LOVELY PROFESSIONAL UNIVERSITY

Project
On
Real Time Chat Application

Git hub Link--
https://github.com/Sujit11yadav/Real-Time-Chat-Appliction

Submitted by

Sujit Yadav
Reg No.- 11713327
Sec- KM013
Roll no- B59

Submitted to

Neha Sharma

Introduction- Node js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside the browser. The most important advantage of using Node is that we can use JavaScript as both a front-end and back-end language.

As we know, JavaScript was used primarily for client-side scripting, in which scripts were embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser.

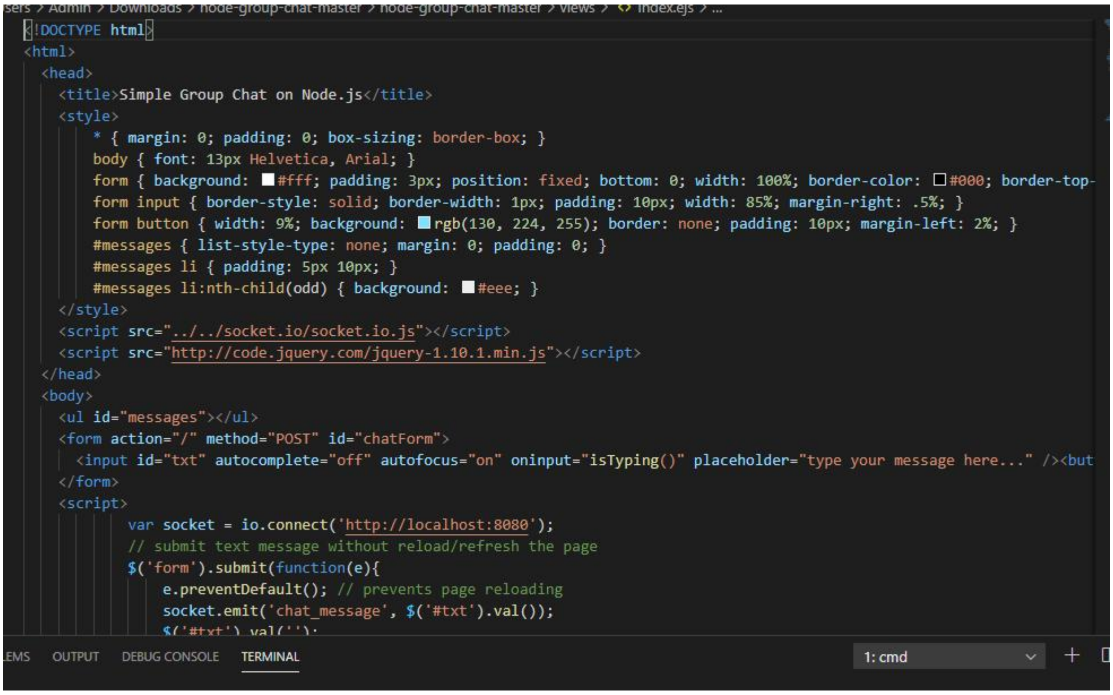
Node.js lets developers use JavaScript to write Command Line tools and for server-side scripting — running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

Real Time Chat Application- Chat, also known as online chat or live chat, is a way of communication and interaction in real-time on the website. This communication is based on text messages but also can have rich media such as video, audio.
Also, real-time chat uses web-based apps that allow communication in a multi-user environment that is usually addressed directly but is anonymous among users.

