

Assignment II

Sadrul Alam Molla

001710501012

BCSE-IV

Jadavpur University

Overview

A multi-client chat application consisting of both client and server programs. In this chat application simultaneously several clients can communicate with each other. For this a single server program is being used that clients connect to. The client programs send the chat text or image (input) to the server and then the server distributes that message (text or image) to all the other clients. Each client then displays the message sent to it by the server. The server should be able to handle several clients concurrently. It works fine as clients come and go.

Goals

1. Make a chat application that can send and receive text and image data
2. A single server program should be used.

Specifications

- The server program is being written in JavaScript, and the Node.js framework has been used to run the program.
- To handle concurrent messages, WebSockets have been used. The following is a brief description about WebSockets:
 1. The WebSocket specification defines an API establishing "socket" connections between a web browser and a server. In plain words: There is a persistent connection between the client and the server and both parties can start sending data at any time.
 2. Connection limitations are no longer a problem since WebSockets represent a single TCP socket connection.
 3. Cross domain communication has been considered and is dealt with within the connection handshake.
 4. Using WebSocket creates a whole new usage pattern for server side applications. While traditional server stacks such as LAMP are designed around the HTTP request/response cycle they often do not deal well with a large number of open WebSocket connections.

Keeping a large number of connections open at the same time requires an architecture that receives high concurrency at a low performance cost. Such architectures are usually designed around either threading or so called non-blocking IO.

- The client program is being written in HTML, jquery, CSS and the inner script in HTML uses JavaScript. The client can be run on any web browser.
- The following experiments have been performed:
 1. Unique username – A user cannot have the same client id at the same time. If a new user tries to enter an existing client id of a running client, an alert is being shown.
 2. Empty input while sending a text. Users need to at least enter a character, otherwise it raises an alert showing that “Please Enter The Text :)”
 3. Multiple clients (multiple users) can join the chat group, wherein each user is being notified about “joining” and “leaving” of any user in the group.
 4. Each client can send a text message or image. If the user wants to send text then click the “send ” button after typing the message & to send image data click the “upload image” button and choose the image.
 5. If the text/image is sent by this user it will show on the right side of the chat and will be of orange color otherwise it will be on the left side and will be of blue color.
 6. While pressing the carriage return key on writing a text, the client adds a new line. Hereby, a user can enter many new line-separated sentences in the same message.

Code Snippet

Server Program(server.js):

```
var server = require('ws').Server;
var s = new server({ port : 8081 });
var present = {};
s.on('connection', function(ws)
{
    ws.on('message', function(message)
    {
        message = JSON.parse(message);
        if(message.type == "username")
        {
            ws.username = message.data;
            if(present[ws.username] == 1)
            {
                ws.username = "duplicateuser";
                present[ws.username] = 0;
                ws.send(JSON.stringify({
                    type: "errmsg",
                    data: "DUPLICATE-CLIENT"
                }));
                ws.close();
            }
            else
            {
                present[ws.username] = 1;
                s.clients.forEach(function(client){
                    client.send(JSON.stringify({
                        type: "username",
                        data: ws.username
                    }));
                });
                console.log(ws.username+" joined the chat");
            }
        }
        if(message.type=="text" || message.type=="binary" ||
message.type=="user")
        {

```

```

        s.clients.forEach(function(client){
            client.send(JSON.stringify({
                type: message.type,
                data: message.data,
            }));
        });
    }
});
ws.on('close', function()
{
    if(present[ws.username] == 1)
    {
        present[ws.username] = 0;
        console.log(ws.username+" left the chat");
        s.clients.forEach(function(client){
            client.send(JSON.stringify({
                type: "message",
                data: ws.username+" left the chat"
            }));
        });
    }
});
});

```

Client Program:

```

<body>
    <div class="card-footer">
        <input type="text" onkeydown="enterKeyPressed(event)" placeholder="Type
your message..." id="puttext">
        <button id="textButton">SEND</button>
        <button id="addImage">Upload image</button>
        <input type="file" id="fileinput"><br><br>
    </div>
    <div id="container"></div>
</body>
</html>
<script type="text/javascript">
    var username = prompt("Please enter your Username");
    var sock = new WebSocket("ws://localhost:8081");
    var log = document.getElementById('container');
    var n=username.length;
    var bl=username;
    function enterKeyPressed(evt)
    {
        evt = evt || window.event;
        if (evt.keyCode == 13)

