Xây dựng ứng dụng gọi video và chat realtime bằng Nodejs + Socket.io + WebRTC

**Cài đặt**

1. Tạo một thư mục và đặt tên video-chat

2. Mở màn hình console (cmd, Terminal) ở tại thư mục và thực hiện npm init

3. Thực hiện cấu hình thông số cho ứng dụng

4. Tiến hành chạy lệnh npm install express ejs socket.io uuid peer

- express: một web framework của Nodejs

- ejs: view engine phổ biến của Nodejs.

- socket.io: Socket.IO is a library that enables real-time, bidirectional and event-based communication between the browser and the server

- uuid: thư viện giúp tạo uuid

- peer: thư viện tạo WebRTC trong Nodejs

5. Tiếp tục thực hiện lệnh npm install nodemon –save-dev

- nodemon: thư viện giúp reload server Nodejs khi có thay đổi code

Cấu hình file package.json

"scripts": {

    "start": "nodemon server.js"

  }

6. Tạo một file server.js ở thư mục gốc

**Tạo server với Expressjs**

const express = require('express');

const http = require('http');

// định nghĩa port của ứng dụng

const PORT = process.env.PORT || 3000;

// khởi tạo app

const app = express();

const server = http.createServer(app);

/\*\*

 \* GET: /

 \*/

app.get("/", (req, res) => {

    res.send(“Hello World”);

});

// lắng nghe tại địa chỉ có port {PORT}

server.listen(PORT);

Chạy lệnh npm run start và truy cập địa chỉ: [http://localhost:3000](http://localhost:3000/)trên trình duyệt

**Tạo giao diện cho trang chat video-chat**

Đầu tiên cần cấu hình view engine cho express

// thiết lập view engine cho ứng dụng

app.set('view engine', 'ejs');

Tiếp đến tạo một file views/rooms.ejs với nội dung

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Video Chat</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="header">

        <div class="logo">

            <h3>Video Chat</h2>

        </div>

    </div>

    <div class="main">

        <div class="main\_\_left">

            <div class="videos\_\_group">

                <div id="video-grid">

                </div>

            </div>

            <div class="options">

                <div class="options\_\_left">

                    <div id="stopVideo" class="options\_\_button">

                        <i class="fa fa-video-camera"></i>

                    </div>

                    <div id="muteButton" class="options\_\_button">

                        <i class="fa fa-microphone"></i>

                    </div>

                </div>

                <div class="options\_\_right">

                    <div id="inviteButton" class="options\_\_button">

                        <i class="fas fa-user-plus"></i>

                    </div>

                </div>

            </div>

        </div>

        <div class="main\_\_right">

            <div class="main\_\_chat\_window">

                <div class="messages">

                </div>

            </div>

            <div class="main\_\_message\_container">

                <input id="chat\_message" type="text" autocomplete="off" placeholder="Type message here...">

                <div id="send" class="options\_\_button">

                    <i class="fa fa-plus" aria-hidden="true"></i>

                </div>

            </div>

        </div>

    </div>

    <script src="/socket.io/socket.io.js"></script>

    <script src="https://kit.fontawesome.com/c939d0e917.js"></script>

    <script src="https://unpkg.com/peerjs@1.3.1/dist/peerjs.min.js"></script>

</body>

</html>

Quay lại file server.js

/\*\*

 \* GET: /

 \*/

app.get("/", (req, res) => {

    res.render('room', { roomId });

});

**Thêm style cho giao diện**

Tạo file public/style.css với nội dung

/\* import font  \*/

@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap");

/\*

 \* Định nghĩa các biến global trong css

\*/

:root {

    --main-darklg: #1d2635;

    --main-dark: #161d29;

    --primary-color: #2f80ec;

    --main-light: #eeeeee;

    font-family: "Poppins", sans-serif;

}

/\*

 \* Style chung

\*/

\* {

    margin: 0;

    padding: 0;

}

/\*

 \* div.header

\*/

.header {

    display: flex;

    justify-content: center;

    align-items: center;

    height: 8vh;

    position: relative;

    width: 100%;

    background-color: var(--main-darklg);

}

.logo > h3 {

    color: var(--main-light);

}

.main {

    overflow: hidden;

    height: 92vh;

    display: flex;

}

.main\_\_left {

    flex: 0.7;

    display: flex;

    flex-direction: column;

}

.videos\_\_group {

    flex-grow: 1;

    display: flex;

    justify-content: center;

    align-items: center;

    padding: 1rem;

    background-color: var(--main-dark);

}

video {

    height: 300px;

    border-radius: 1rem;

    margin: 0.5rem;

    width: 400px;

    object-fit: cover;

    transform: rotateY(180deg);

    -webkit-transform: rotateY(180deg);

    -moz-transform: rotateY(180deg);

}

.options {

    padding: 1rem;

    display: flex;

    background-color: var(--main-darklg);

}

.options\_\_left {

    display: flex;

}

.options\_\_right {

    margin-left: auto;

}

.options\_\_button {

    display: flex;

    justify-content: center;

    align-items: center;

    background-color: var(--primary-color);

    height: 50px;

    border-radius: 5px;

    color: var(--main-light);

    font-size: 1.2rem;

    width: 50px;

    margin: 0 0.5rem;

}

.background\_\_red {

    background-color: #f6484a;

}

.main\_\_right {

    display: flex;

    flex-direction: column;

    flex: 0.3;

    background-color: #242f41;

}

.main\_\_chat\_window {

    flex-grow: 1;

    overflow-y: scroll;

}

.main\_\_chat\_window::-webkit-scrollbar {

    display: none;

}

.main\_\_message\_container {

    padding: 1rem;

    display: flex;

    align-items: center;

    justify-content: center;

}

.main\_\_message\_container > input {

    height: 50px;

    flex: 1;

    font-size: 1rem;

    border-radius: 5px;

    padding-left: 20px;

    border: none;

}

.messages {

    display: flex;

    flex-direction: column;

    margin: 1.5rem;

}

.message {

    display: flex;

    flex-direction: column;

}

.message > b {

    color: #eeeeee;

    display: flex;

    align-items: center;

    text-transform: capitalize;

}

.message > b > i {

    margin-right: 0.7rem;

    font-size: 1.5rem;

}

.message > span {

    background-color: #eeeeee;

    margin: 1rem 0;

    padding: 1rem;

    border-radius: 5px;

}

#video-grid {

    display: flex;

    justify-content: center;

    flex-wrap: wrap;

}

#showChat {

    display: none;

}

.header\_\_back {

    display: none;

    position: absolute;

    font-size: 1.3rem;

    top: 17px;

    left: 28px;

    color: #fff;

}

@media (max-width: 700px) {

    .main\_\_right {

        display: none;

    }

    .main\_\_left {

        width: 100%;

        flex: 1;

    }

    video {

        height: auto;

        width: 100%;

    }

    #showChat {

        display: flex;

    }

}

**Thiết lập cho Room**

Import thư viện uuid

const { v4: uuidv4 } = require('uuid');

và sửa lại router cho ứng dụng

/\*\*

 \* GET: /

 \* Thực hiện redirect về trang /{00000000-0000-0000-0000-000000000000}

 \*/

app.get("/", (req, res) => {

    res.redirect(`/${uuidv4()}`);

});

/\*\*

 \* GET: /:room

 \* Render giao diện video chat

 \*/

app.get('/:room', (req, res) => {

    let roomId = req.params.room;

    res.render('room', { roomId });

});