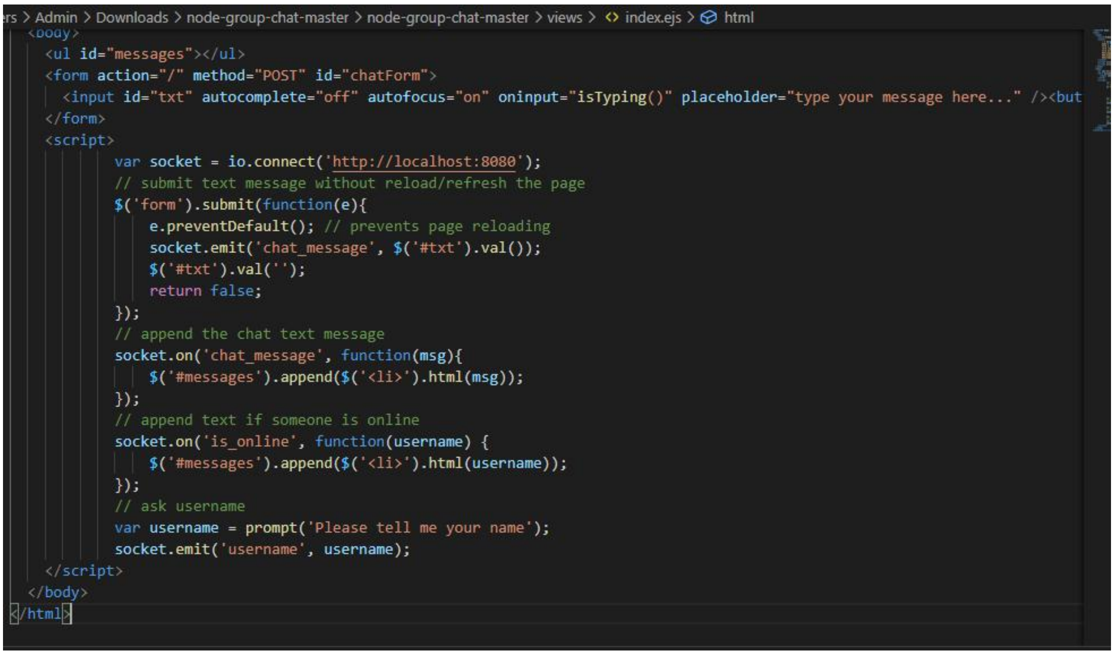
Express.js

Express.js, or simply Express, is a web application framework for Node.js. Express provides a robust set of features for web and mobile applications. Express provides a thin layer of fundamental web application features, without obscuring Node.js features.