```

```

        {
            document.getElementById('puttext').value += '/';
        }
    }
    sock.onopen = function()
    {
        sock.send(JSON.stringify({
            type: "username",
            data: username
        }));
    }
    sock.onmessage = function()
    {
        $("#container").append("<br>");
        var json = JSON.parse(event.data);
        if(json.type == "errmsg" && json.data == "DUPLICATE-CLIENT")
        {
            alert(username+" is already Taken! Please try another name");
            return;
        }
        else if(json.type == "username")
        {
            if(json.data != username)
            {
                log.innerHTML += '<p style="text-align: center;">'+json.data+"
joined the chat</p>";
            }
        }
        else if(json.type == "text")
        {
            var str=json.data.substring(3,n+3);
            var m=json.data.length;
            console.log(str);
            if(str!= username)
            {
                log.innerHTML += '<div class="message-blue"><p
class="message-content">'+json.data+"</p></div>";
            }
            else
            {
                log.innerHTML += '<div class="message-orange"><p
class="message-content">'+<b>You'+json.data.substring(n+3,m)+"</p></div>";
            }
        }
        else if(json.type=="binary")
        {
            var img=json.data;
            if(bl!=username)

```

```

        {
            log.innerHTML += '<div
class="message-blue"><b>'+bl+'</b><br>'+json.data+'</div>';
        }
        else
        {
            log.innerHTML += '<div
class="message-orange"><b>You:</b><br>'+json.data+'</div>';
        }
        $("#log").append("<br>");
    }
    else if(json.type=="message")
    {
        log.innerHTML += '<p style="text-align: center;">'+json.data+'</p>';
    }
    else
    {
        bl=json.data;
    }
}
document.getElementById("textButton").onclick=function()
{
    var Text = document.getElementById('puttext').value;
    if(Text=="")
    {
        alert("Please Enter The Text :");
        return;
    }
    var newText = "<b>"+username+'</b><br>';
    for(var i=0;i<Text.length;i++)
    {
        if(Text.charAt(i) == '/')
        {
            newText += '<br>';
        }
        else
        {
            newText += Text.charAt(i);
        }
    }
    sock.send(JSON.stringify({
        type: "text",
        data: newText,
    }));
    document.getElementById('puttext').value = "";
}
$(document).ready(function(){
    $("#addImage").click(function()

