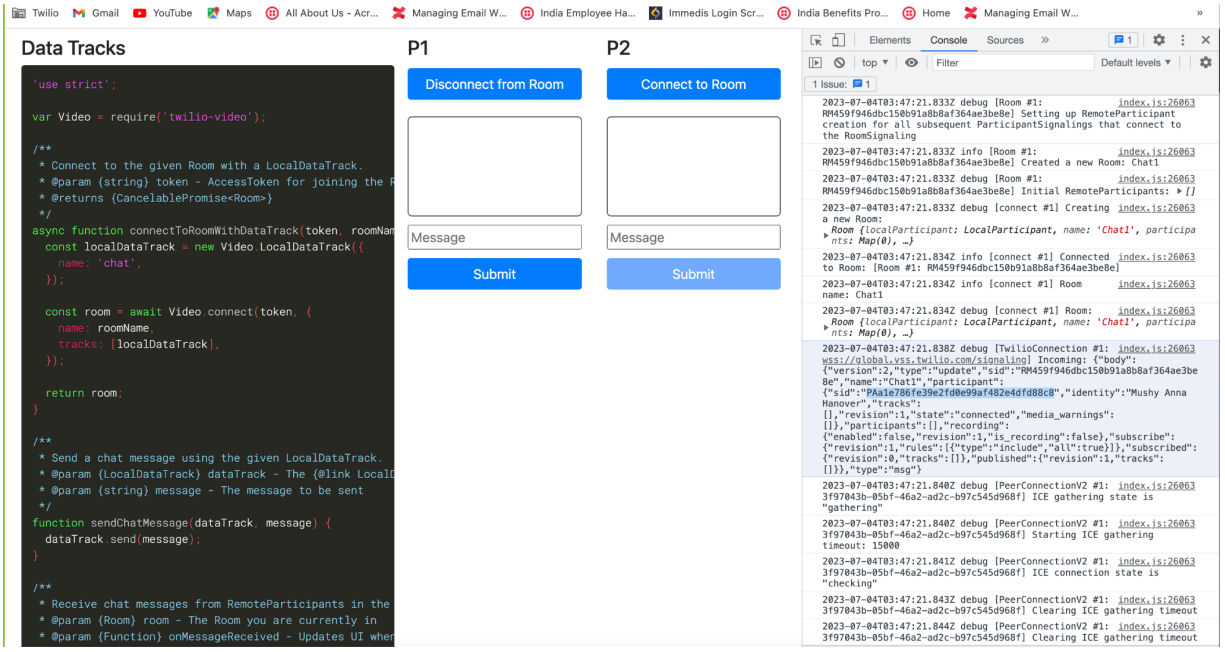
JS QuickStart Examples:

Example Screenshots: DataTracks

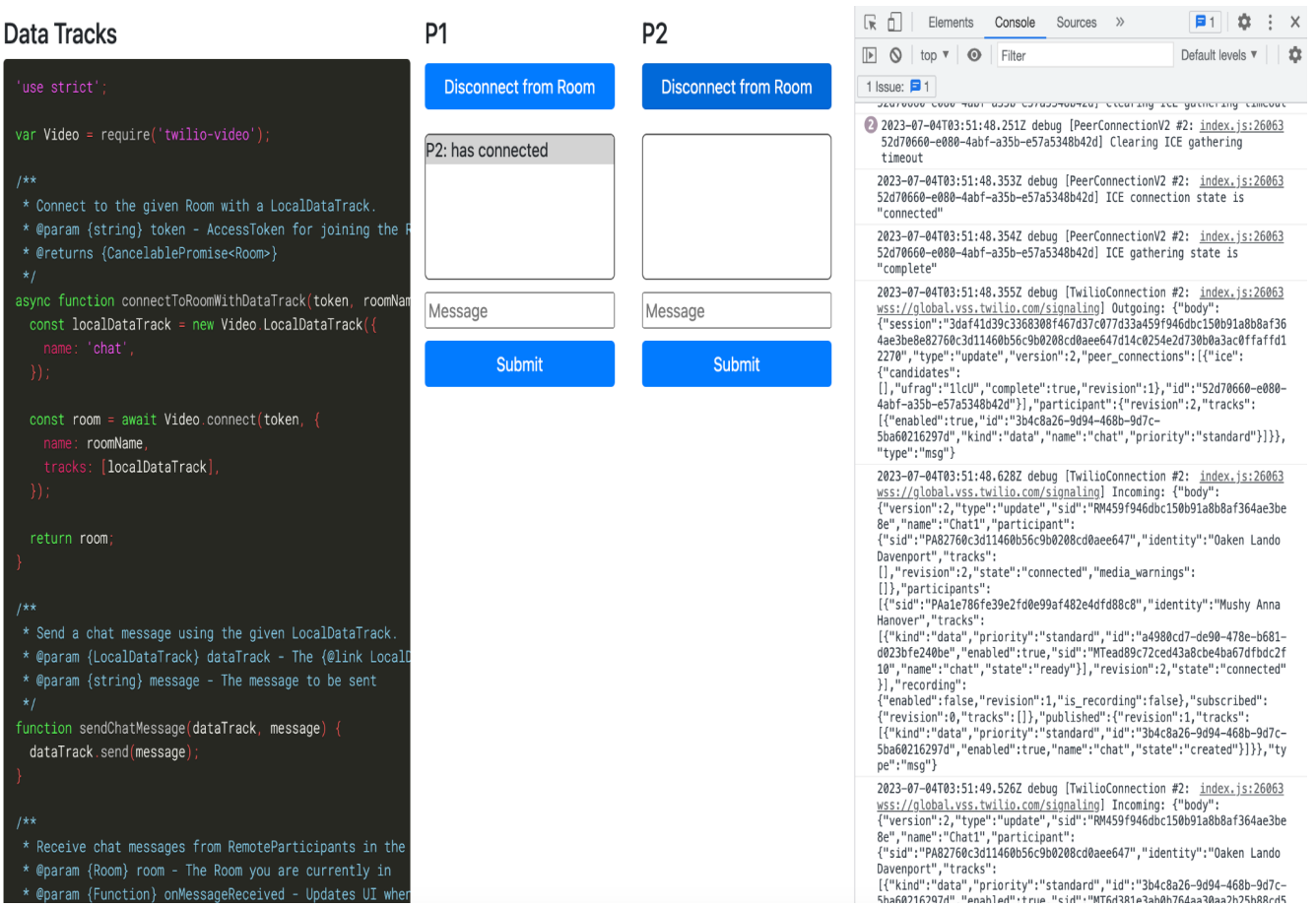
Screenshot #1



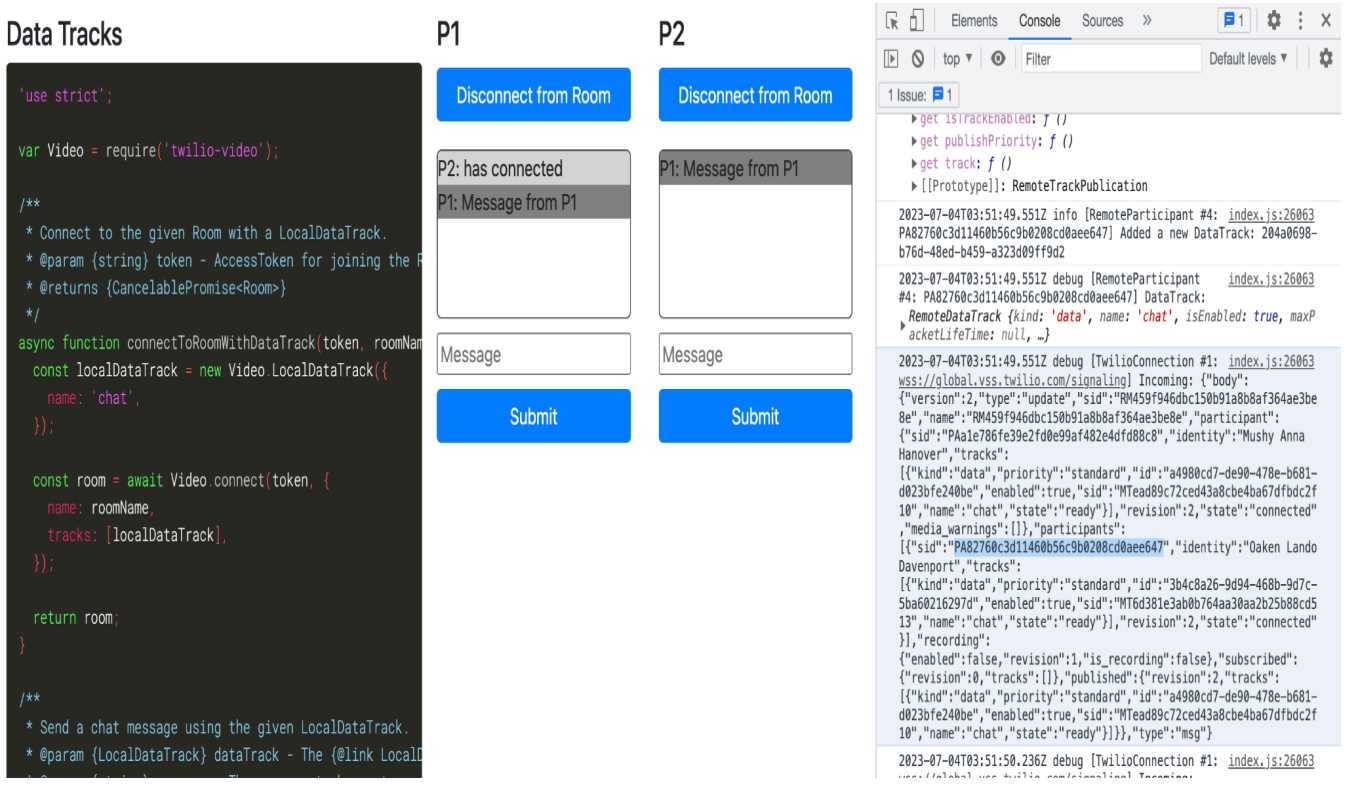
Screenshot #2



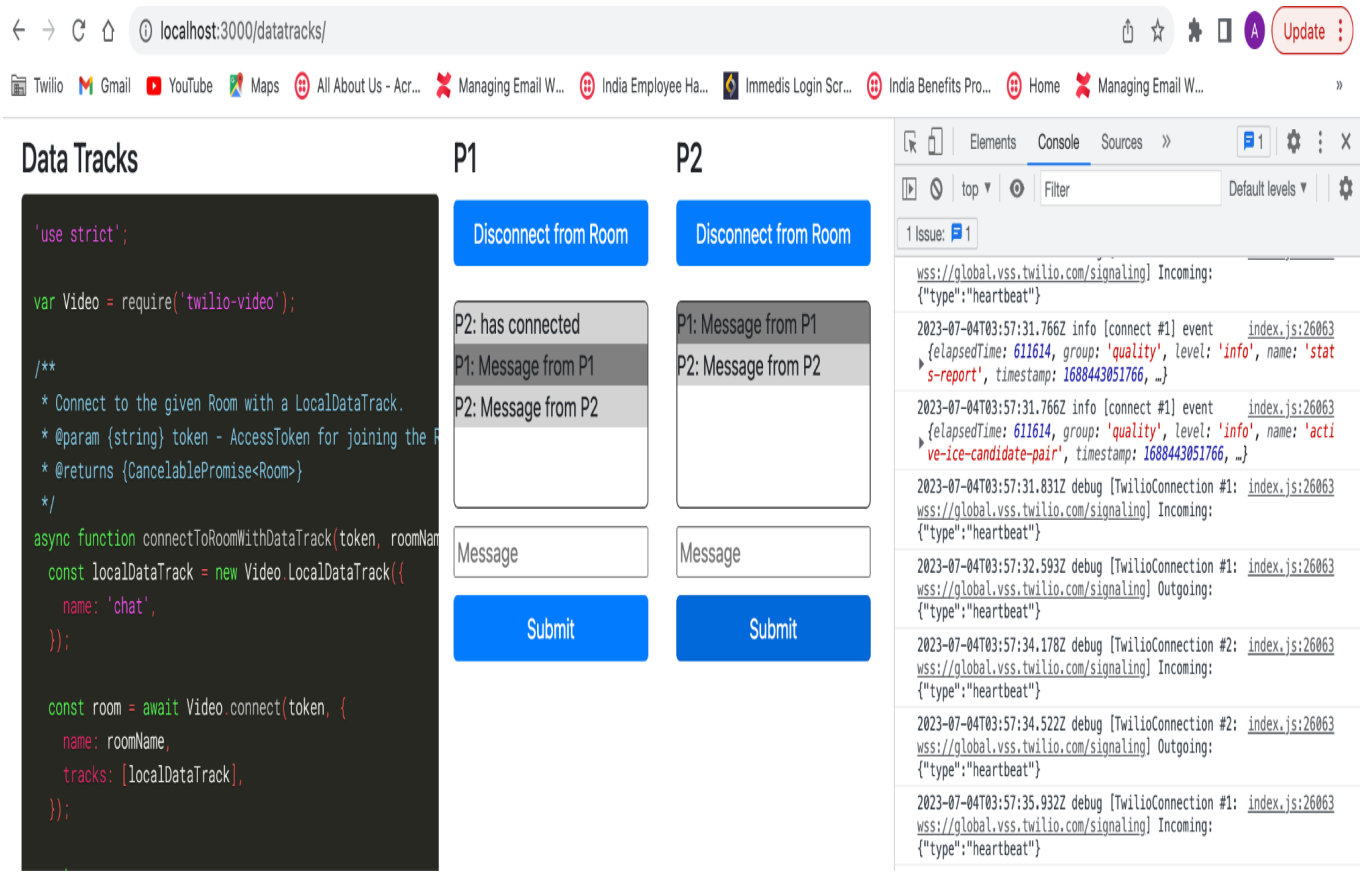
Screenshot #3



Screenshot #4



Screenshot #5


localhost:3000/datatracks/
Update
Twilio
Gmail
YouTube
Maps
All About Us - Acr...
Managing Email W...
India Employee Ha...
Immedis Login Scr...
India Benefits Pro...
Home
Managing Email W...
Data Tracks
'use strict';
var Video = require('twilio-video');
/**
* Connect to the given Room with a LocalDataTrack.
* @param {string} token - AccessToken for joining the R
* @returns {CancelablePromise<Room>}
*/
async function connectToRoomWithDataTrack(token, roomNam
const localDataTrack = new Video.LocalDataTrack({
name: 'chat',
});
const room = await Video.connect(token, {
name: roomName,
tracks: [localDataTrack],
});
P1
Disconnect from Room
P2: has connected
P1: Message from P1
P2: Message from P2
Message
Submit
P2
Disconnect from Room
P1: Message from P1
P2: Message from P2
Message
Submit
Elements
Console
Sources
top
Filter
Default levels
1 Issue: 1
wss://global.vss.twilio.com/signaling] Incoming:
{"type":"heartbeat"}
2023-07-04T03:57:31.766Z info [connect #1] event
index.js:26063
{elapsedTime: 611614, group: 'quality', level: 'info', name: 'stat
s-report', timestamp: 1688443051766, …}
2023-07-04T03:57:31.766Z info [connect #1] event
index.js:26063
{elapsedTime: 611614, group: 'quality', level: 'info', name: 'acti
ve-ice-candidate-pair', timestamp: 1688443051766, …}
2023-07-04T03:57:31.831Z debug [TwilioConnection #1:
index.js:26063
wss://global.vss.twilio.com/signaling] Incoming:
{"type":"heartbeat"}
2023-07-04T03:57:32.593Z debug [TwilioConnection #1:
index.js:26063
wss://global.vss.twilio.com/signaling] Outgoing:
{"type":"heartbeat"}
2023-07-04T03:57:34.178Z debug [TwilioConnection #2:
index.js:26063
wss://global.vss.twilio.com/signaling] Incoming:
{"type":"heartbeat"}
2023-07-04T03:57:34.522Z debug [TwilioConnection #2:
index.js:26063
wss://global.vss.twilio.com/signaling] Outgoing:
{"type":"heartbeat"}
2023-07-04T03:57:35.932Z debug [TwilioConnection #1:
index.js:26063
wss://global.vss.twilio.com/signaling] Incoming:
{"type":"heartbeat"}

Example- Bandwidth Constraints

Screenshot #1

Bandwidth Constraints

```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with the given bandwidth constraints.
 * @param {string} token - Token for joining the Room
 * @param {string} roomName - Room name
 * @param {?number} maxAudioBitrate - Max audio bitrate (bps)
 * @param {?number} maxVideoBitrate - Max video bitrate (bps)
 * @returns {CancelablePromise<Room>}
 */
function connectWithBandwidthConstraints(token, roomName, maxAudioBitrate, maxVideoBitra
  return Video.connect(token, {
    maxAudioBitrate: maxAudioBitrate,
    maxVideoBitrate: maxVideoBitrate,
    name: roomName
  });
}
```

Participant's Media

Select Bandwidth Constraints

Max Audio Bitrate (bps)

16000

Max Video Bitrate (bps)

64000

Connect to Room

Screenshot #2

Bandwidth Constraints

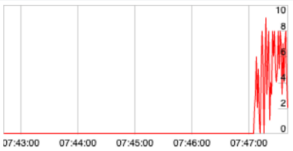
```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with the given bandwidth constraints.
 * @param {string} token - Token for joining the Room
 * @param {string} roomName - Room name
 * @param {?number} maxAudioBitrate - Max audio bitrate (bps)
 * @param {?number} maxVideoBitrate - Max video bitrate (bps)
 * @returns {CancelablePromise<Room>}
 */
function connectWithBandwidthConstraints(token, roomName, maxAudioBitrate, maxVideoBitra
  return Video.connect(token, {
    maxAudioBitrate: maxAudioBitrate,
    maxVideoBitrate: maxVideoBitrate,
    name: roomName
  });
}

/**
 * Update the bandwidth constraints of a Room.
 * @param {Room} room - The Room whose bandwidth constraints have to be updated
 * @param {?number} maxAudioBitrate - Max audio bitrate (bps)
 * @param {?number} maxVideoBitrate - Max video bitrate (bps)
 */
function updateBandwidthConstraints(room, maxAudioBitrate, maxVideoBitrate) {
  room.localParticipant.setParameters({
    maxAudioBitrate: maxAudioBitrate,
    maxVideoBitrate: maxVideoBitrate
  });
}

exports.connectWithBandwidthConstraints = connectWithBandwidthConstraints;
exports.updateBandwidthConstraints = updateBandwidthConstraints;
```

Participant's Media

Select Bandwidth Constraints

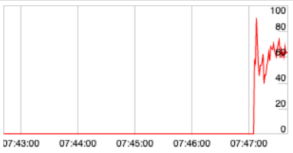
Max Audio Bitrate (bps)

16000



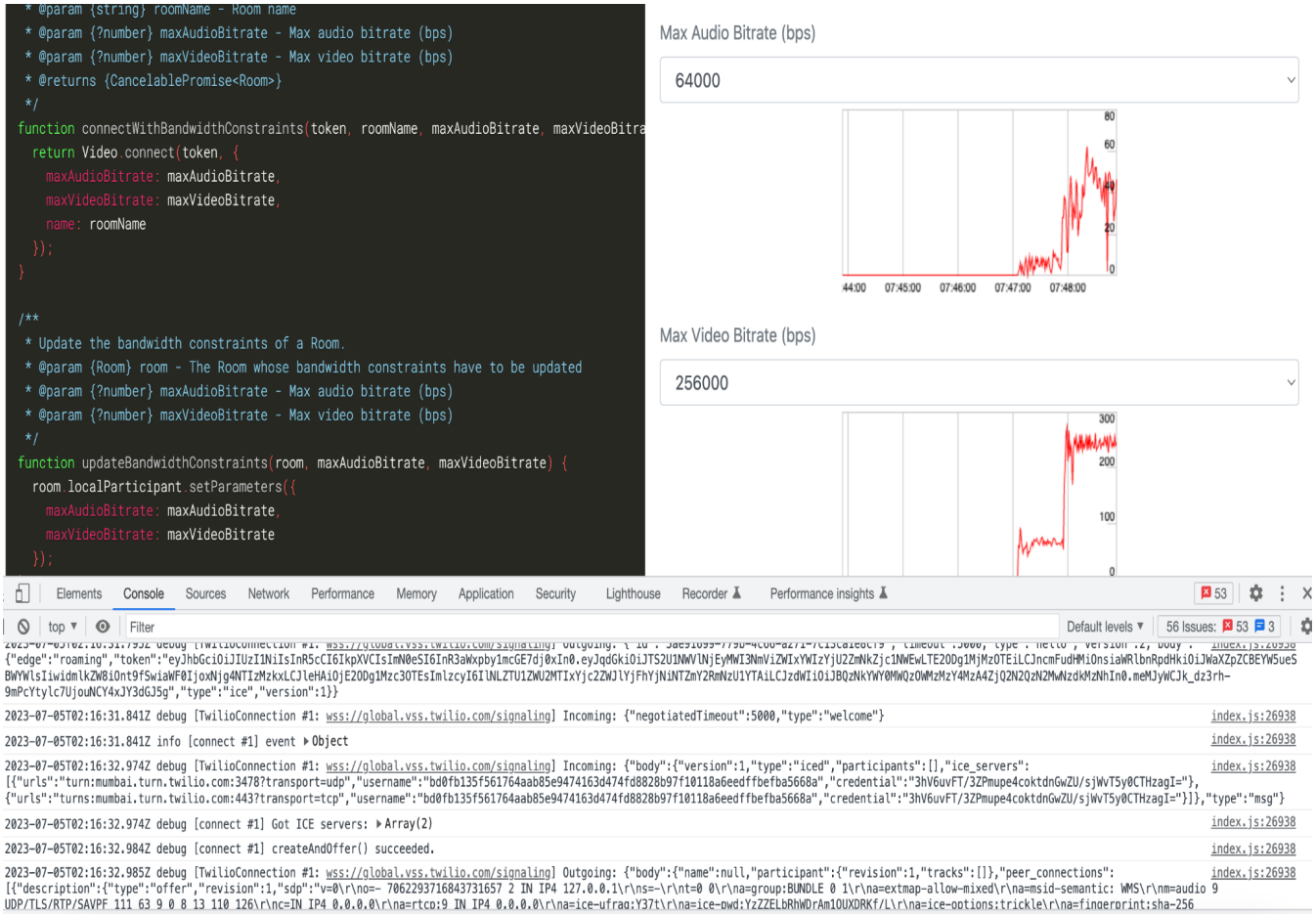
Max Video Bitrate (bps)

