

嵌入式實驗 Experiment No.1 : Ping Chat

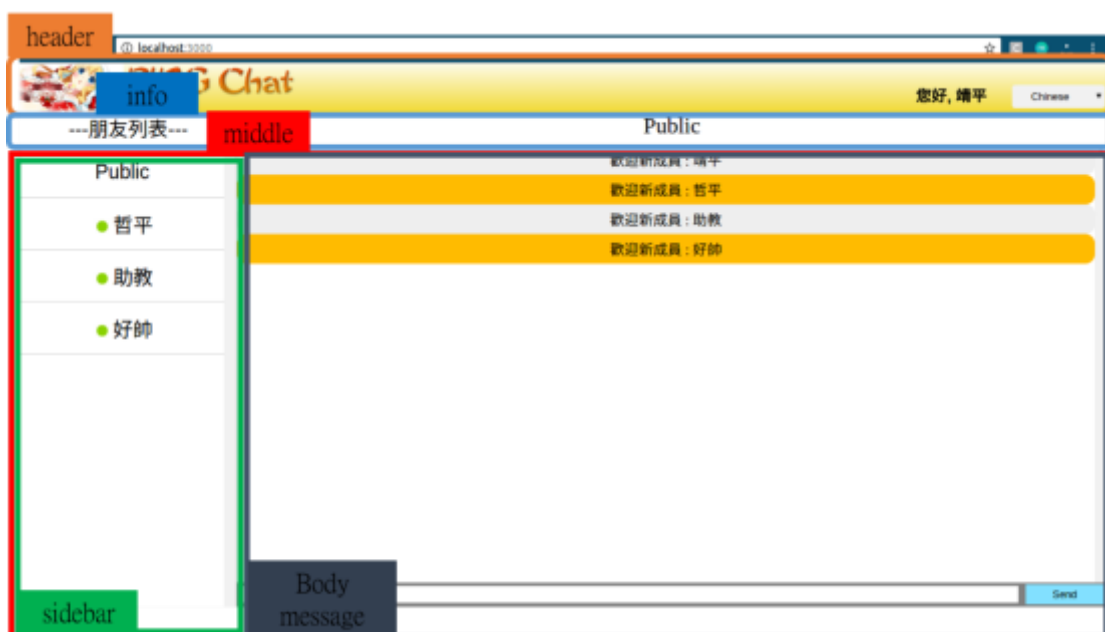
電機四 B02901073 楊靖平

電機四 B02901181 蔡哲平

一、功能介紹

a. 介面介紹：

聊天室主要區塊(<div>)：header、info、middle(含有 sidebar、bodymessage)



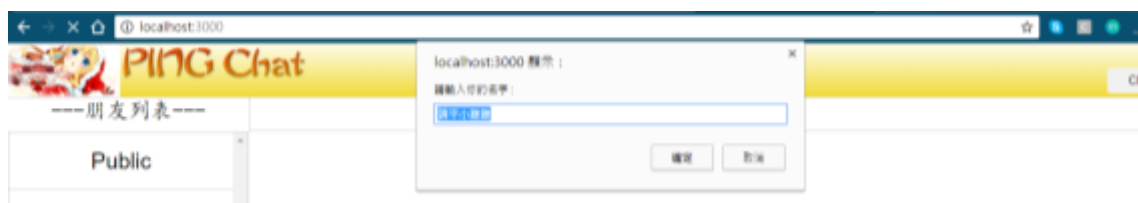
HTML Code

```
chat.html
10 <div id="header">
11 <div id="top1">
12 
13 </div>
14 <div id="top2">
15 
16 </div>
17 <div id="top3">
18 <form>
19 <select>
20 <option value="ch">chinese</option>
21 <option value="en">English</option>
22 <option value="jp">Japanese</option>
23 <option value="fr">French</option>
24 </select>
25 </form>
26 </div>
27 <div id="top4">
28 <p id="hello"></p>
29 </div>
30 </div>
31 <div id="info">
32 <div id="c-name">
33 <p id="fr"> ---朋友列表--- </p>
34 </div>
35 <div id="c-name">
36 <p id="who"> Public</p>
37 </div>
38 </div>
39 <div id="middle">
40 <div id="sidebar">
41 <div class="img">
42 <button id="public" onclick="clicked(this.id)"> <span>Public</span> </button>
43 </div>
44 </div>
45 <div id="bodymessage">
46 <div id="m">
47 <ul id="messages"></ul>
48 </div>
49 <div id="5">
50 <form action="">
51 <input id="m" autocomplete="off" />
52 <button>Send</button>
53 </form>
54 </div>
55 </div>
56 </div>
57 <script src="/socket.io/socket.io.js"></script>
```