```

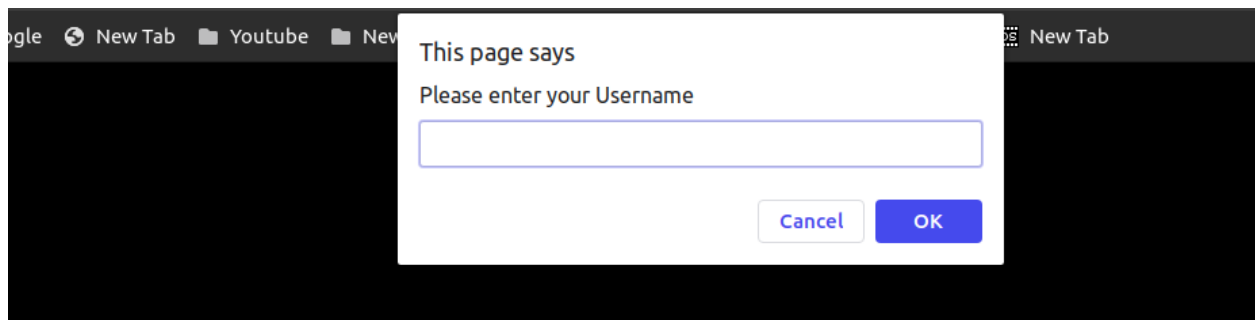
```

{
    $("#fileinput").click();
});
$("#fileinput").change(function()
{
    if (this.files && this.files[0])
    {
        var reader = new FileReader();
        reader.onload = imageIsLoaded;
        reader.readAsDataURL(this.files[0]);
    }
});
function imageIsLoaded(e)
{
    var x = 'foo';
    var picture = ''
    sock.send(JSON.stringify({
        type: "user",
        data: username,
    }));
    sock.send(JSON.stringify({
        type: "binary",
        data: picture,
    }));
}
});
</script>

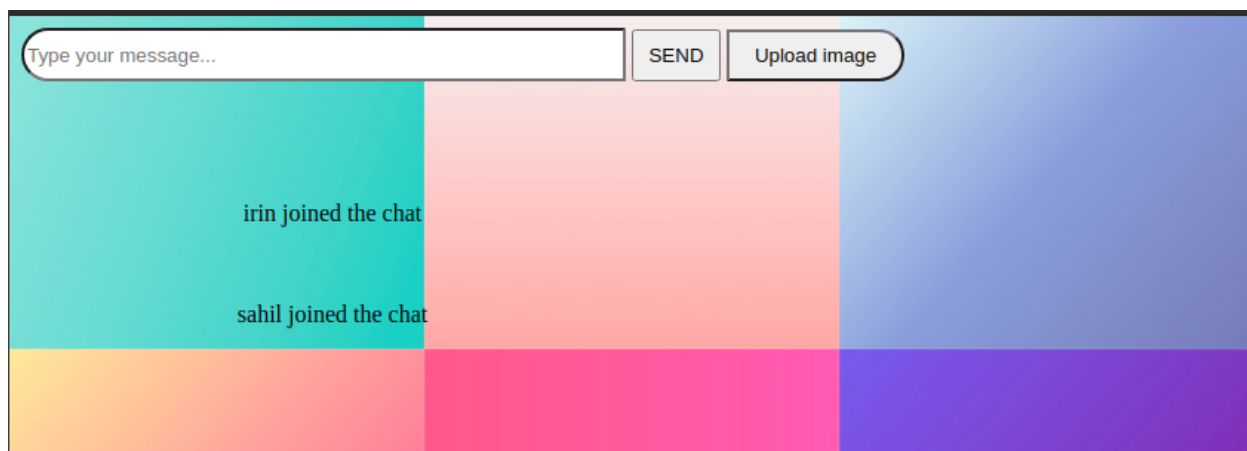
```