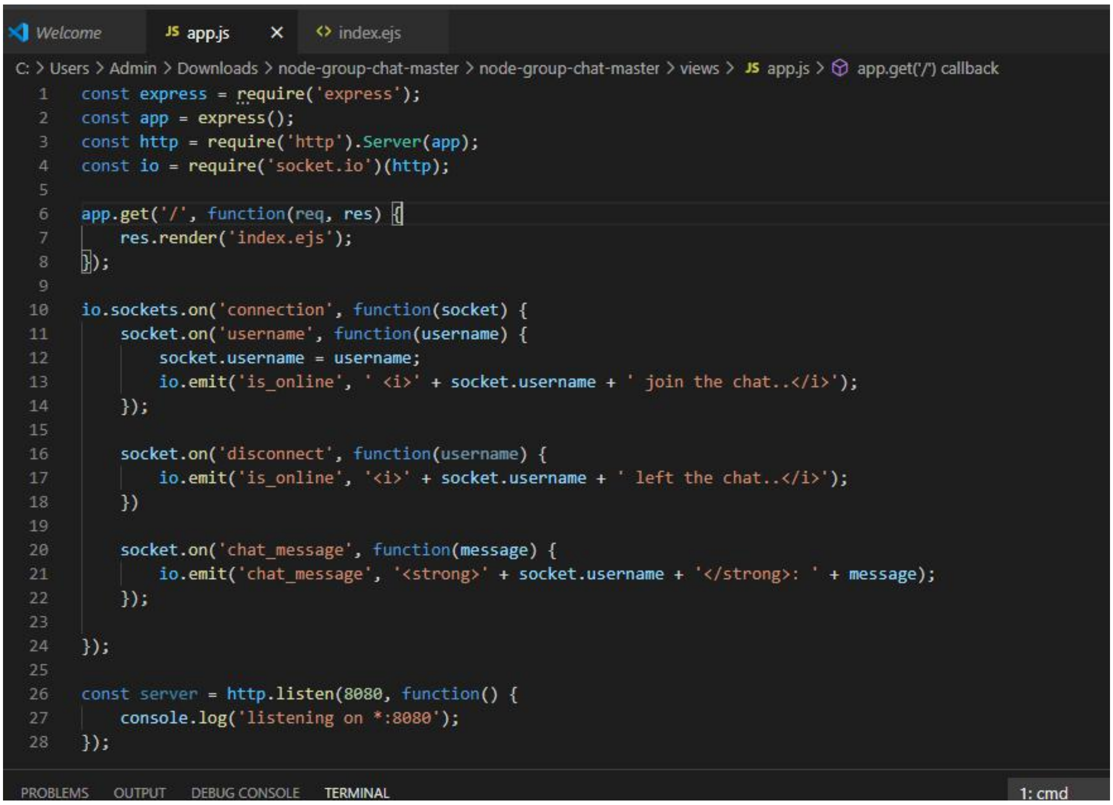
Code Snippets:-


sers > Admin > Downloads > node-group-chat-master > node-group-chat-master > views > index.ejs > ...
EMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: cmd

```
<!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-
        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..." /><but
    </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('');
```

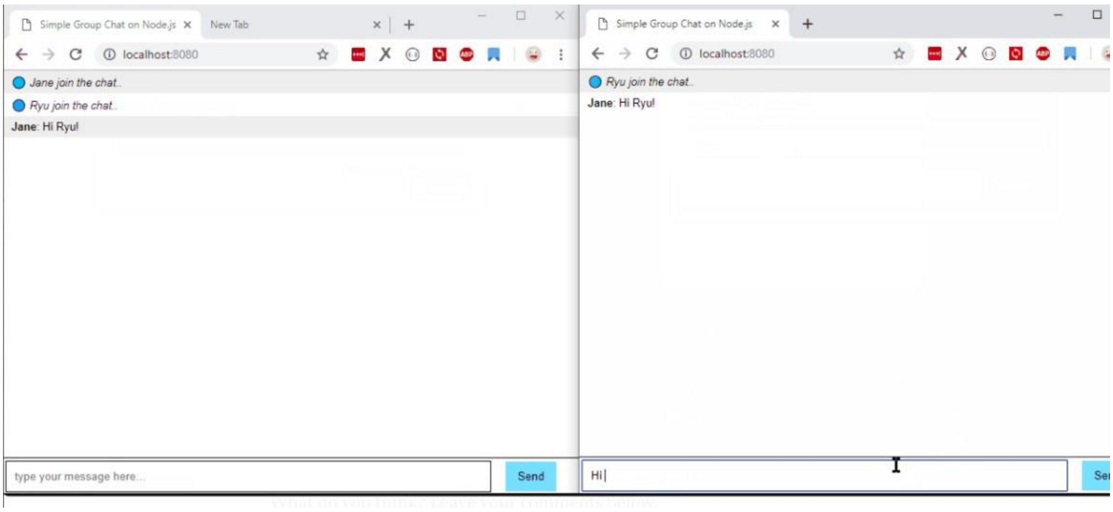

rs > Admin > Downloads > node-group-chat-master > node-group-chat-master > views > index.ejs > html

```
  <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..." /><but
    </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>
```


Welcome
app.js
index.ejs
C: > Users > Admin > Downloads > node-group-chat-master > node-group-chat-master > views > app.js > app.get('/') callback
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: cmd

```javascript
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 + ' join 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');
});
```

<u>Output-</u>



Jane join the chat..

Ryu join the chat..

