

به نام خدا

فاز ۲ - مینی پروژه درس مهندسی اینترنت

بردیا حیدری نژاد - حسین خلیلی - عرفان قالیباف

در فاز قبل یک سیستم ویدیو چت ساده بر اساس پروتکل **webRTC** ایجاد کردیم که عملیات **signaling** به وسیله درخواست های **XHR** و **SSE** انجام می شد. در این فاز هدف جایگزینی **XHR** و **SSE** با **WebSocket** بود تا یک ارتباط **full-duplex** ایجاد شود.

ساختار پروژه به صورت زیر است .

```
root
|
|
index.html
package.json
server.js
|_js
|   |
|   main.js
|_assets
|   |
|   |_styles
|       |
|       main.css
|   |_scss
|       |
|       main.scss
```

در فایل **index.html** ظاهر سایت طراحی شده است. که شامل یک فریم ویدیو برای نمایش ویدیو خود فرد، یک فریم ویدیو برای نمایش تصویر دریافتی، دکمه های مربوط به ایجاد و ورود به یک **room** و دکمه قطع تماس می باشد.

```

<!DOCTYPE html>
<html>
  <head>
    <title>WebRTC Simple Video-Chat</title>
    <link rel="stylesheet" type="text/css" href="assets/styles/main.css">
  </head>

  <body>
    <div class="container">
      <div class="sidebar">
        <div style="margin: auto;text-align: center;">
          <button type="button" onclick="createOrJoin()" id="createbtn">Create Room
        </button>
          <button type="button" onclick="createOrJoin()" id="joinbtn">Join Room</
            button>
        </div>

      </div>
      <input id="endCallButton" onclick="hangup()" type="button" disabled value="END
        "/>
      <div class="main">

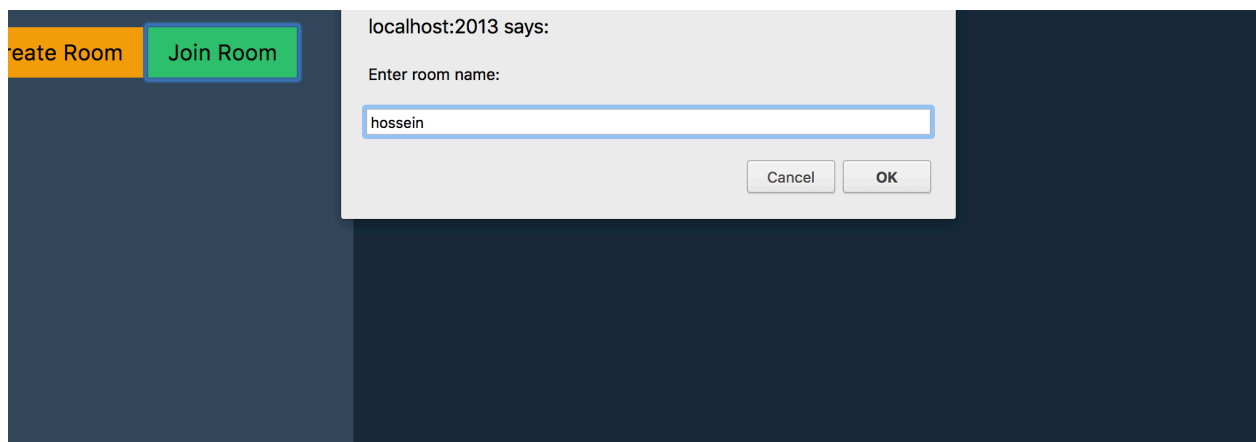
        <video id="localVideo" autoplay muted></video>
        <video id="remoteVideo" autoplay></video>

      </div>

    </div>
    <script src='/socket.io/socket.io.js'></script>
    <script src='js/lib/adapter.js'></script>
    <script src='js/main.js'></script>

```

با کلیک بر روی دکمه create room یک پنجره popUp باز میشود و از شما میخواهد که نام room را وارد کنید.



کدهای مربوط به این کارکرد در تابع createOrJoin در فایل main.js نوشته شده اند.

```

var room = prompt("Enter room name:");
if (room !== "") {
  socket.emit('creatorjoin', room);
}

```

همانطور که مشاهده می‌شود پس از زدن دکمه **ok** توسط کاربر **client** یک پیغام مبنی بر ایجاد یا اضافه شدن به یک روم را از طریق سوکت ایجاد شده ارسال می‌کند. سوکت مورد نظر در ابتدای این فایل و با استفاده از دستور

```
var socket = io.connect();
```

ایجاد و متصل شده است.

سپس نوبت **client** دیگر است که به روم اضافه شود. برای این کار باید روی دکمه **join** کلیک کرده و نام **room** را وارد کند. سپس دستورات مربوط مانند حالت ایجاد یک روم اجرا می‌شوند. اما چون این **room** وجود دارد در نتیجه کاربر به آن اضافه می‌شود. دستورات مربوط به کنترل این موضوع در فایل **server.js** قرار دارند و به شرح زیر است.

```
socket.on('creatorjoin', function (room) {  
    var numClients = io.sockets.clients(room).length;  
  
    log('Room ' + room + ' has ' + numClients + ' client(s)');  
    log('Request to create or join room', room);  
  
    if (numClients == 0){  
        s_room = room;  
        log("First Peer Requested to Create Room: " + room);  
        socket.join(room);  
        socket.emit('created', room);  
    } else if (numClients == 1 && s_room == room) {  
        log("Second Peer Requested to Join Room: " + room);  
        io.sockets.in(room).emit('join', room);  
        socket.join(room);  
        socket.emit('joined', room);  
    } else { // max two clients or not right room name  
        log("Request to Join Room is ignored!");  
        socket.emit('ignore', room);  
    }  
  
});
```

ابتدا سرور لیست تمام **client** های **room** را جمع آوری می‌کند. سپس اگر تعداد این **client** ها صفر بود به این معنی است که باید یک روم جدید ایجاد شود. در غیر این صورت باید کاربر به روم اضافه شود. نتیجه با استفاده از دستور **emit** به تمام کلاینت ها اعلام می‌شود.

بقیه دستورات signaling توسط تابع

```
socket.on('message', function (message) {  
    log('Got message: ', message);  
    // For a real app, should be room only (not broadcast)  
    socket.broadcast.emit('message', message);  
});
```

در فایل **server.js** به کلاینت های مربوط به یک **room** ارسال می گردد.
در سمت کلاینت دستورات انتظار دریافت دستورات زیر را از سرور داریم.

socket.on('created') روم ایجاد شده

socket.on('join') یک نفر تقاضای اضافه شدن به روم را داده است

socket.on('ignore') فرد نتوانسته به روم اضافه شود

socket.on('joined') فرد جدیدی به روم اضافه شده است.

پس از همه این کار ها اتصال برقرار شده و تصویر مانند شکل زیر است.