Output

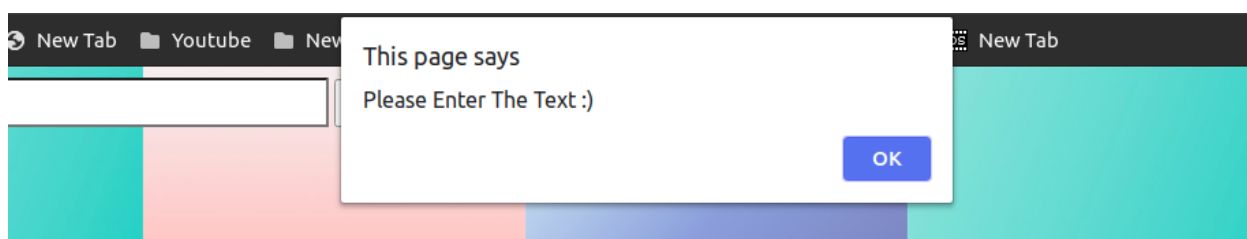
Server



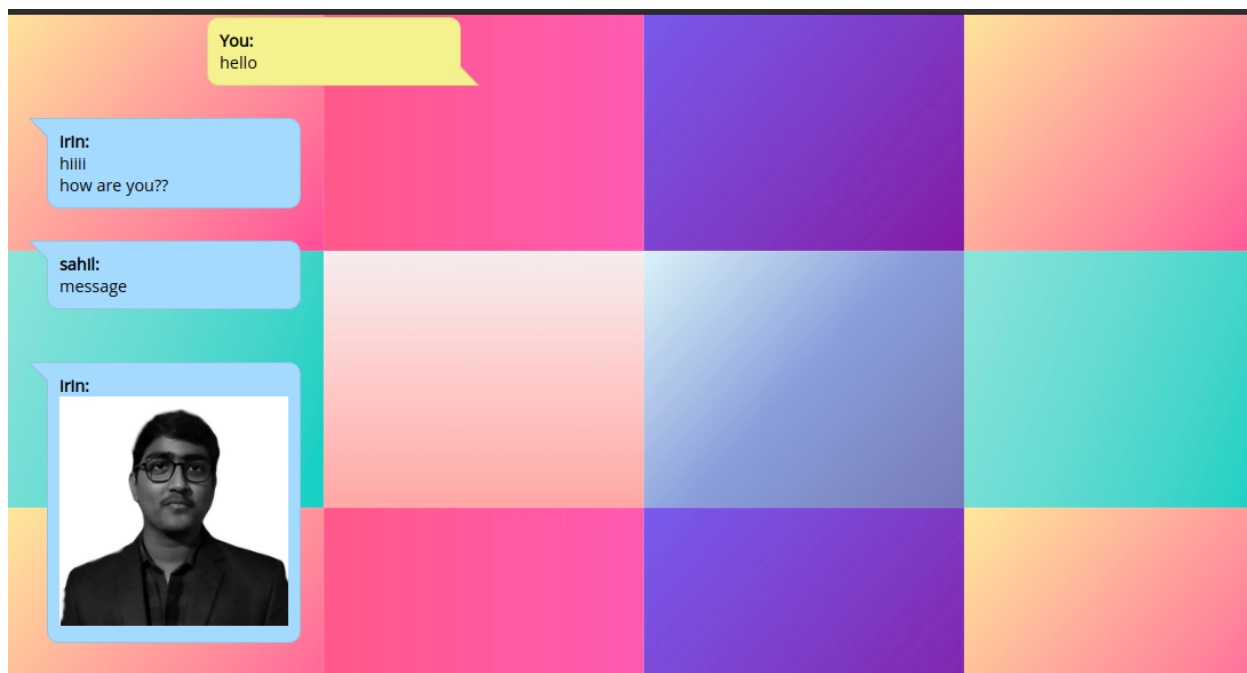
When open a new client chat it asks for your username



