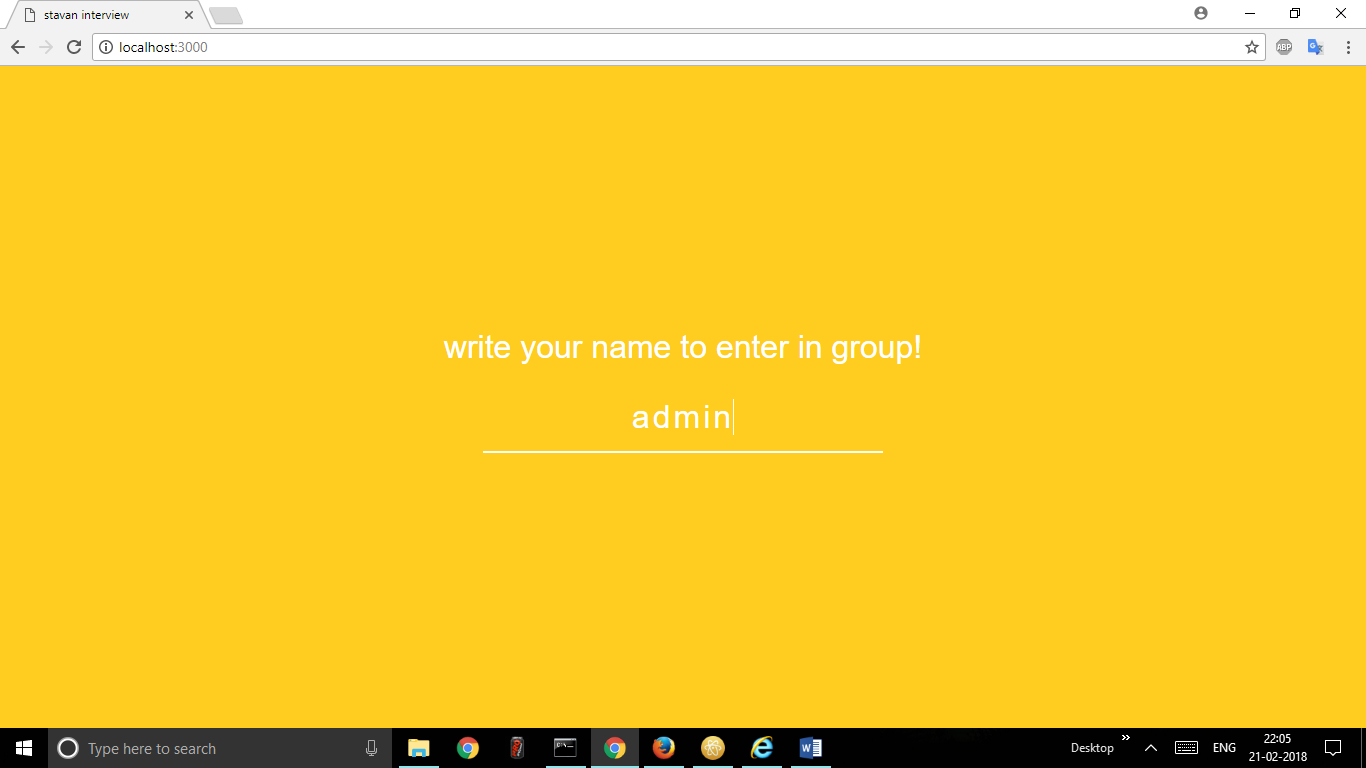
I create simple group chat application in nodejs with socket.io,