64000

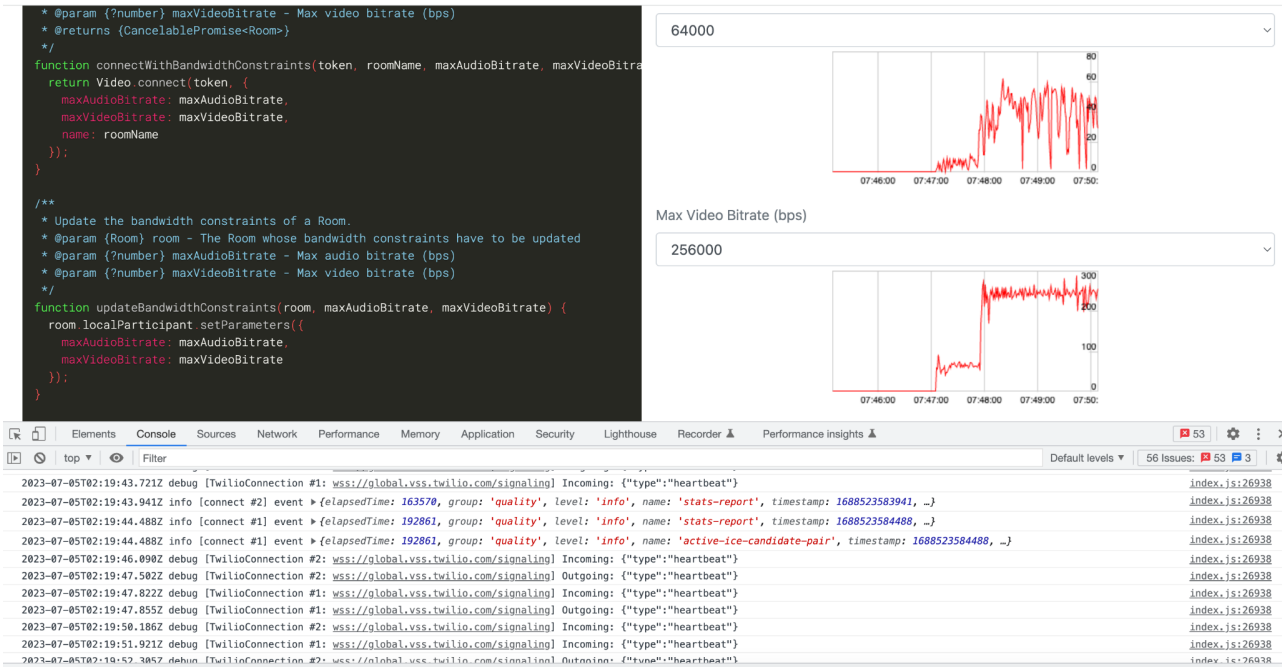


Disconnect from Room

Screenshot #3

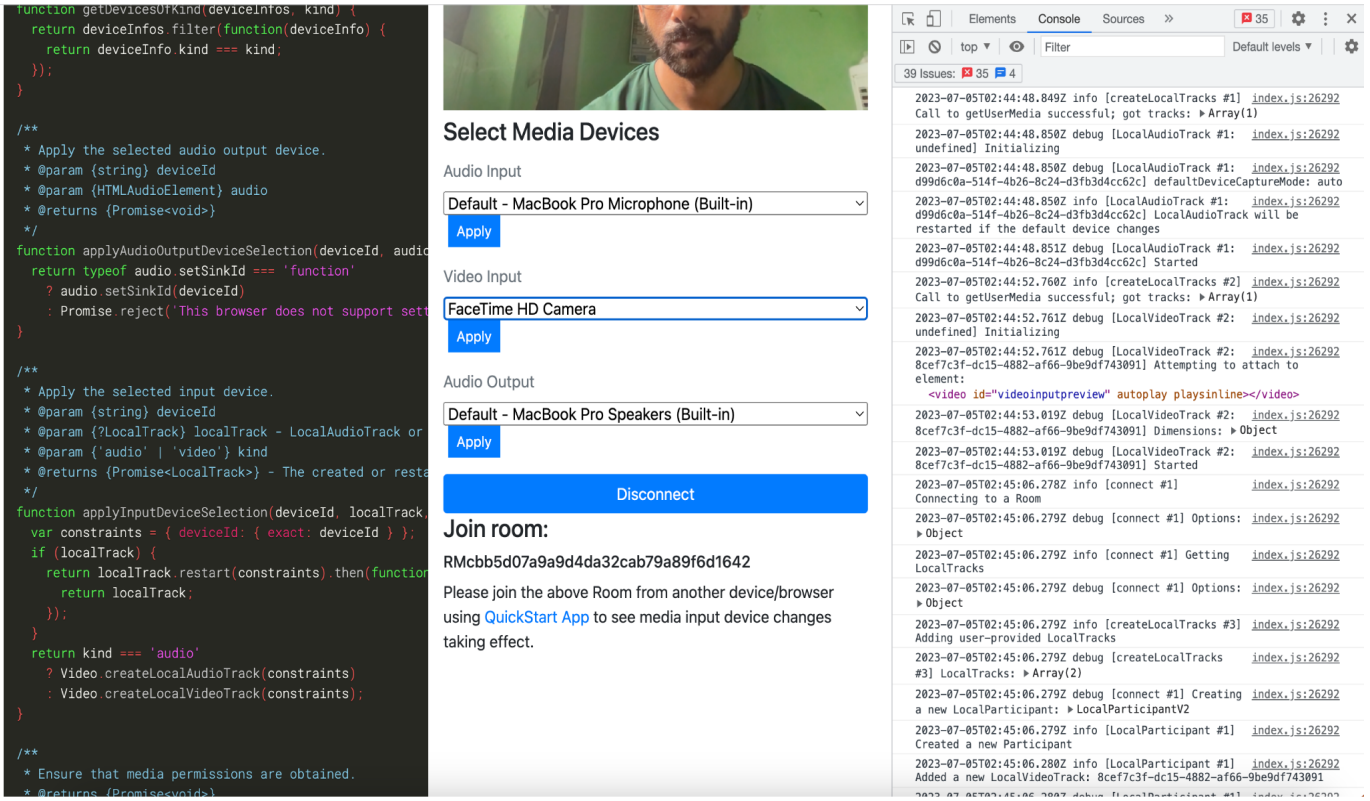


Screenshot #4

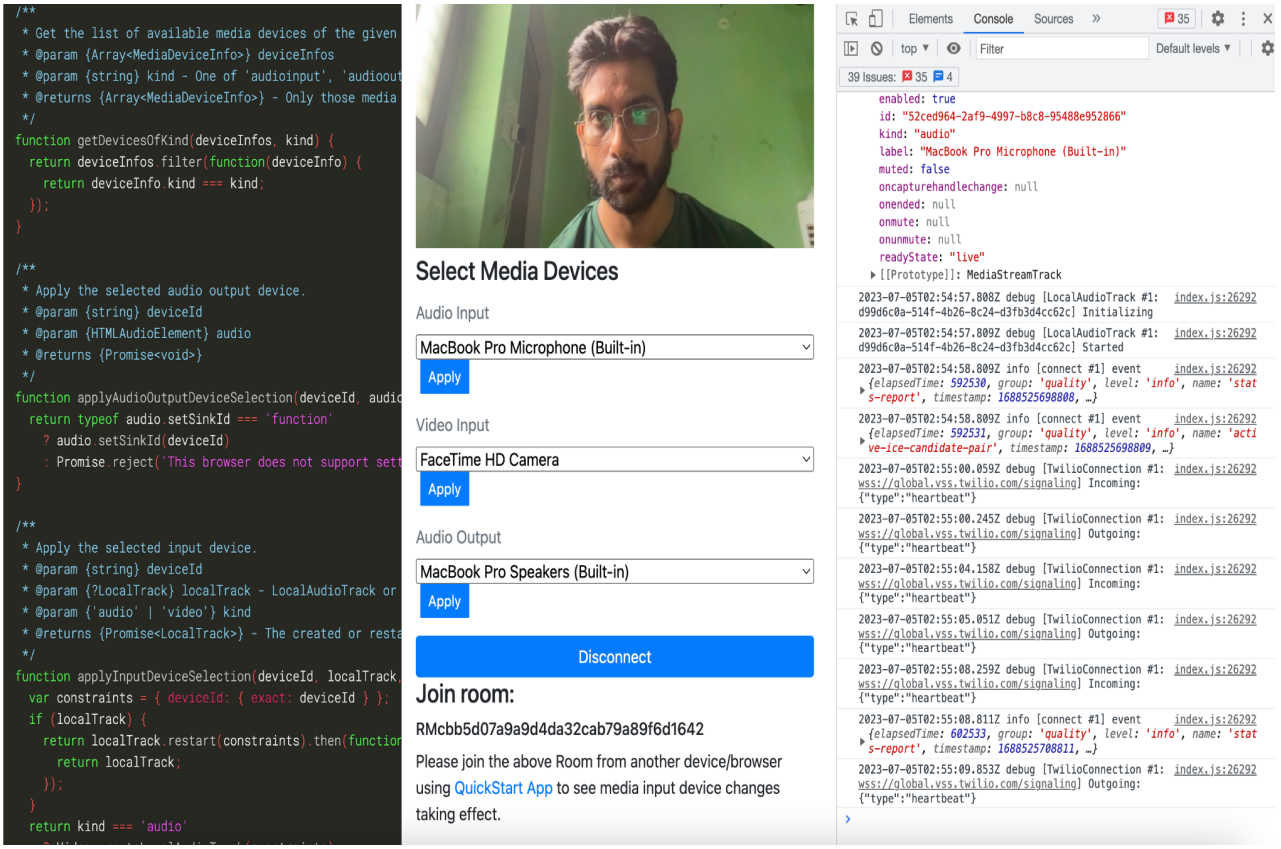


Example- Media Device Selection

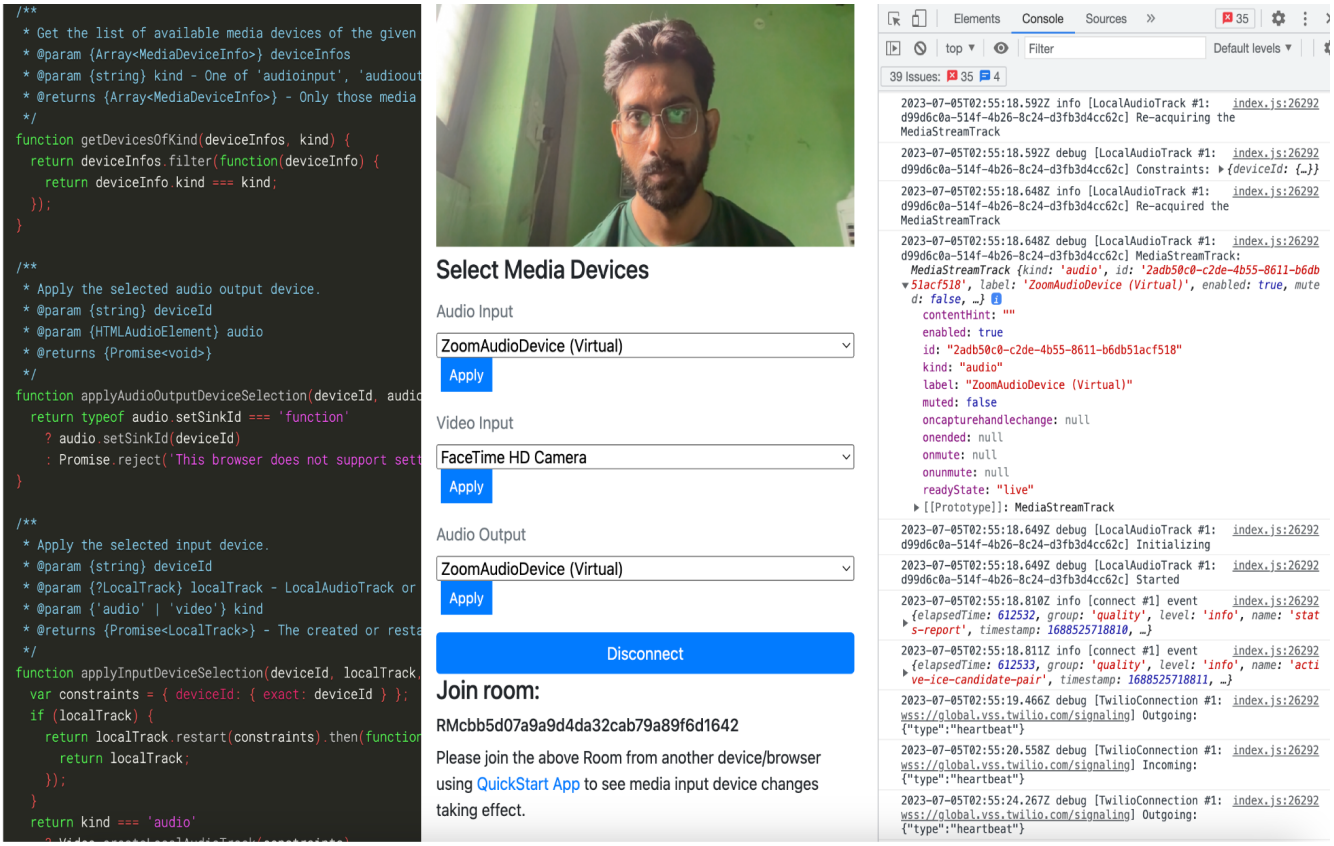
Screenshot #1



Screenshot #2

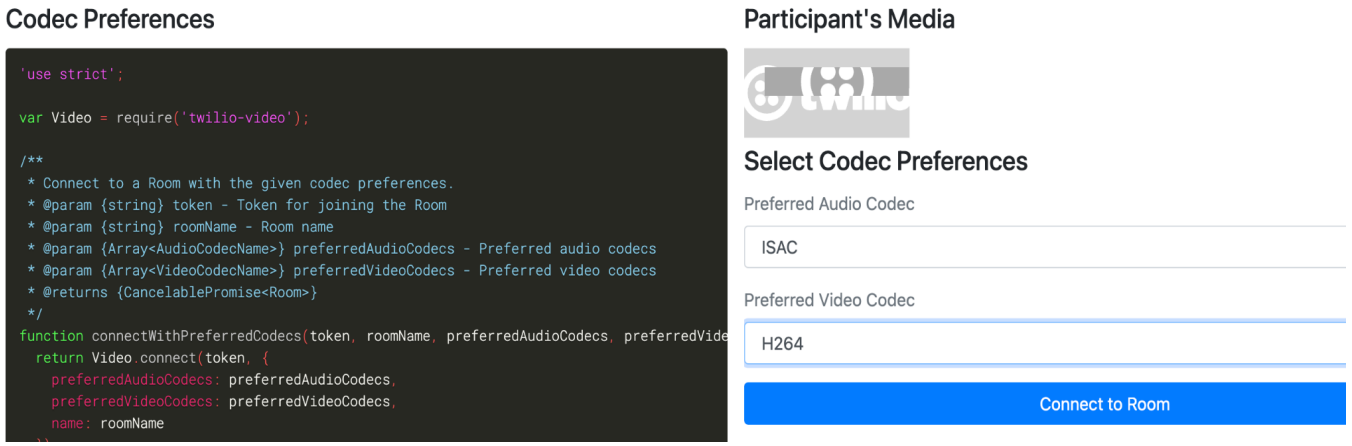


Screenshot #3



Example - Codec Preferences

Screenshot #1



Screenshot #2

Codec Preferences

```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with the given codec preferences.
 * @param {string} token - Token for joining the Room
 * @param {string} roomName - Room name
 * @param {Array<AudioCodecName>} preferredAudioCodecs -
 * @param {Array<VideoCodecName>} preferredVideoCodecs -
 * @returns {CancelablePromise<Room>}
 */
function connectWithPreferredCodecs(token, roomName, pre
  return Video.connect(token, {
    preferredAudioCodecs: preferredAudioCodecs,
    preferredVideoCodecs: preferredVideoCodecs,
    name: roomName
  });
}

exports.connectWithPreferredCodecs = connectWithPreferre
```

Participant's Media

Select Codec Preferences

Preferred Audio Codec

ISAC

Selected Audio Codec: opus

Preferred Video Codec

H264

Selected Video Codec: H264

Disconnect from Room

Screenshot #3

Codec Preferences

```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with the given codec preferences.
 * @param {string} token - Token for joining the Room
 * @param {string} roomName - Room name
 * @param {Array<AudioCodecName>} preferredAudioCodecs -
 * @param {Array<VideoCodecName>} preferredVideoCodecs -
 * @returns {CancelablePromise<Room>}
 */
function connectWithPreferredCodecs(token, roomName, pre
  return Video.connect(token, {
    preferredAudioCodecs: preferredAudioCodecs,
    preferredVideoCodecs: preferredVideoCodecs,
    name: roomName
  });
}

exports.connectWithPreferredCodecs = connectWithPreferre
```