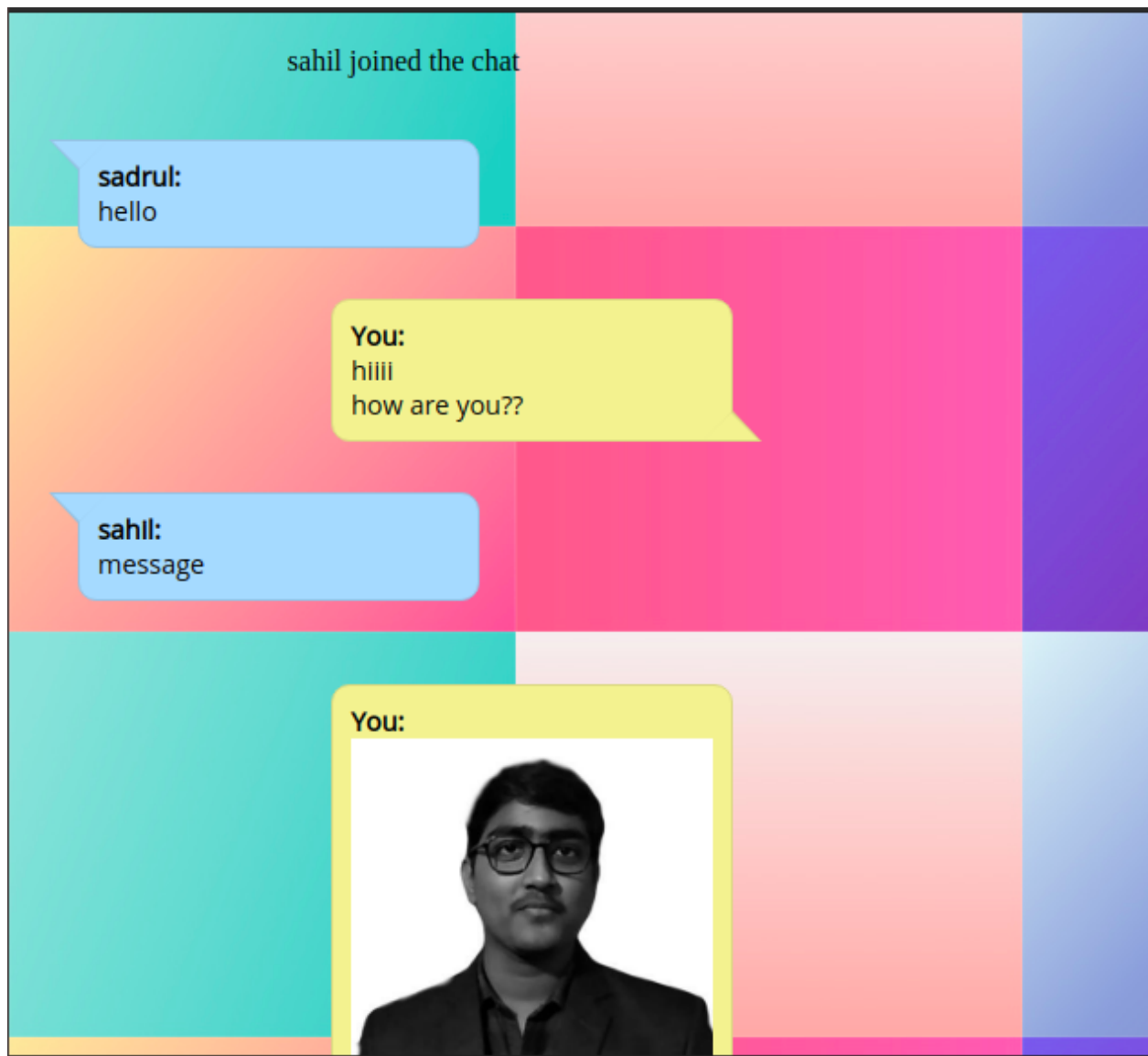
When a new client enter the chat



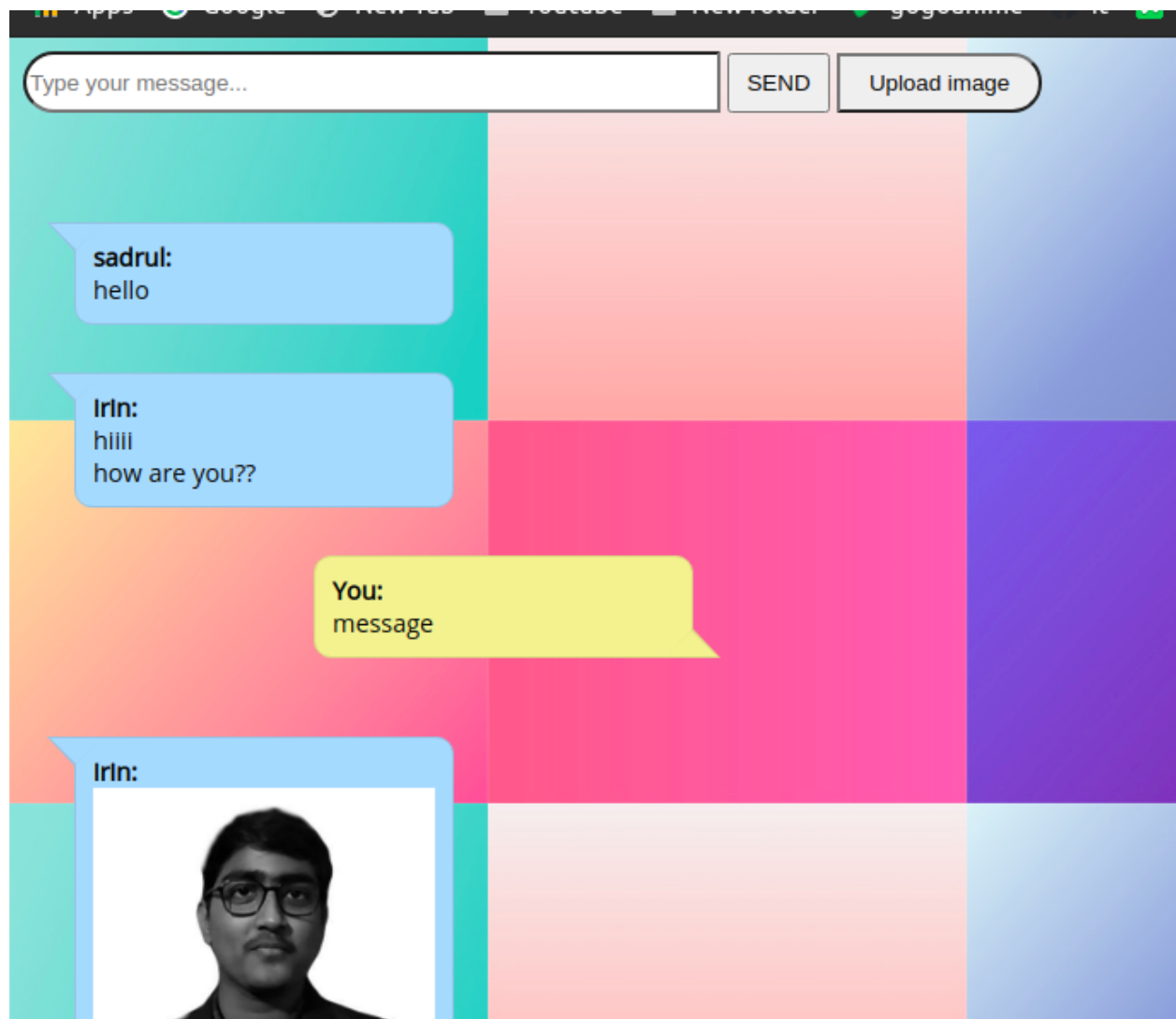
Alert message if you try to send a empty text