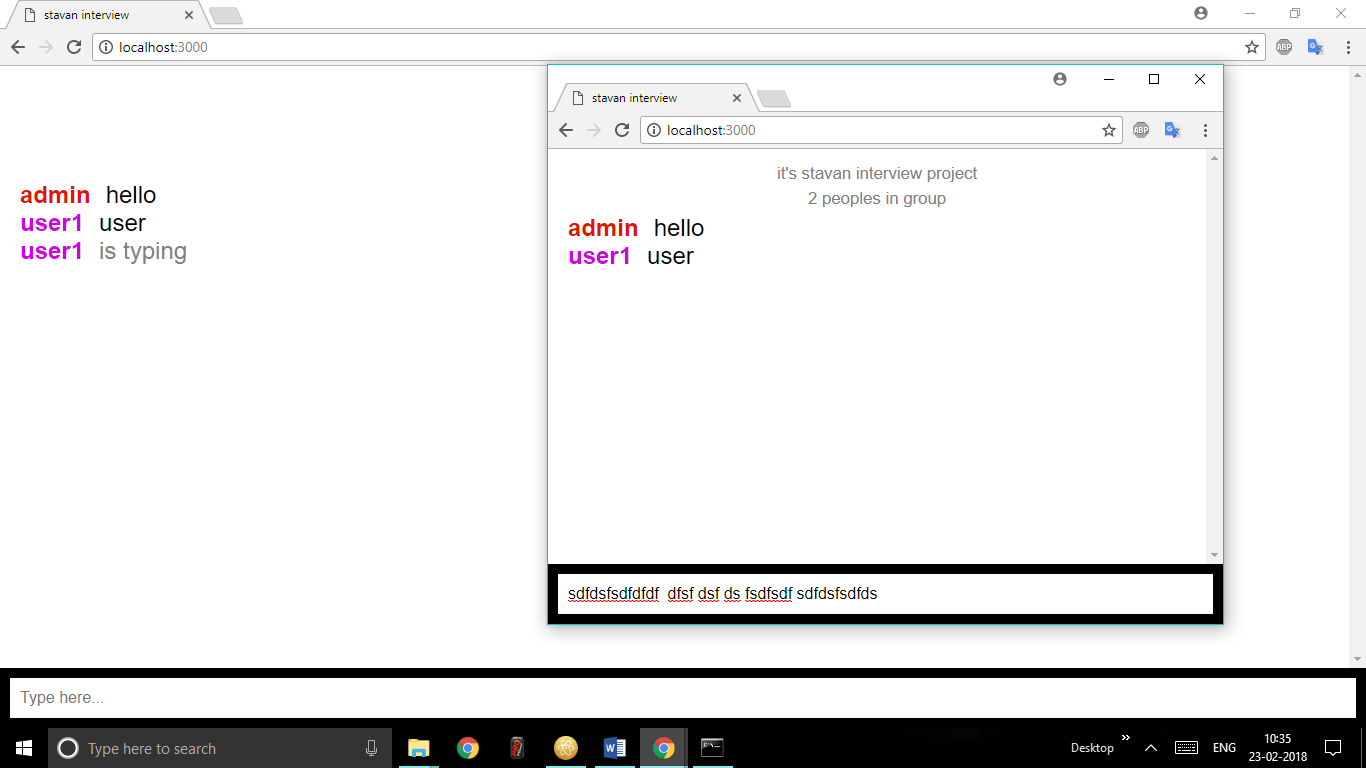
Download this working demo.

https://github.com/stavanparmar/group-chat.git

**How it works!**

1. Go to folder.
2. Open terminal (command prompt) and go to folder path.
3. npm install
4. make sure before execute npm install command you copy all the files to the folder because npm install use you package.json file to install appropriate modules.
5. npm install express –save
6. node index.js
7. go to browser write localhost:3000



1. 

**Index.js**

// Setup basic express server

var express = require('express');

var app = express();

var path = require('path');

var server = require('http').createServer(app);

var io = require('socket.io')(server);

var port = process.env.PORT || 3000;

server.listen(port, function () {

console.log('Server listening at port %d', port);

});

// Routing

app.use(express.static(path.join(\_\_dirname, 'public')));

// Chatroom

var numUsers = 0;

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

var addedUser = false;

// when the client emits 'new message', this listens and executes

socket.on('new message', function (data) {

// we tell the client to execute 'new message'

socket.broadcast.emit('new message', {

username: socket.username,

message: data

}); });

// when the client emits 'add user', this listens and executes

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

if (addedUser) return;

// we store the username in the socket session for this client

socket.username = username;

++numUsers;

addedUser = true;

socket.emit('login', {

numUsers: numUsers

});

// echo globally (all clients) that a person has connected

socket.broadcast.emit('user joined', {

username: socket.username,

numUsers: numUsers

});

});

// when the client emits 'typing', we broadcast it to others

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

socket.broadcast.emit('typing', {

username: socket.username

});

});

// when the client emits 'stop typing', we broadcast it to others

socket.on('stop typing', function () {

socket.broadcast.emit('stop typing', {

username: socket.username

});

});

// when the user disconnects.. perform this

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

if (addedUser) {

--numUsers;

// echo globally that this client has left

socket.broadcast.emit('user left', {

username: socket.username,

numUsers: numUsers

});

}

});

})