Participant's Media

Select Codec Preferences

Preferred Audio Codec

Opus

Selected Audio Codec: opus

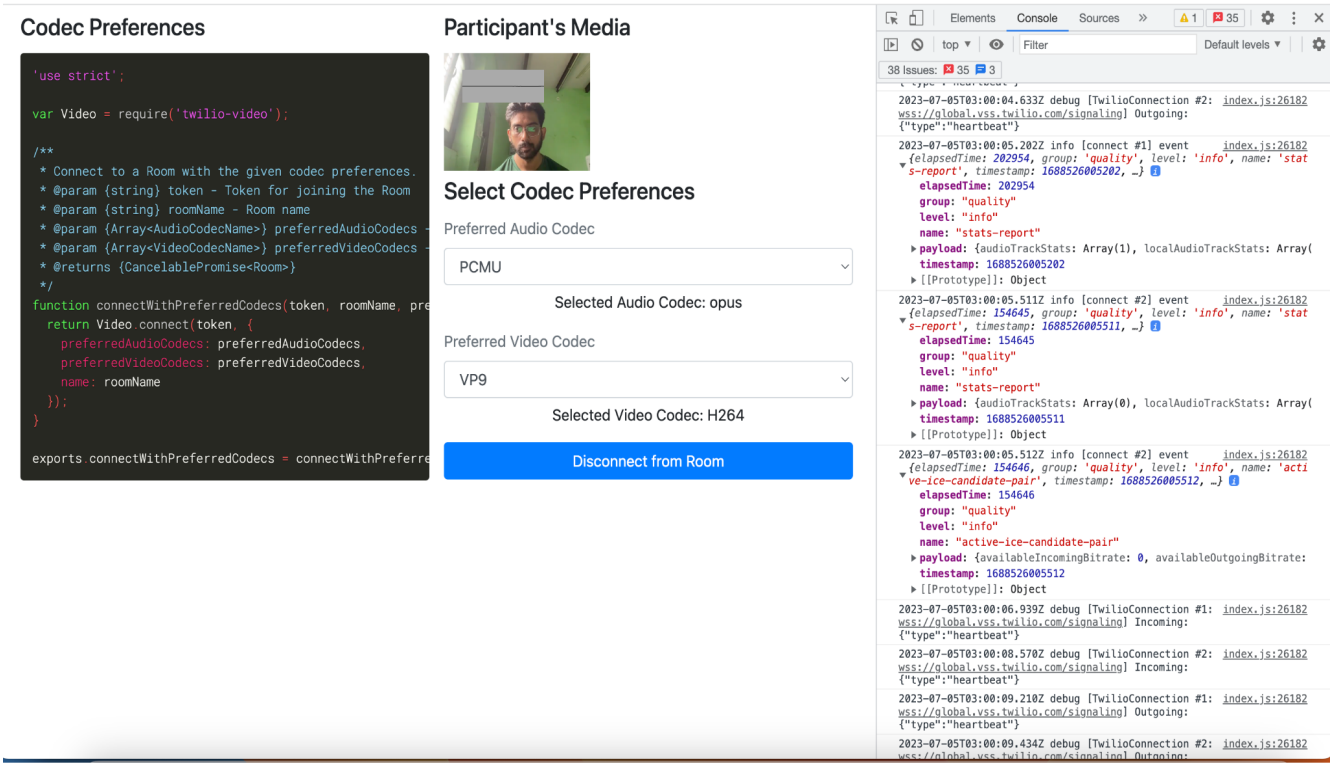
Preferred Video Codec

VP9

Selected Video Codec: H264

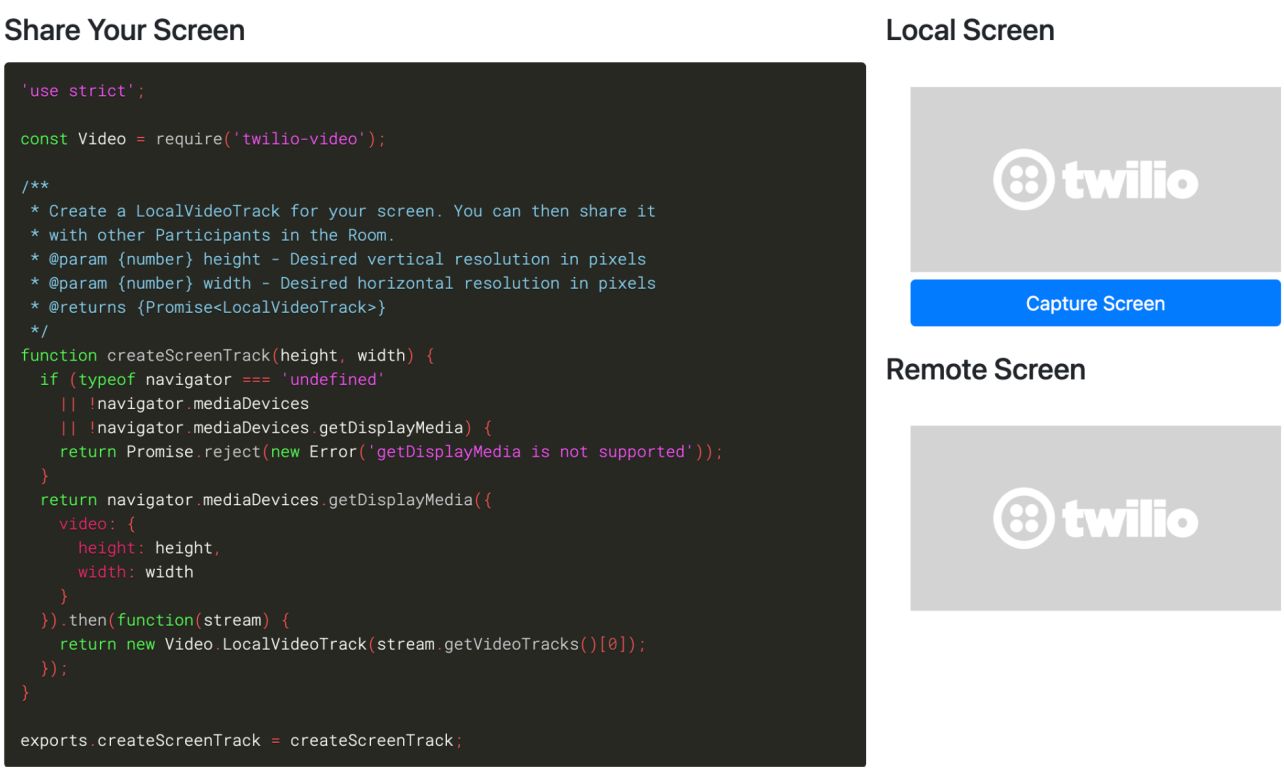
Disconnect from Room

Screenshot #4

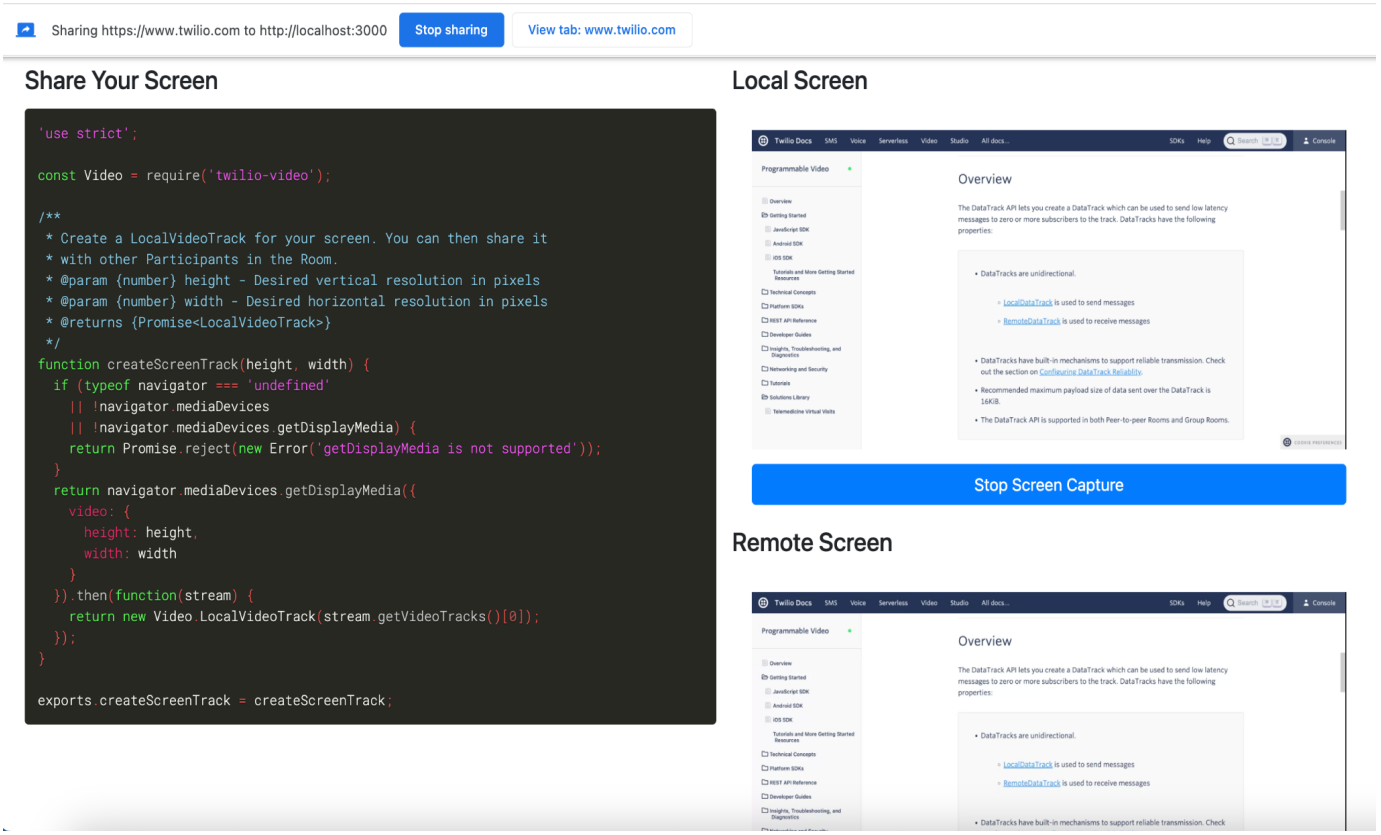


Example - Share Your Screen

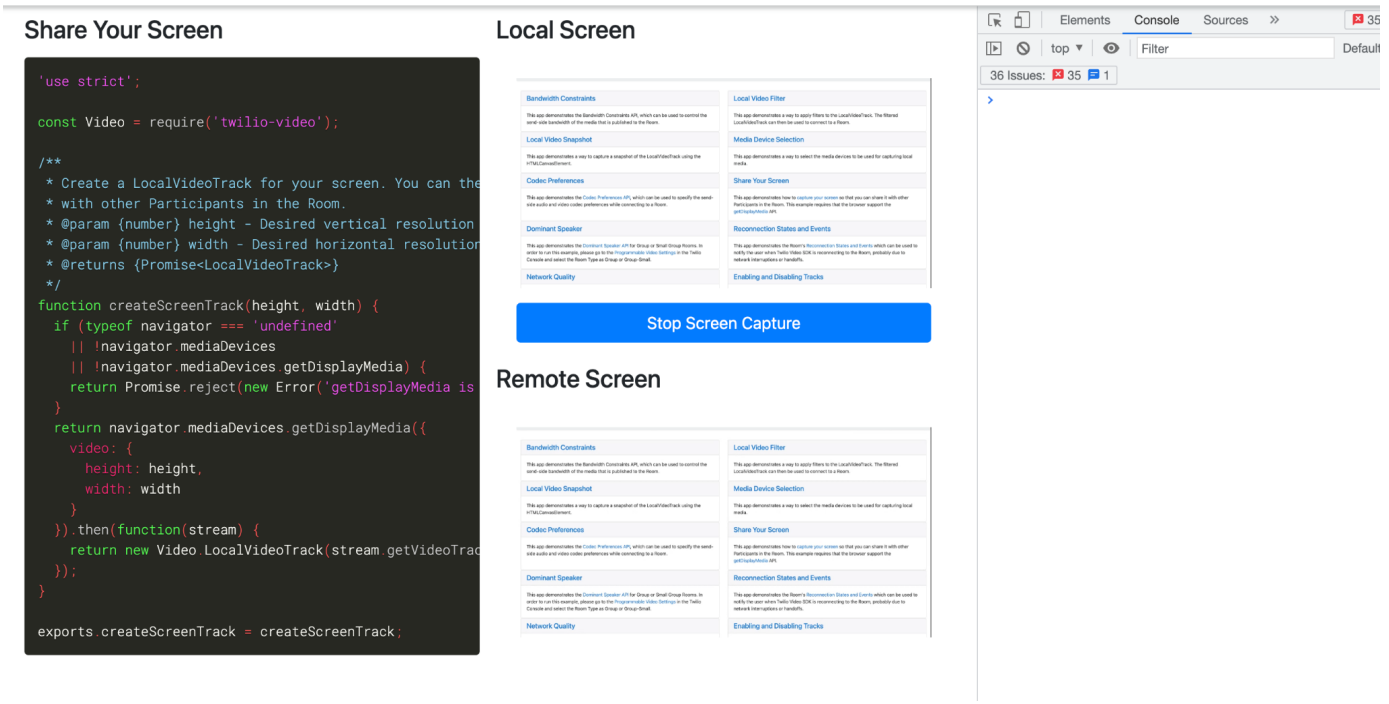
Screenshot #1



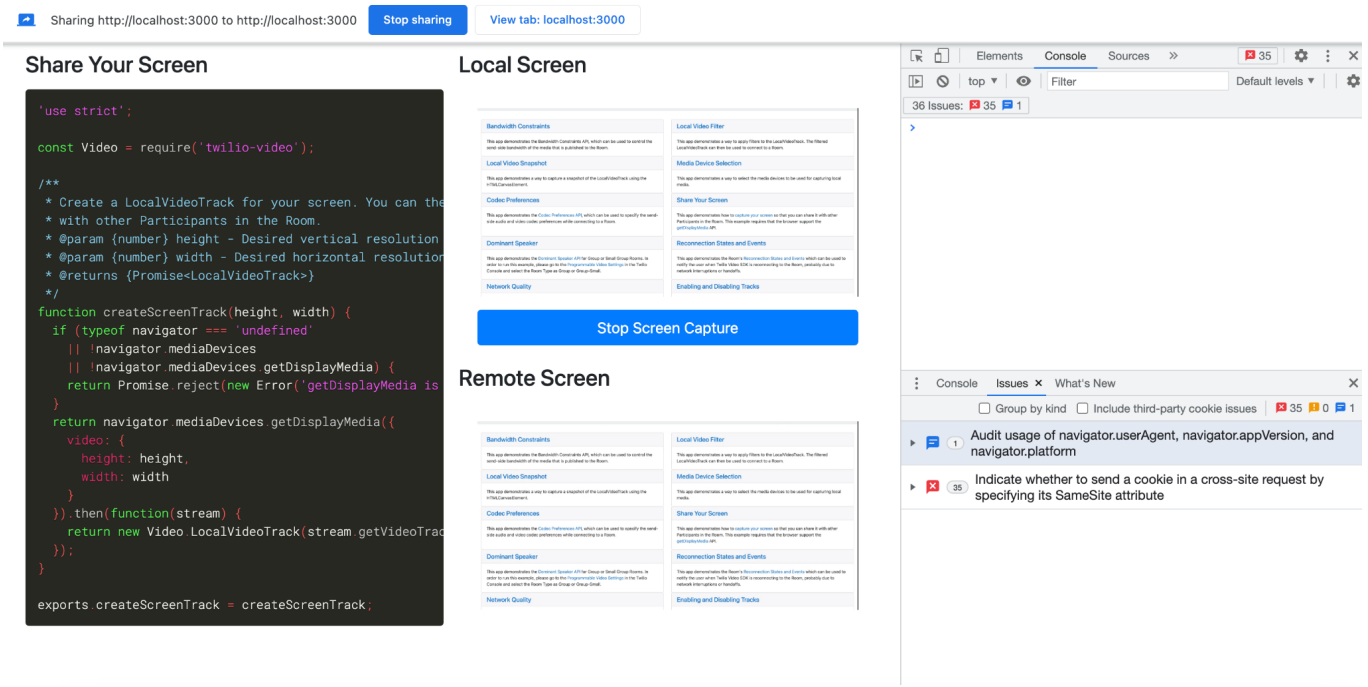
Screenshot #2



Screenshot #3

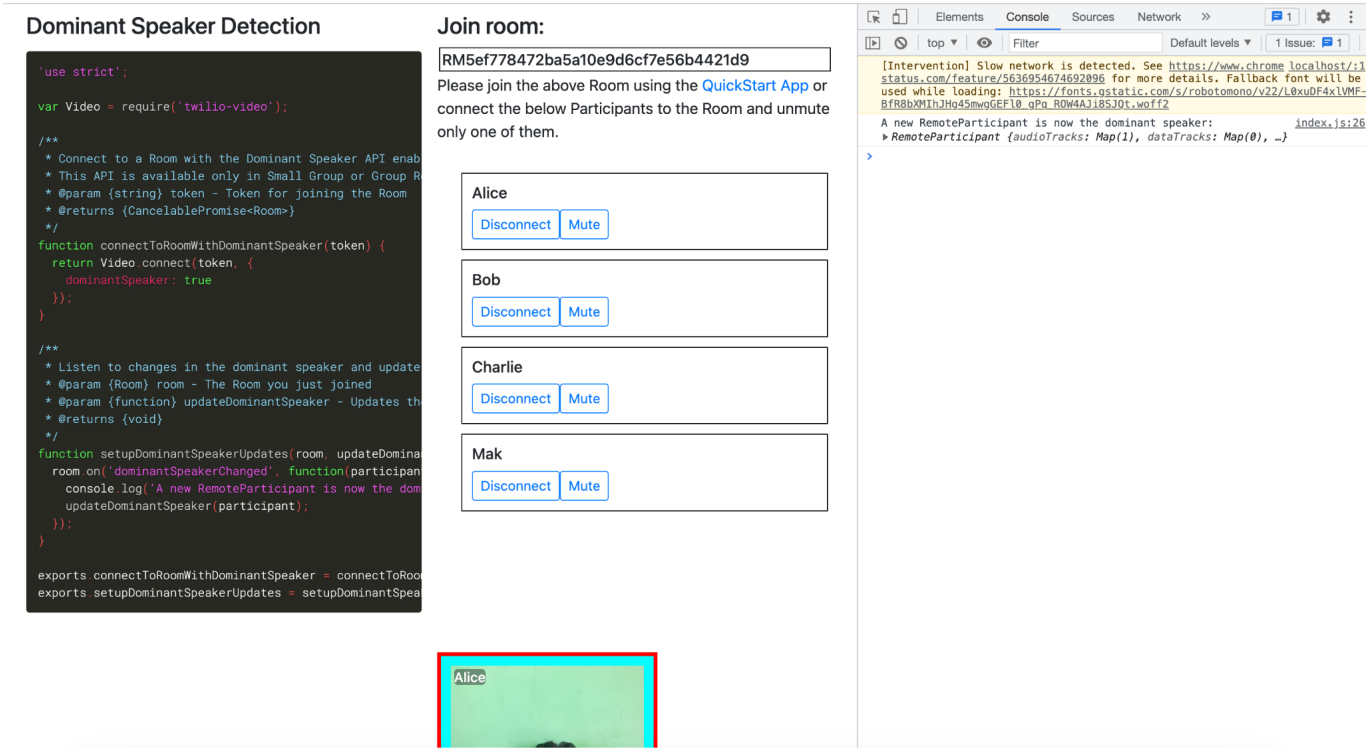


Screenshot #4



Example - Dominant Speaker Detection

Screenshot #1



Screenshot #2

Dominant Speaker Detection

```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with the Dominant Speaker API enab
 * This API is available only in Small Group or Group R
 * @param {string} token - Token for joining the Room
 * @returns {CancelablePromise<Room>}
 */
function connectToRoomWithDominantSpeaker(token) {
  return Video.connect(token, {
    dominantSpeaker: true
  });
}

/**
 * Listen to changes in the dominant speaker and update
 * @param {Room} room - The Room you just joined
 * @param {function} updateDominantSpeaker - Updates th
 * @returns {void}
 */
function setupDominantSpeakerUpdates(room, updateDomina
  room.on('dominantSpeakerChanged', function(participan
    console.log('A new RemoteParticipant is now the dom
    updateDominantSpeaker(participant);
  });
}

exports.connectToRoomWithDominantSpeaker = connectToRoo
exports.setupDominantSpeakerUpdates = setupDominantSpea
```

Join room:

RM663b89d0b22a5307597f4083b80c3800

Please join the above Room using the QuickStart App or connect the below Participants to the Room and unmute only one of them.

Bob	Charlie
Connect	Connect
Alice	**Mak**
Connect	Connect

Screenshot #3

Dominant Speaker Detection

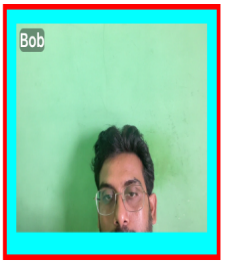
```
'use strict';

var Video = require('twilio-video');

/**
 * Connect to a Room with the Dominant Speaker API enab
 * This API is available only in Small Group or Group R
 * @param {string} token - Token for joining the Room
 * @returns {CancelablePromise<Room>}
 */
function connectToRoomWithDominantSpeaker(token) {
  return Video.connect(token, {
    dominantSpeaker: true
  });
}

/**
 * Listen to changes in the dominant speaker and update
 * @param {Room} room - The Room you just joined
 * @param {function} updateDominantSpeaker - Updates th
 * @returns {void}
 */
function setupDominantSpeakerUpdates(room, updateDomina
  room.on('dominantSpeakerChanged', function(participan
    console.log('A new RemoteParticipant is now the dom
    updateDominantSpeaker(participant);
  });
}

exports.connectToRoomWithDominantSpeaker = connectToRoo
exports.setupDominantSpeakerUpdates = setupDominantSpea
```

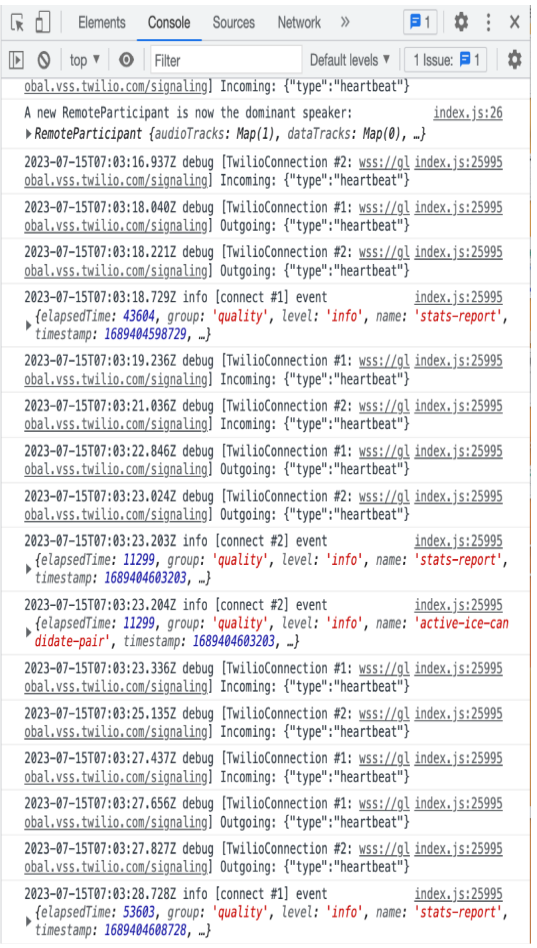
Join room:

RM663b89d0b22a5307597f4083b80c3800

Please join the above Room using the QuickStart App or connect the below Participants to the Room and unmute only one of them.

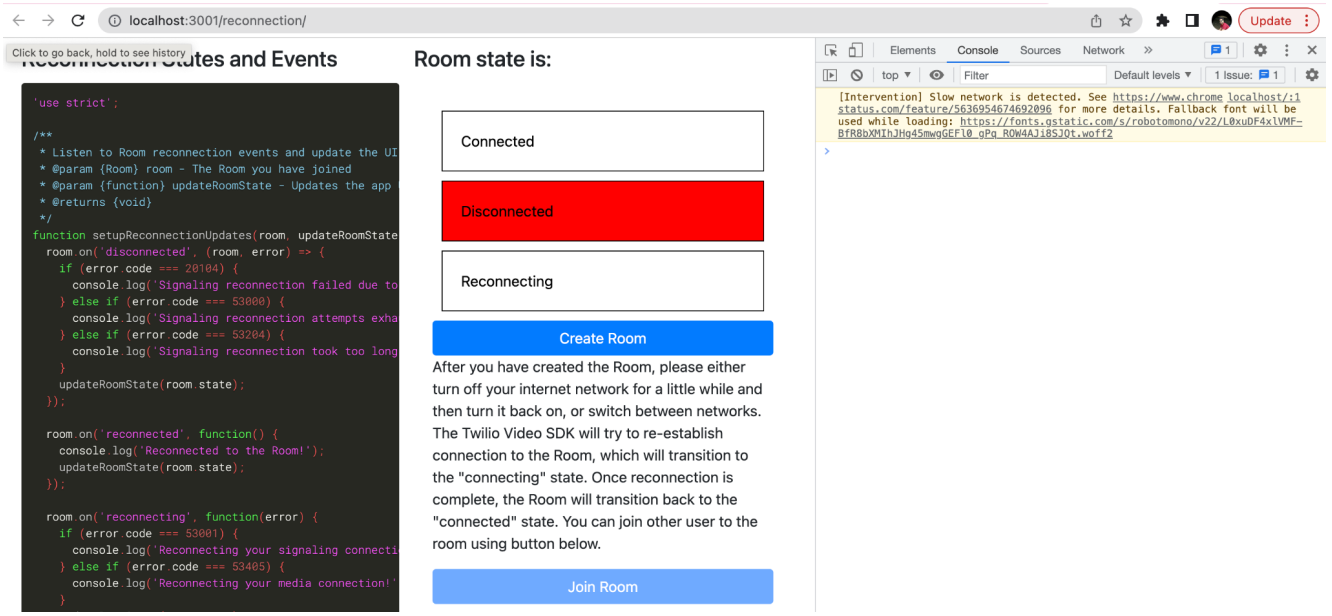
Bob	Mak
Disconnect Mute	Connect
Alice	
Connect	
Charlie	
Connect	



