**Objective :** Backend: Using NodeJS, build a real time chat application or email sender.

**Index.js**

const express = require('express');

const app = express();

const http = require('http');

const server = http.createServer(app);

const { Server } = require("socket.io");

const io = new Server(server);

app.get('/', (req, res) => {

res.sendFile(\_\_dirname + '/index.html');

});

io.on('connection', (socket) => {

console.log('a user connected');

socket.on('disconnect', () => {

console.log('user disconnected');

});

socket.on('chat message', (msg) => {

console.log('message: ' + msg);

io.emit('chat message', msg);

});

});

server.listen(3001, () => {

console.log('listening on \*:3000');

});

**Index.html**

<!DOCTYPE html>

<html>

<head>

<title>Socket.IO chat</title>

<style>

body {

margin: 0;

padding-bottom: 3rem;

font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif;

}

#form {

background: rgba(0, 0, 0, 0.15);

padding: 0.25rem;

position: fixed;

bottom: 0;

left: 0;

right: 0;

display: flex;

height: 3rem;

box-sizing: border-box;

backdrop-filter: blur(10px);

}

#input {

border: none;

padding: 0 1rem;

flex-grow: 1;

border-radius: 2rem;

margin: 0.25rem;

}

#input:focus {

outline: none;

}

#form>button {

background: #333;

border: none;

padding: 0 1rem;

margin: 0.25rem;

border-radius: 3px;

outline: none;

color: #fff;

}

#messages {

list-style-type: none;

margin: 0;

padding: 0;

}

#messages>li {

padding: 0.5rem 1rem;

}

#messages>li:nth-child(odd) {

background: #cddd9a;

}

</style>

</head>

<body>

<ul id="messages"></ul>

<form id="form" action="">

<input id="input" autocomplete="off" /><button>Send</button>

</form>

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

<script>

var socket = io();

var messages = document.getElementById('messages');

var form = document.getElementById('form');

var input = document.getElementById('input');

form.addEventListener('submit', function(e) {

e.preventDefault();

if (input.value) {

socket.emit('chat message', input.value);

input.value = '';

}

});

socket.on('chat message', function(msg) {

var item = document.createElement('li');

item.textContent = msg;

messages.appendChild(item);

window.scrollTo(0, document.body.scrollHeight);

});

</script>

</body>

</html>

**OUTPUT**

**User a: User b:**