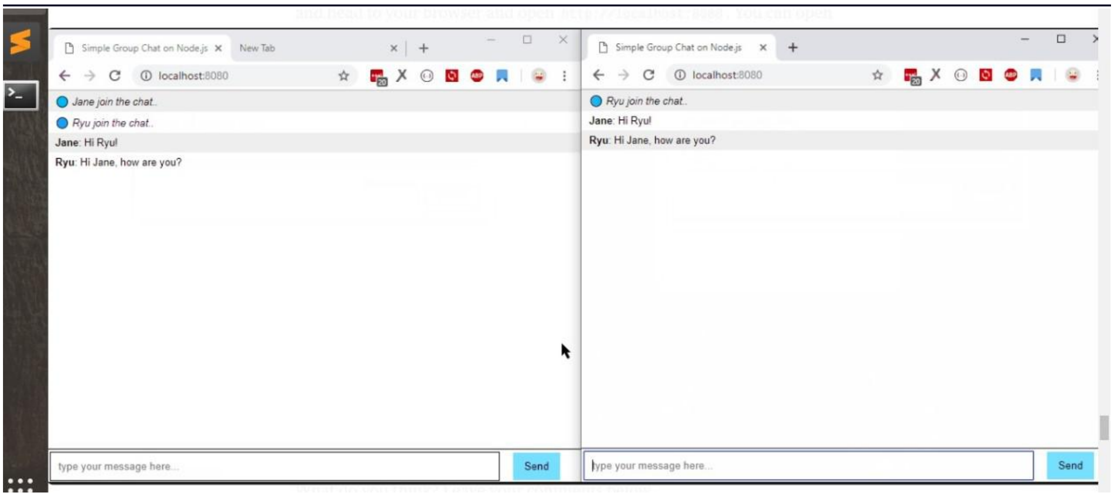
Jane: Hi Ryu!

Ryu join the chat..

Jane: Hi Ryu!

type your message here...

Hi



Jane join the chat..

Ryu join the chat..

Jane: Hi Ryu!

Ryu: Hi Jane, how are you?

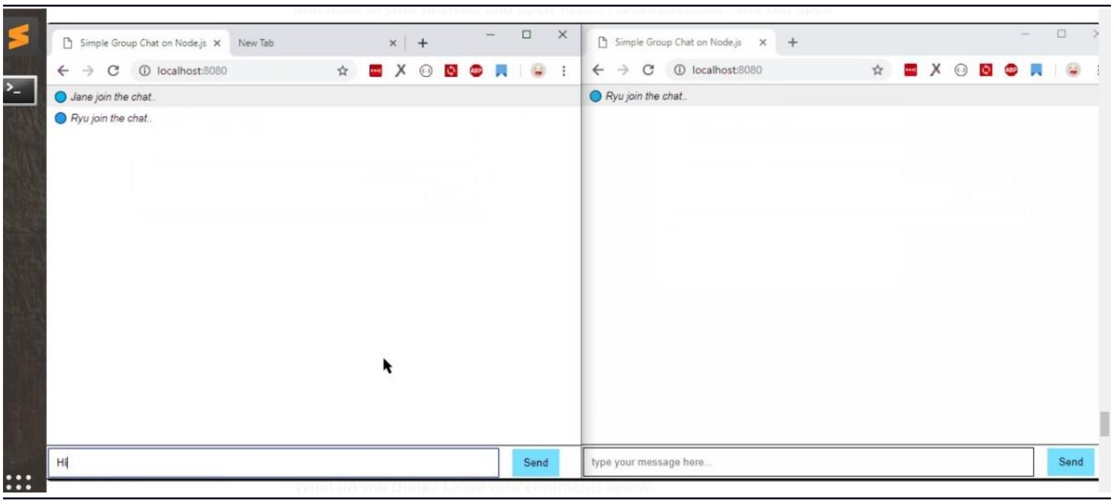
Ryu join the chat..

Jane: Hi Ryu!

Ryu: Hi Jane, how are you?

type your message here...

type your message here...



Jane join the chat..

Ryu join the chat..

Ryu join the chat..

Hi

type your message here...