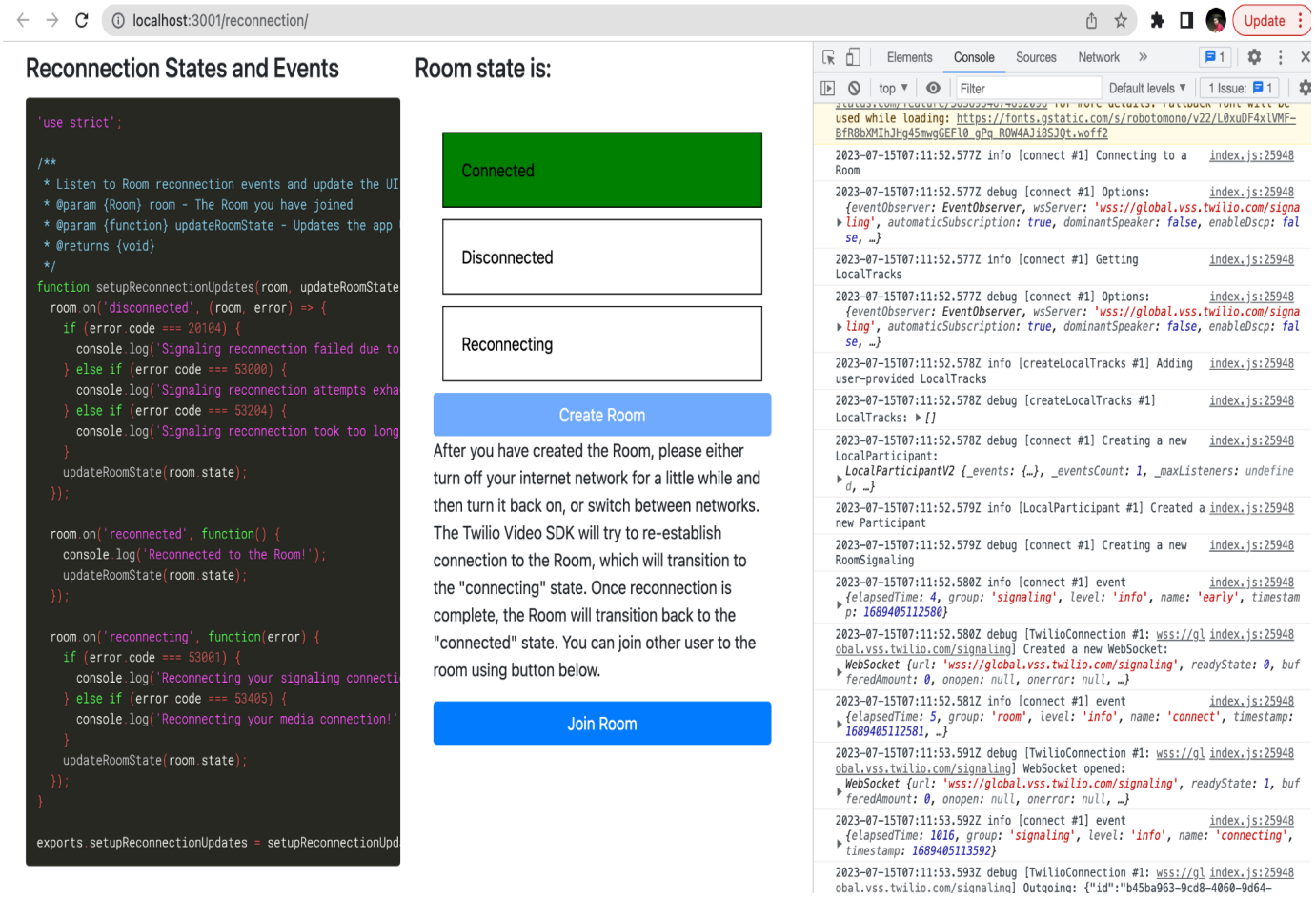


Example - Reconnection States and Events

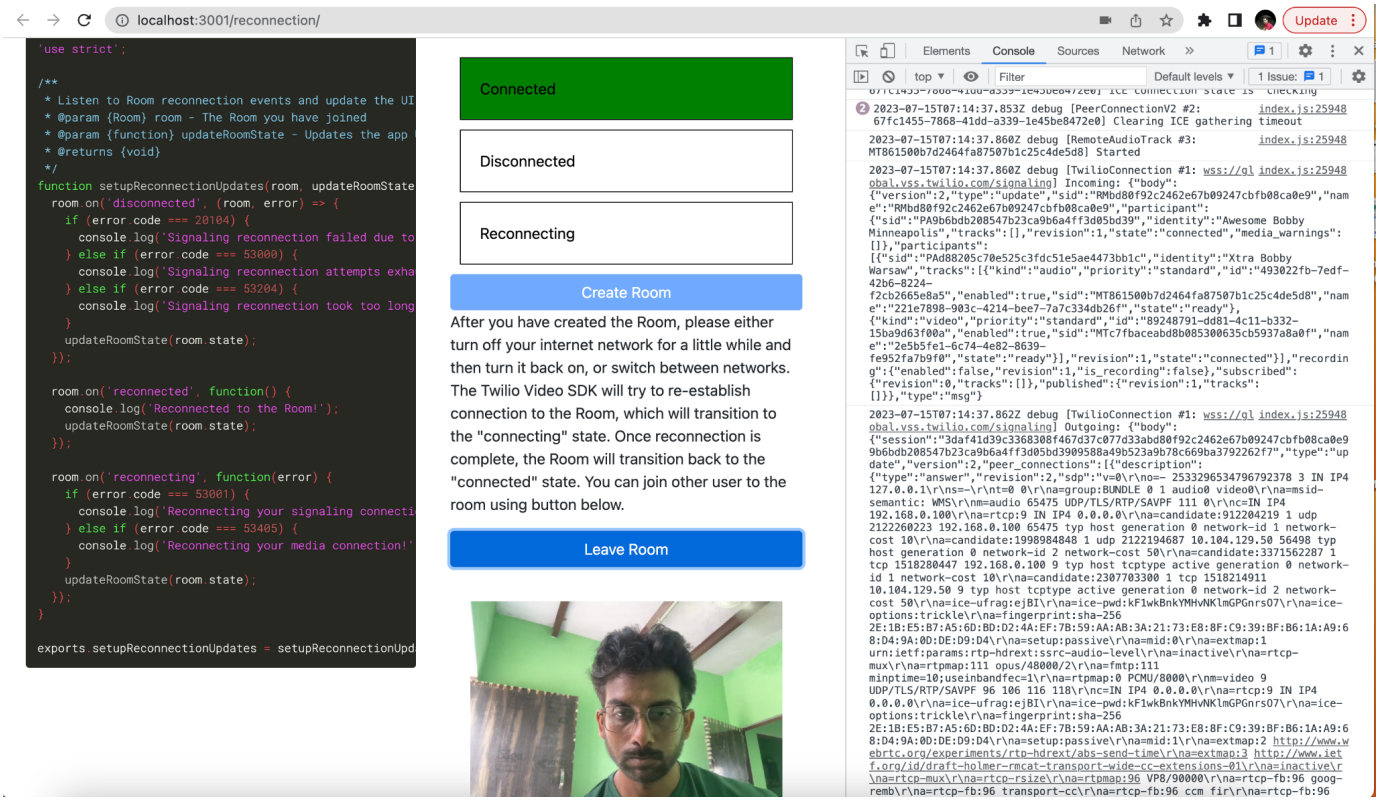
Screenshot #1



Screenshot #2

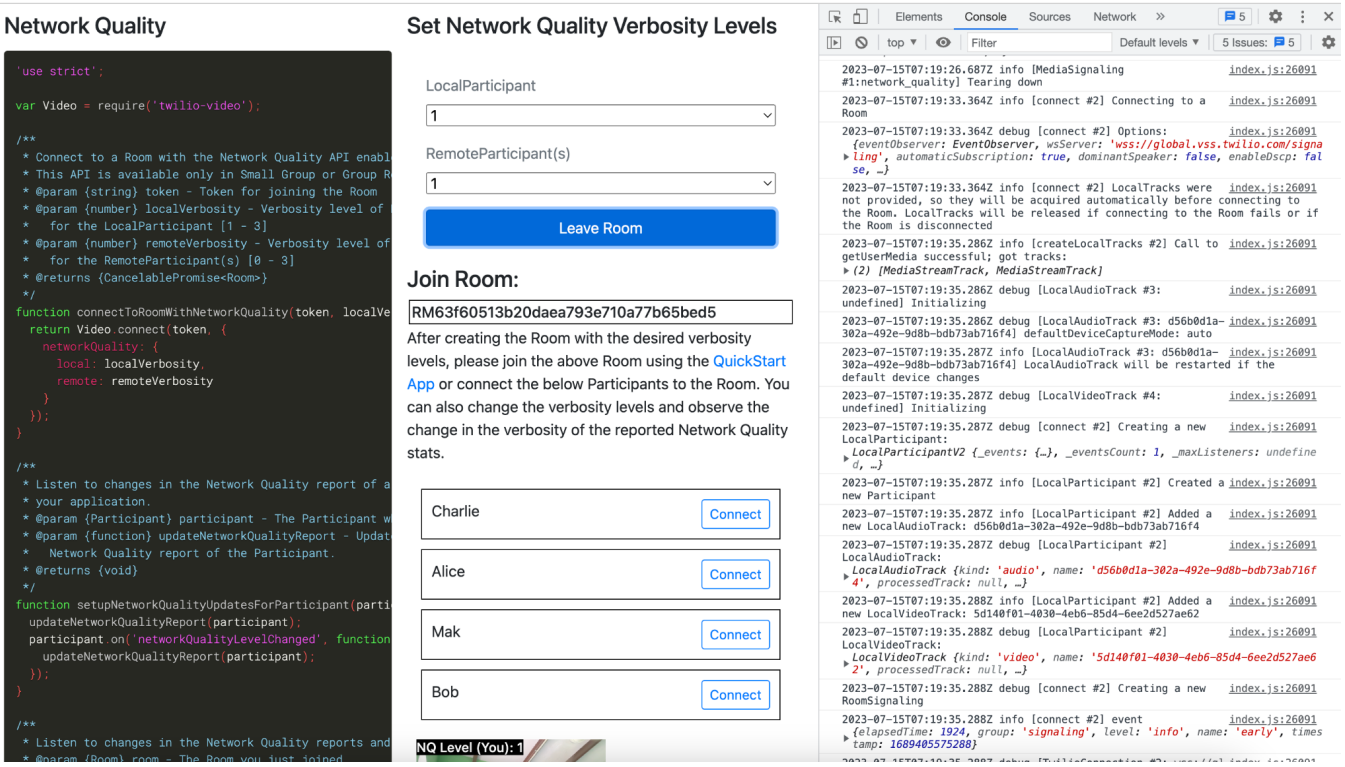


Screenshot #3

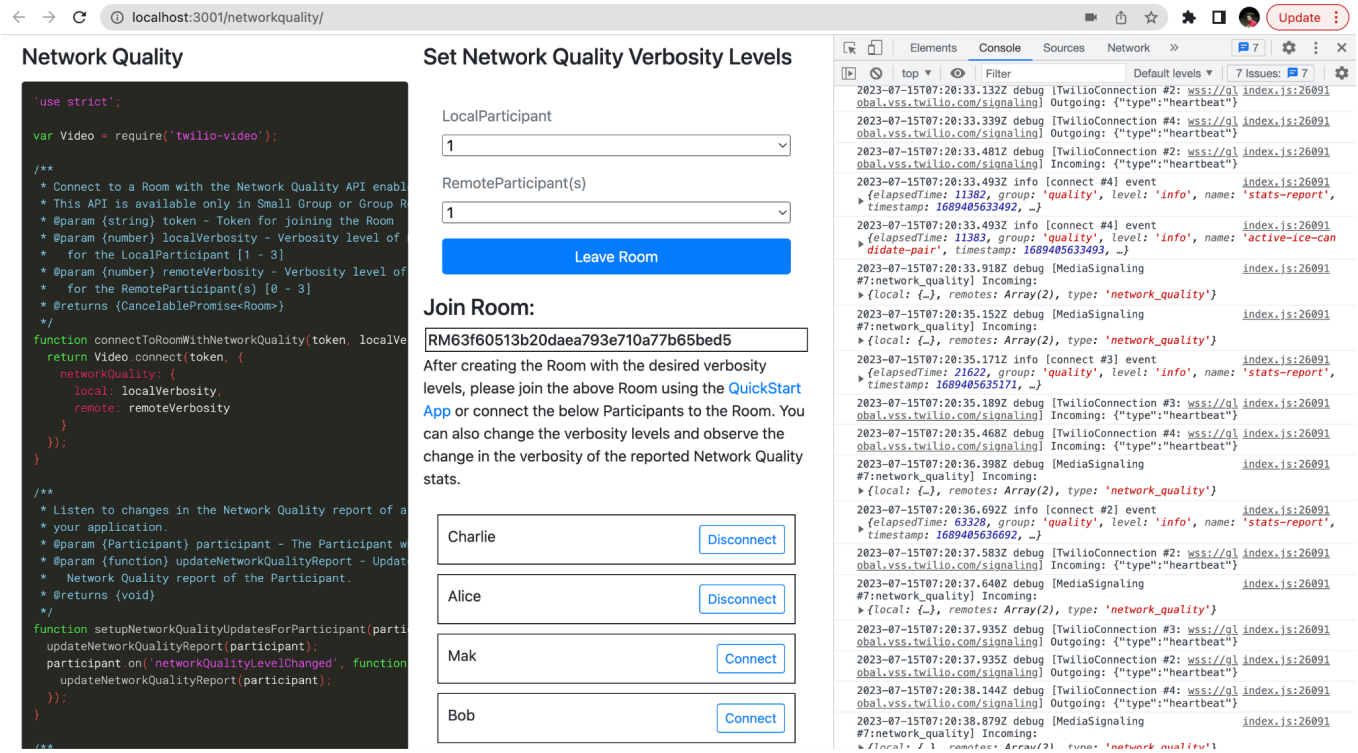


Example - Network Quality

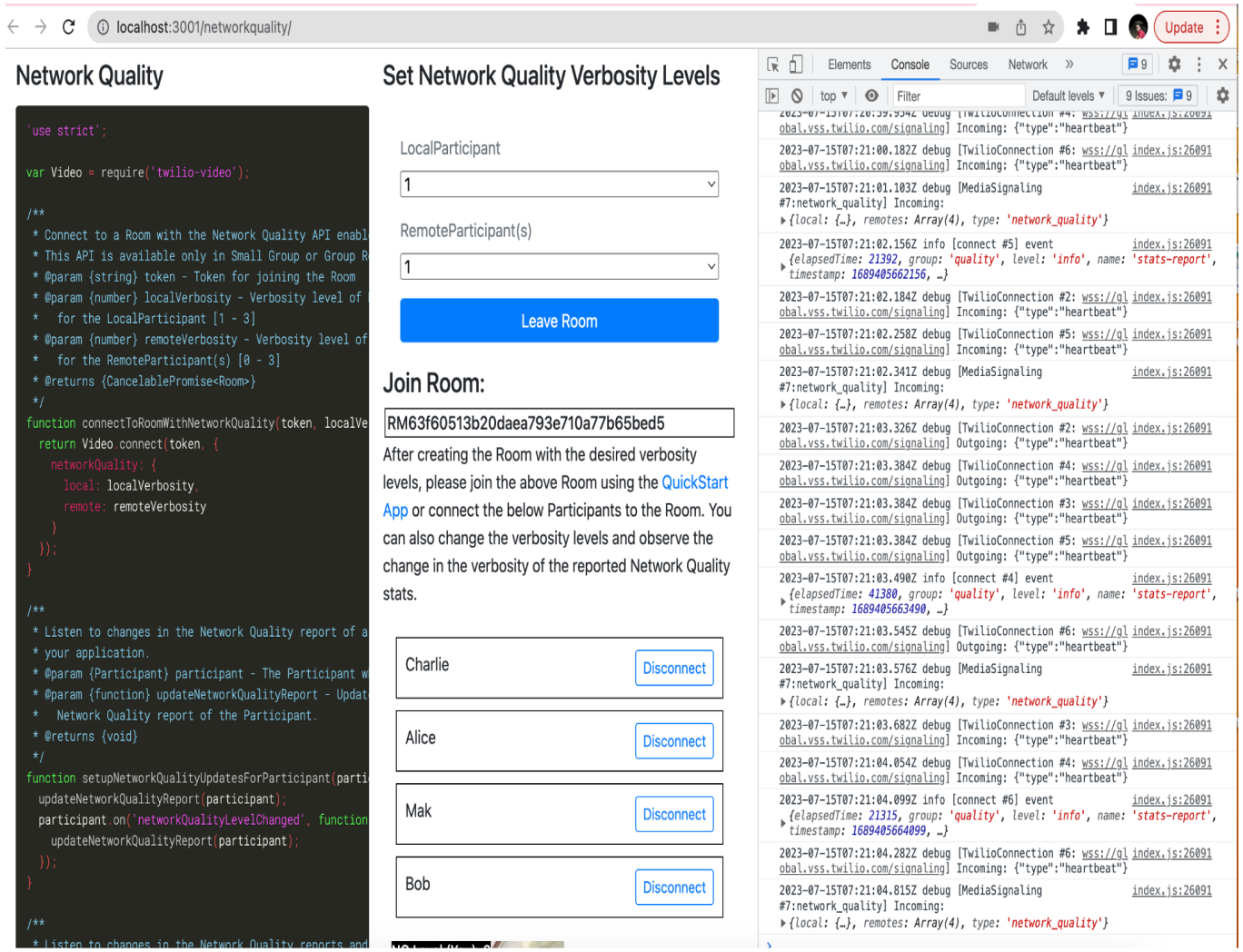