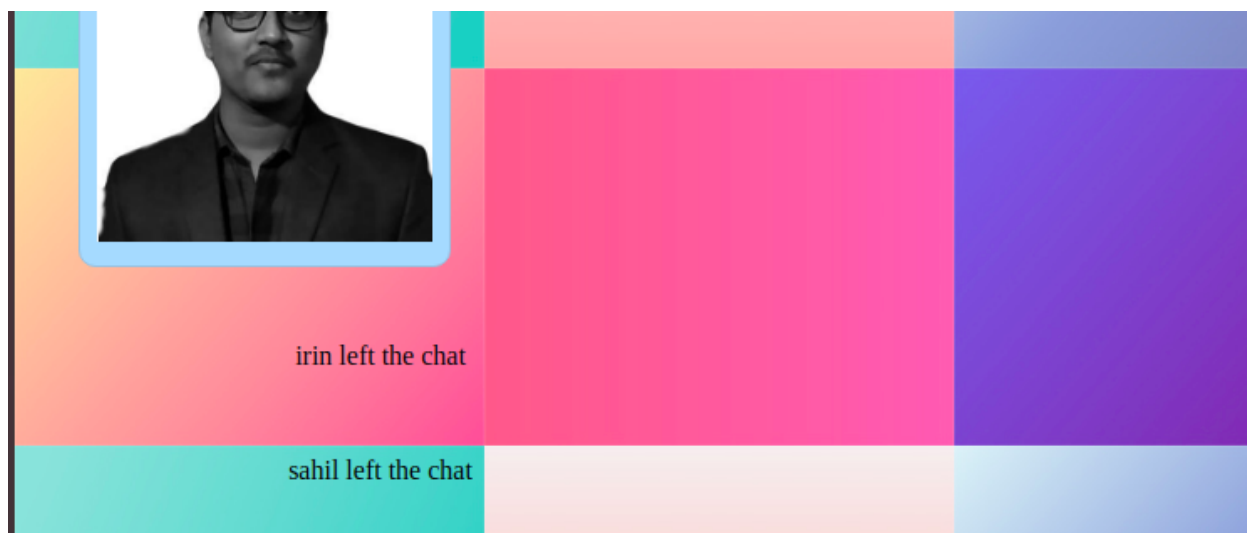
Chat 1



Chat 2



Chat 3



When someone left the chat

```
sadrul@sadrul:~/Desktop/Internet_tech/ass2/code$ node server.js
sadrul joined the chat
irin joined the chat
sahil joined the chat
irin left the chat
sahil left the chat
□
```

Server