**Exercise 11 – Chatroom Application**

**Name:** Badri MSV **Roll No:** 195002017

**Subject:** UIT1611 – Web Programming Lab **Faculty:** Dr. S. Sasirekha

**Date:** 07/06/2022

**Aim:**

To create a chat application with multiple rooms using Node and Express.

**Procedure:**

* First initialise the express framework using require() call, and declare a variable app to use the express framework.
* Similarly, instantiate two variables for HTTP server and Socket connection.
* Then, render index.ejs when we got to localhost:8080.
* Now, when the socket rst receives a connection(listening on 8080), it prompts for a username and declares that username is online.
* When the connection is disconnected, the application declares that the user has left the room.
* Then listen on port 8080.

**index.ejs:**

<!DOCTYPE html>

<html>

<head>

    <title>Simple Group Chat on Node.js</title>

    <style>

        \* {

            margin: 0;

            padding: 0;

            box-sizing: border-box;

        }

        body {

            font: 13px Helvetica, Arial;

        }

        form {

            background: #fff;

            padding: 3px;

            position: fixed;

            bottom: 0;

            width: 100%;

            border-color: #000;

            border-top-style: solid;

            border-top-width: 1px;

        }

        form input {

            border-style: solid;

            border-width: 1px;

            padding: 10px;

            width: 85%;

            margin-right: .5%;

        }

        form button {

            width: 9%;

            background: rgb(130, 224, 255);

            border: none;

            padding: 10px;

            margin-left: 2%;

        }

        #messages {

            list-style-type: none;

            margin: 0;

            padding: 0;

        }

        #messages li {

            padding: 5px 10px;

        }

        #messages li:nth-child(odd) {

            background: #eee;

        }

    </style>

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

    <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>

</head>

<body>

    <ul id="messages"></ul>

    <form action="/" method="POST" id="chatForm">

        <input id="txt" autocomplete="off" autofocus="on" oninput="isTyping()"

            placeholder="type your message here..." /><button>Send</button>

    </form>

    <script>

        var socket = io.connect('http://localhost:8080');

        // submit text message without reload/refresh the page

        $('form').submit(function (e) {

            e.preventDefault(); // prevents page reloading

            socket.emit('chat\_message', $('#txt').val());

            $('#txt').val('');

            return false;

        });

        // append the chat text message

        socket.on('chat\_message', function (msg) {

            $('#messages').append($('<li>').html(msg));

        });

        // append text if someone is online

        socket.on('is\_online', function (username) {

            $('#messages').append($('<li>').html(username));

        });

        // ask username

        var username = prompt('Please tell me your name');

        socket.emit('username', username);

    </script>

</body>

</html>

**index.js:**

const express = require('express');

const app = express();

const http = require('http').Server(app);

const io = require('socket.io')(http);

app.get('/', function(req, res) {

    res.render('index.ejs');

});

io.sockets.on('connection', function(socket) {

    socket.on('username', function(username) {

        socket.username = username;

        io.emit('is\_online', '🔵 <i>' + socket.username + ' joined the chat..</i>');

    });

    socket.on('disconnect', function(username) {

        io.emit('is\_online', '🔴 <i>' + socket.username + ' left the chat..</i>');

    })

    socket.on('chat\_message', function(message) {

        io.emit('chat\_message', '<strong>' + socket.username + '</strong>: ' + message);

    });

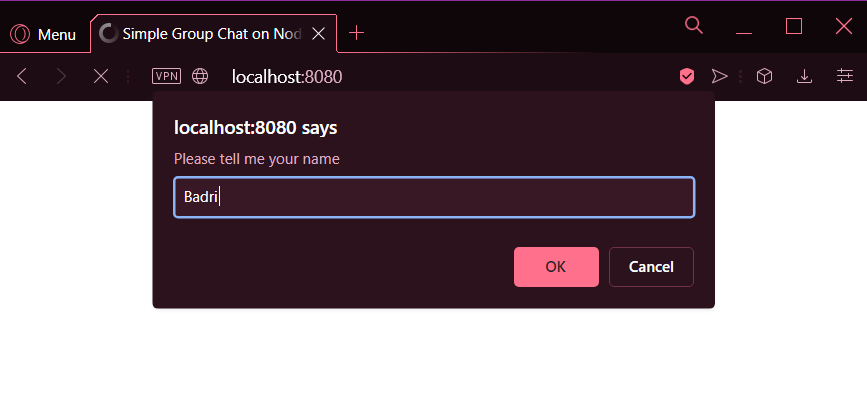
});