Screenshot #1



Screenshot #2

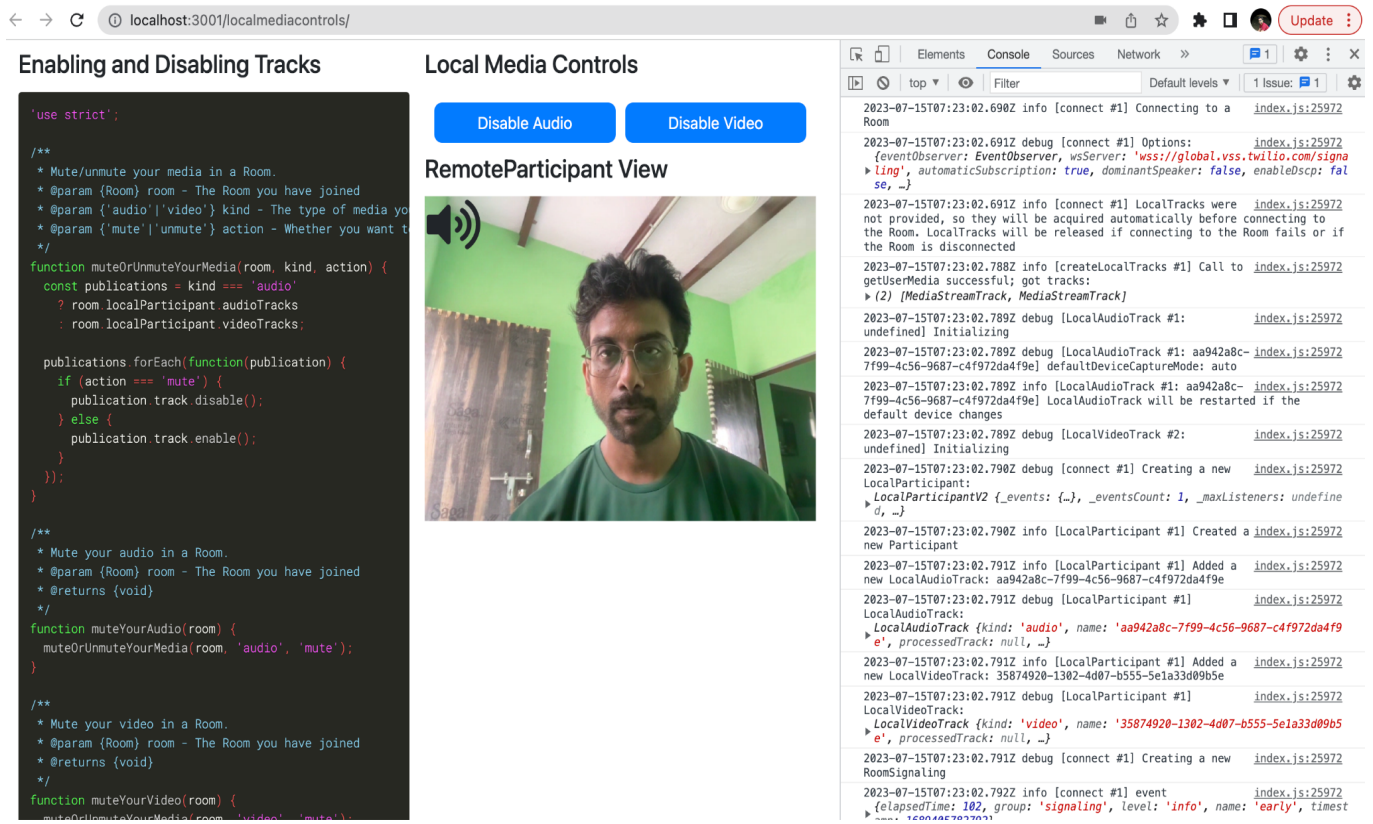


Screenshot #3

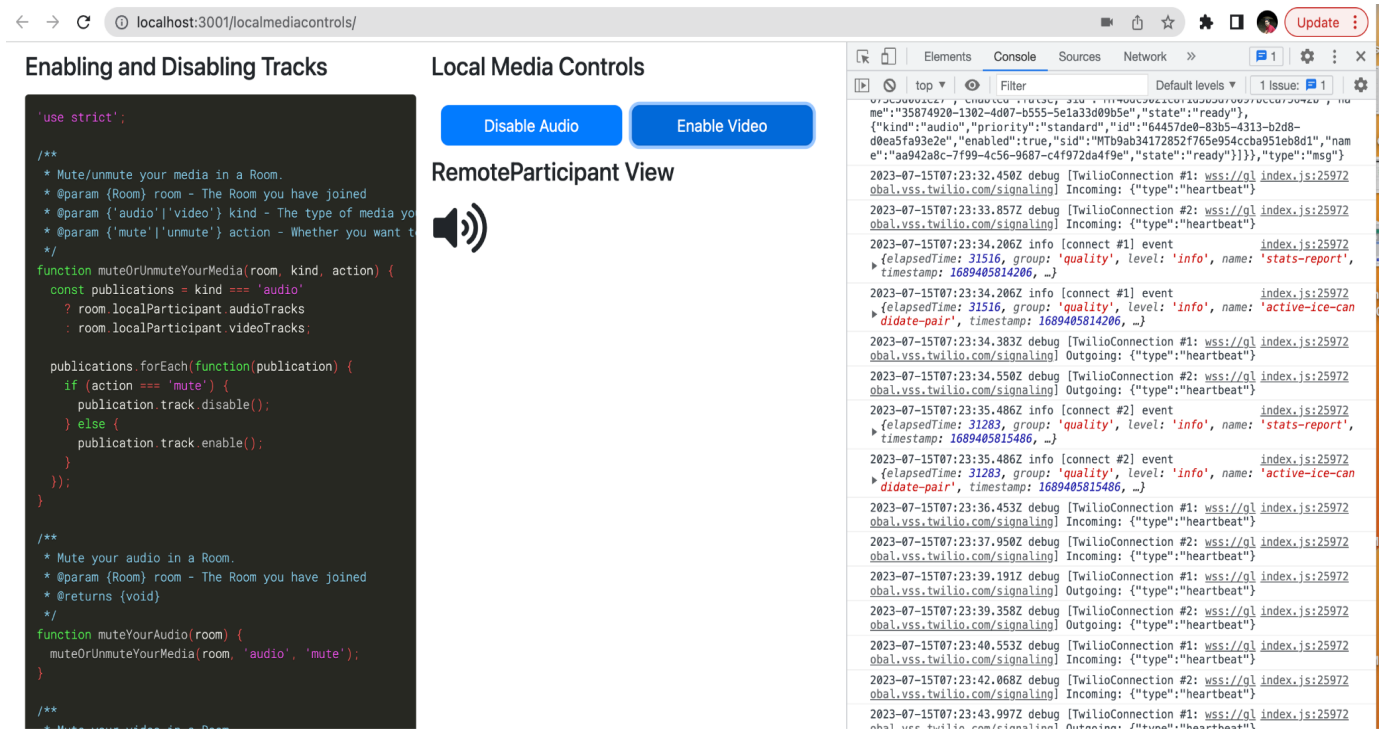


Example - Enabling and Disabling Tracks

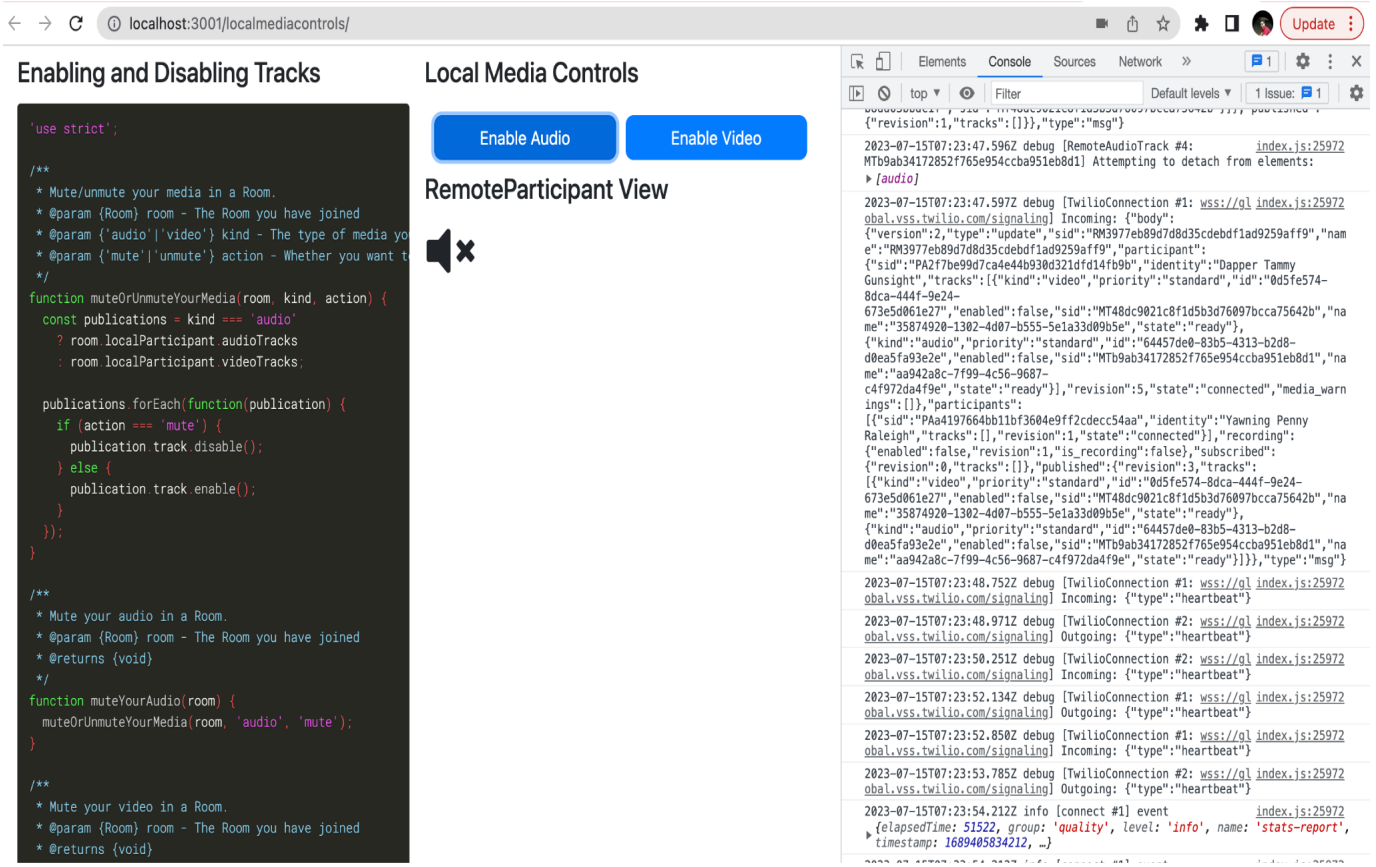
Screenshot #1



Screenshot #2

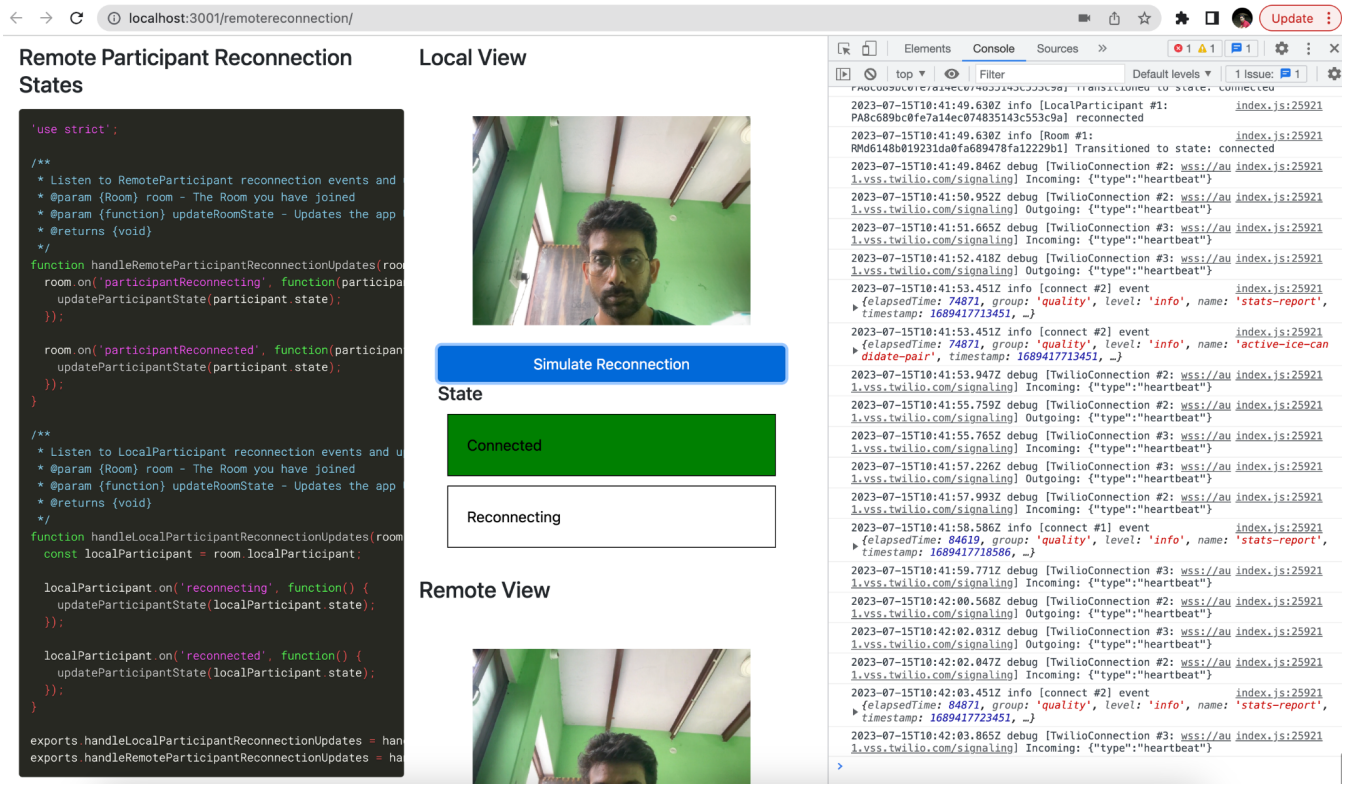


Screenshot #3