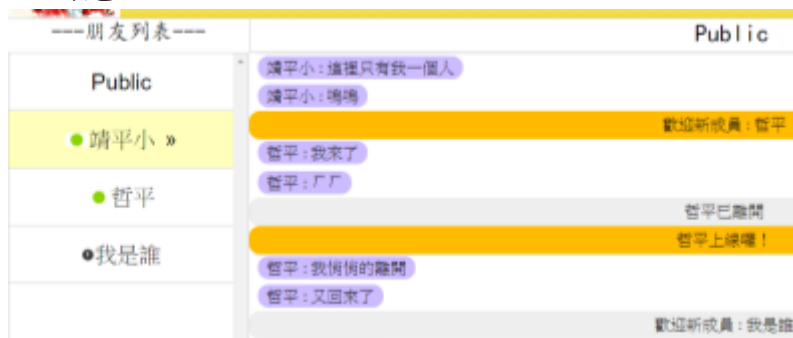
b. 主要功能



1. 系統提示：系統在 Public(公共聊天室)上，在每個人上限或離線時，皆會有提示訊息。(歡迎新成員 X X or X X 已離開)
2. 與誰聊天：Ping Chat 中有群聊和私訊功能，在聊天室最上方，清楚得知現在自己在跟誰聊天，完全不會錯頻。
3. 誰在線上：左邊框框可以看到自己的朋友們，綠燈表示現在在線上，灰色表示離線。
4. 登入系統時(如下)，系統自動提示請輸入名字，如此以方便我們存取聊天資料。以確保再次登入，聊天訊息依然存在！



5. 當滑鼠在左方朋友列表指向任何一個人名，系統貼心的加上底色，以及動畫效果(非常好玩)，以提醒您在滑鼠按下去那一瞬間，會跑出和誰的對話訊息。



二、程式架構

a. 前端

```
function addGroupChat(data) {
  if (data.broadcast === true) { // 群聊 - broadcast
    var li = $('<li id="bc"/>');
    var span = $('<span>').text(data.msg);
    li.append(span);
    $('#messages').append(li);
  } else { // 群聊 - nonbroadcast
    var id = data.name === username ? "me" : "friend";
    var msg = data.name + ' : ' + data.msg;
    var li = $('<li id="' + id + '"/>');
    var span = $('<span>').text(msg);
    li.append(span);
    $('#messages').append(li);
  }
  $("#bodymessage").scrollTop($("#bodymessage")[0].scrollHeight);
}

function addPrivateChat(data) {
  var id = data.name === username ? "me" : "friend";
  console.log(id);
  var li = $('<li id="' + id + '"/>');
  var span = $('<span>').text(data.msg);
  li.append(span);
  $('#messages').append(li);

  $("#bodymessage").scrollTop($("#bodymessage")[0].scrollHeight);
}
```

PING CHAT 分為“群聊”以及“私聊”

addGroupChat 和 addPrivateChat 兩個函式在說明當我們系統發送聊天訊息給使用者時，如何及時更新在頁面上。

Ps. 滾輪會自動選到最下面以讀取最新訊息！

我們在加入左方朋友列表的按鈕時，考慮了滑鼠滑過時的效果、點下去切換視窗、以及如何動態新增朋友的框框(新使用者加入)，以下逐一介紹。

1. 滑鼠滑過時的效果：我們將朋友列表的按鈕的 id 均設為“smg”，當滑鼠滑過(hover)時，cursor 的 opacity 會變成 1(不透明)，將文字向左推，產生動態的效果。

```
208 .smg {
209     transition-duration: 0.4s;
210 }
211
212 .smg: hover {
213     background-color: #FFFFBB;
214     color: white;
215 }
216
217 .smg span {
218     cursor: pointer;
219     display: inline-block;
220     position: relative;
221     transition: 0.5s;
222 }
223
224 .smg span: after {
225     content: '\00bb';
226     position: absolute;
227     opacity: 0;
228     top: 0;
229     right: -20px;
230     transition: 0.5s;
231 }
232
233 .smg: hover span {
234     padding-right: 25px;
235 }
236
237 .smg: hover span: after {
238     opacity: 1;
239     right: 0;
240 }
241
```