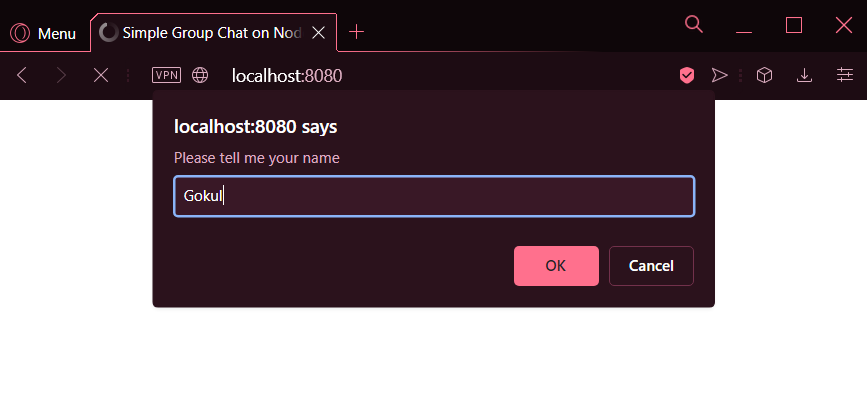
const server = http.listen(8080, function() {

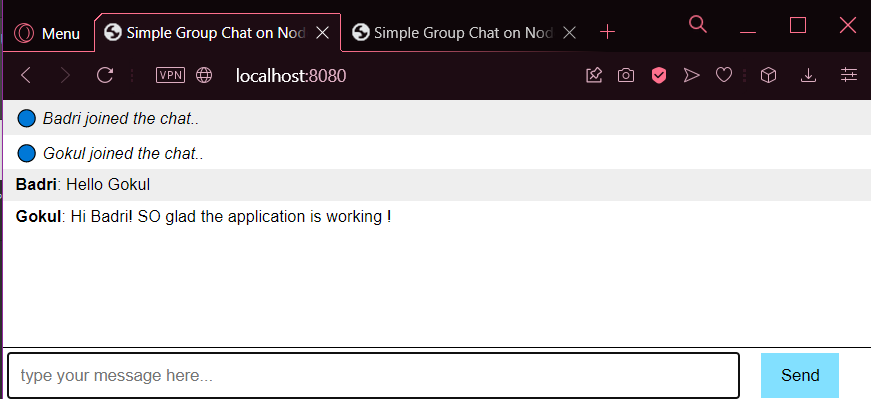
    console.log('listening on \*:8080');

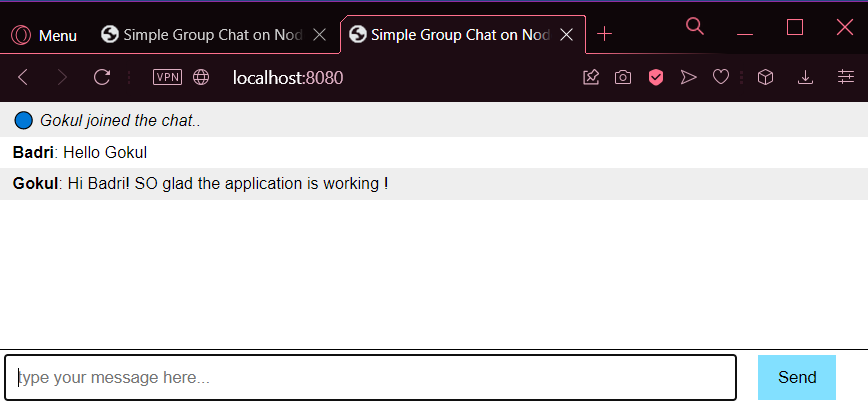
});

**Output:**









**Result:**

Therefore, Chat Application has been implemented using Node JS and Express.