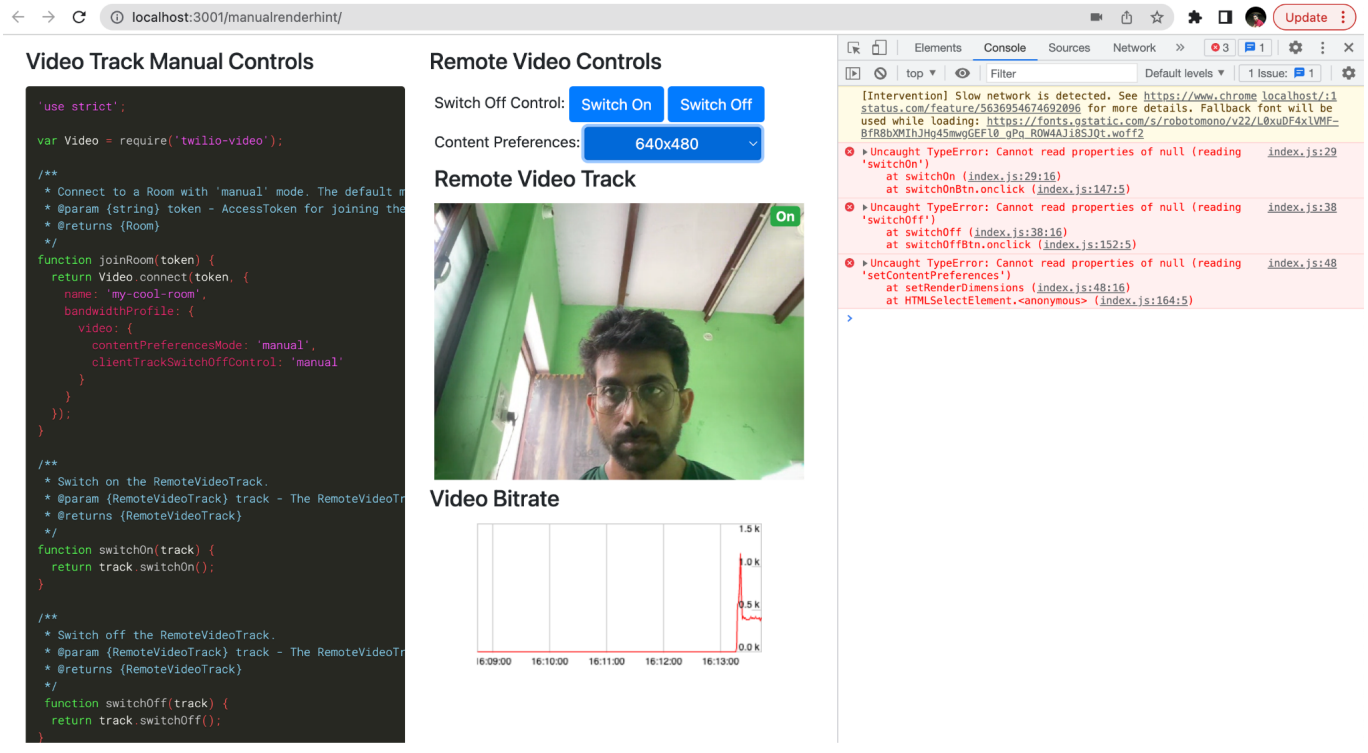
Example - Remote Participant Reconnection States

Screenshot #1



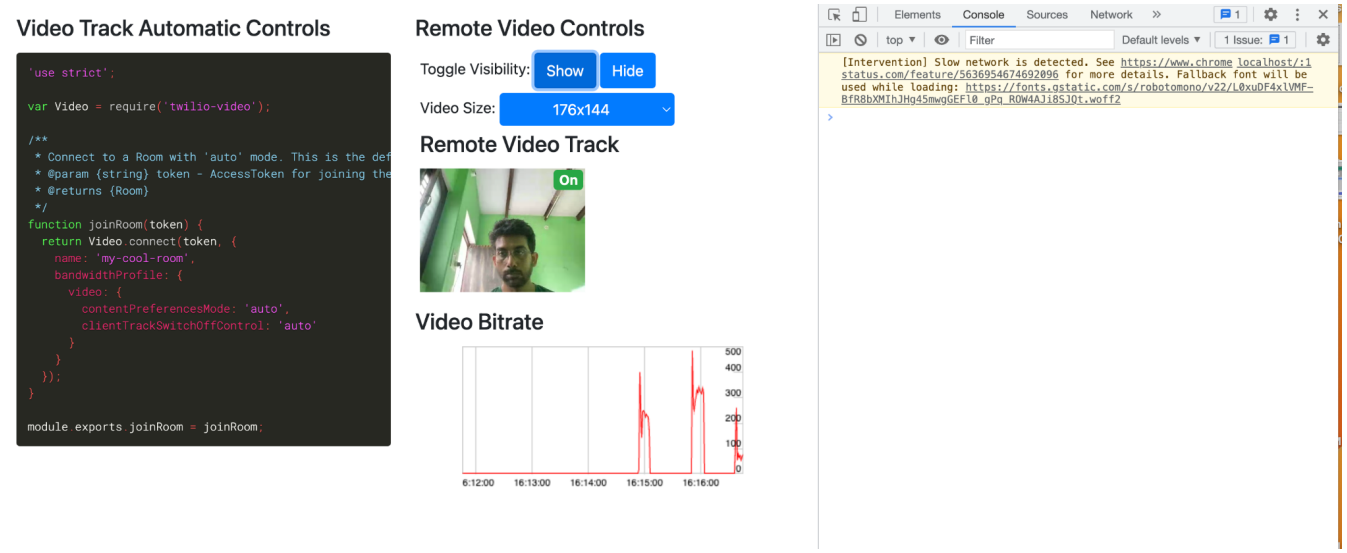
Example - Video Track Manual Controls

Screenshot #1

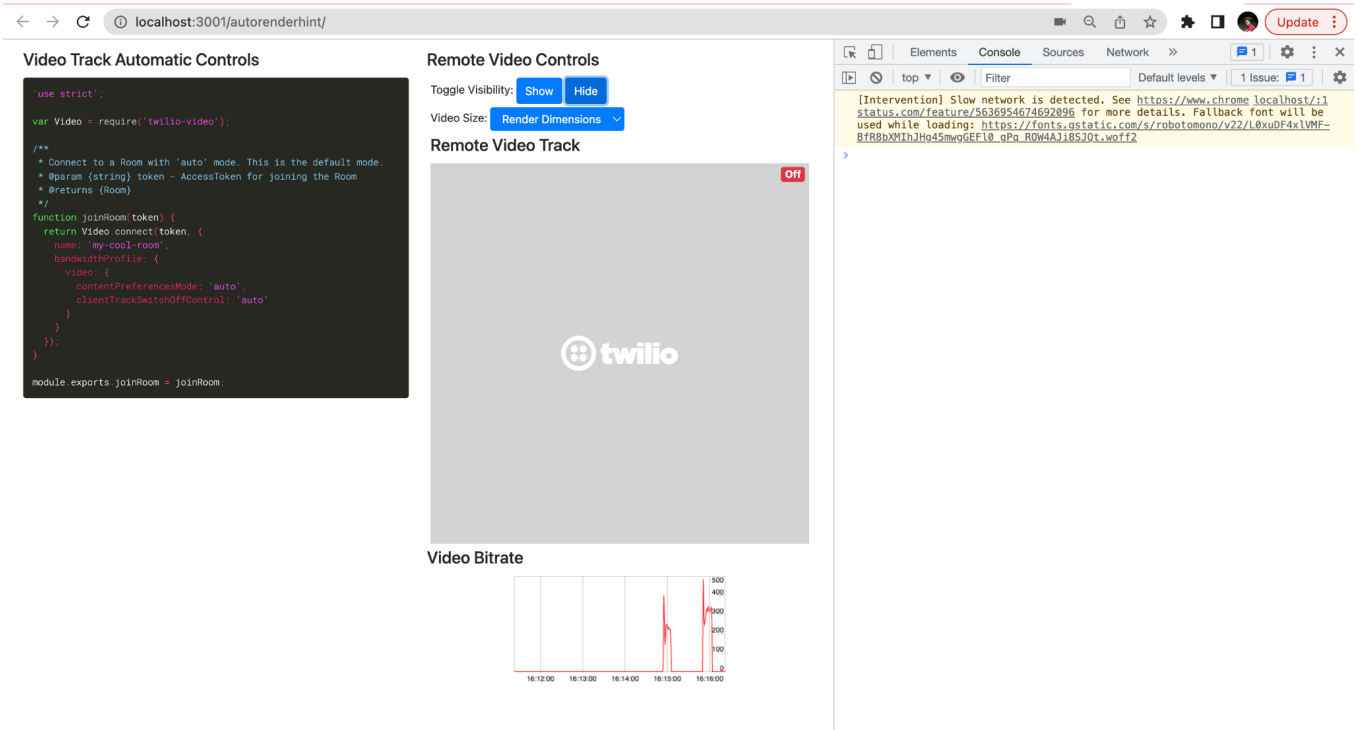


Example - Video Track Manual Controls

Screenshot #1



Screenshot #2



Screenshot #3