2. 點朋友列表後切換視窗：點下按鈕時，我們會先確認目前聊天室頁面(nowpage)是否和點下的名字相符，相符代表不用切換，反之，我們將聊天室裡面的訊息先通通刪去，再補上預先儲存好的聊天內容，根據聊天內容的傳送者(自己 or 朋友)，區分成左右兩邊顯示。

```
101
102     function clicked(name) {
103         console.log("clicked : " + name);
104         if (nowpage != name) {
105             nowpage = name;
106             document.getElementById("who").innerHTML = nowpage;
107             $('#messages').empty();
108             if (nowpage == "public") {
109                 for (var i = 0; i < groupChat.length; i++)
110                     addGroupChat(groupChat[i]);
111             } else {
112                 for (var i = 0; i < privateChat.length; i++) {
113                     if (privateChat[i].name == nowpage)
114                         addPrivateChat(privateChat[i]);
115                     else if (privateChat[i].rcv == nowpage)
116                         addPrivateChat(privateChat[i]);
117                 }
118             }
119         }
120     }
121 }
```

3. 如何動態新增朋友的框框(新使用者加入)

```
136
137 socket.on('show private user', function(user) {
138     if (user.name != username) {
139         console.log(user.name + " is " + user.online);
140         var d = $('<div class="msg"/>');
141         var r = $('<button id="' + user.name + '" onclick="clicked(this.id)"/>');
142         var span = $('<span>').text(user.name);
143         if (user.online == true) {
144             var img = $('');
145             r.append(img);
146         } else {
147             var img = $('');
148             r.append(img);
149         }
150         r.append(span);
151         d.append(r);
152         $('#sidebar').append(d);
153     }
154 });
```

每當出當入系統時，或是有新使用者加入，我們會在朋友列表新增朋友按鈕，並顯示是否在線/離線。

此為系統傳送有人離線或是再上線的提示，讓使用者即時得知朋友們的狀態。

```
155
156 socket.on('a user disconnect', function(name) {
157     if (name != username)
158         document.getElementById('_' + name).src = "offline.jpg";
159 });
160 socket.on('a user reconnect', function(name) {
161     if (name != username)
162         document.getElementById('_' + name).src = "online.jpg";
163 });
```

b. 後端

1. 資料處理：

message data:

data.name：傳訊息的人

data.rcv：收訊息的人

data.msg：訊息內容

data.broadcast：是否為系統訊息

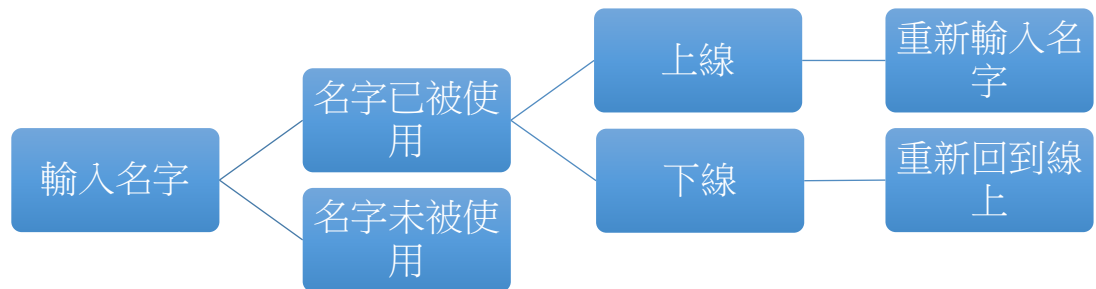
user data

user.name：使用者名稱

user.online：是否在線上

2. 登入系統功能：

使用 **prompt** 功能實作。並且使用者名稱不得重複，判斷式如下。



3. 群聊功能：

當伺服器接收到 **socket** 得到的群聊訊息，便會紀錄寄件人，並發送此訊息給所有 **socket**。其中，系統廣播和使用者訊息會根據 **data.broadcast** 區分，並以不同樣式的訊息框展現。

4. 私聊功能：

client 端會記錄此訊息的收件人（根據當前頁面判斷）和寄件人，並傳給 **server** 端，**server** 端會將此訊息傳給每個 **client** 端，若 **client** 與收件人或寄件人相符，便會接收此訊息。

此方法可能有潛在危險，可能訊息會遭旁人竊聽，此外，因為沒有在 **socket** 之間建立連線，將此訊息寄給不必要的人會造成資源浪費。

三、遭遇困難

- a. 一開始無法讀取在同一個資料夾內的其他資訊，查了很多資料，發現需要加入 **public** 路徑 (第 17 行)，才能讀到其他資訊如 **css,img** 檔。

```
14  
15 var express = require('express');  
16 var path = require('path');  
17 app.use(express.static(__dirname, 'public'));  
18
```