

ESLab-Hw1

Chat on Line

b02901183 林彥廷

b02203029 許鎔嶽

系統介紹:

a. login.html

- ◆ `sessionStorage.clear()` 會清空所有的資料，讓一開始的登入介面是乾淨的。
- ◆ `sessionStorage` 負責將帳號與密碼回傳給 `server.js`，若帳號與密碼任一未輸入則出現錯誤警訊。

b. chat.html

- ◆ `function timetostring()` 負責顯示每則留言的時間。
- ◆ `function addOnlineUser()` 負責顯示頁面新增的聊天帳號對話框。
- ◆ `function addNewMsg()` 留完訊息按下 **Enter** 後，此函式會將有效字串新增在一個新的對話框上。

c. reg.html

- ◆ `sessionStorage.clear()` 會清空所有的資料，讓一開始的登入介面是乾淨的。
- ◆ `function check_account()` 會檢查帳號的有效性，若有效顯示綠色，無效則顯示紅色。
- ◆ `function check_password()` 會檢查密碼的重複性，若有重複顯示綠色，無重複則顯示紅色。

d. regfinish.html

- ◆ 註冊完的結束頁面，會在 5 秒後自動導向主頁面(登入頁面)。

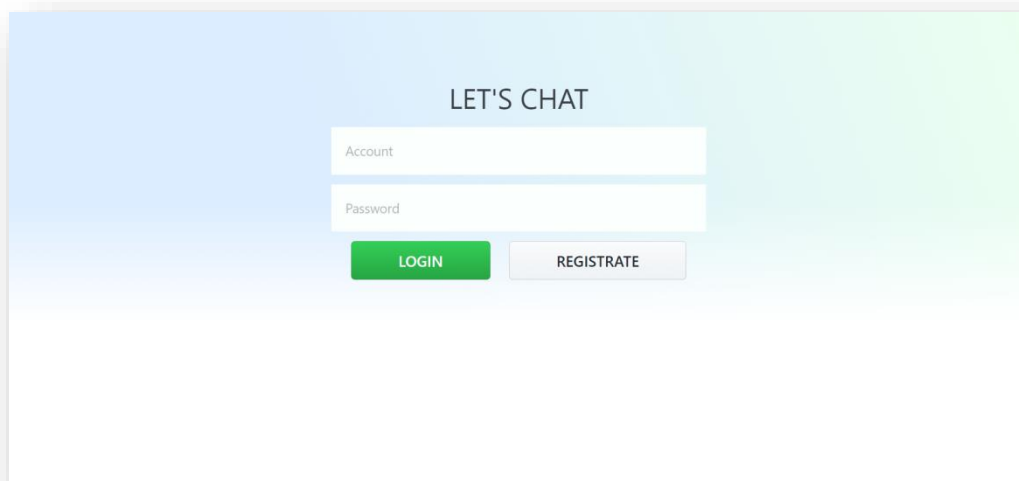
e. server.js

- ◆ `io.on("connection", function(socket))` 建立連線與 socket 設定。
- ◆ `socket.on("login verify", function(username, password))` 負責確認是否是有效登入。
- ◆ `socket.on("robot", function())` 在註冊頁面負責 random 產生重複確認 robot 的順序。
- ◆ `socket.on("chat message", function(username, key, msg))` 建立一對一連線後，每當有新訊息 send 時，socket 會給予新的對話框，包括使用者資訊、聊天內容、時間。
- ◆ `socket.on("disconnect", function())` 離線時，server 端會顯示 client gone。

聊天室介紹:

a. 註冊帳號

首先，先進入主頁面，一開始因為沒有註冊，沒辦法 LOGIN，我們可以按下右下方的 REGISTRATE 鍵進行註冊。



Initial login form titled "LET'S CHAT". It features two input fields: "Account" and "Password". Below the fields are two buttons: a green "LOGIN" button and a grey "REGISTRATE" button.

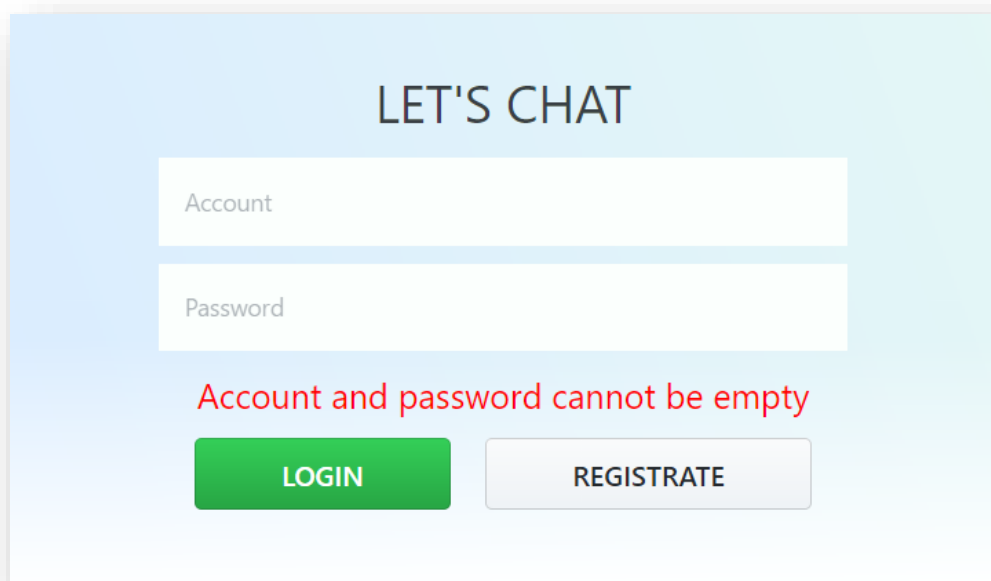
LET'S CHAT

Account

Password

LOGIN REGISTRATE

在註冊之前，因為帳號與密碼欄皆不能為空白，若直接按下 LOGIN，頁面會顯示不能為空白的紅語標示。



Login form titled "LET'S CHAT" after clicking the LOGIN button. The "Account" and "Password" fields are empty. A red error message "Account and password cannot be empty" is displayed below the fields. The "LOGIN" and "REGISTRATE" buttons remain at the bottom.

LET'S CHAT

Account

Password

Account and password cannot be empty

LOGIN REGISTRATE

LET'S CHAT
REGISTRATION

Account: valid account

Password: match!

Email:

☐ I am a robot
☒ I am not a robot
☐ I am a robot
☐ I am a robot

接著我們就按下 **REGISTRATE** 鍵進行註冊，畫面會跳到註冊頁面，我們先註冊一個帳戶名為 **Mary** 的帳號，這邊我們有幾個驗證帳號的小功能，像是如果帳號先前尚未有人註冊過，就會顯示 **valid account**，如果第二次輸入密碼符合第一次輸入，輸入欄右邊也會出現 **match** 的標誌提醒，但如果 **Mary** 已經被註冊過，則會出現這樣的頁面。

LET'S CHAT
REGISTRATION

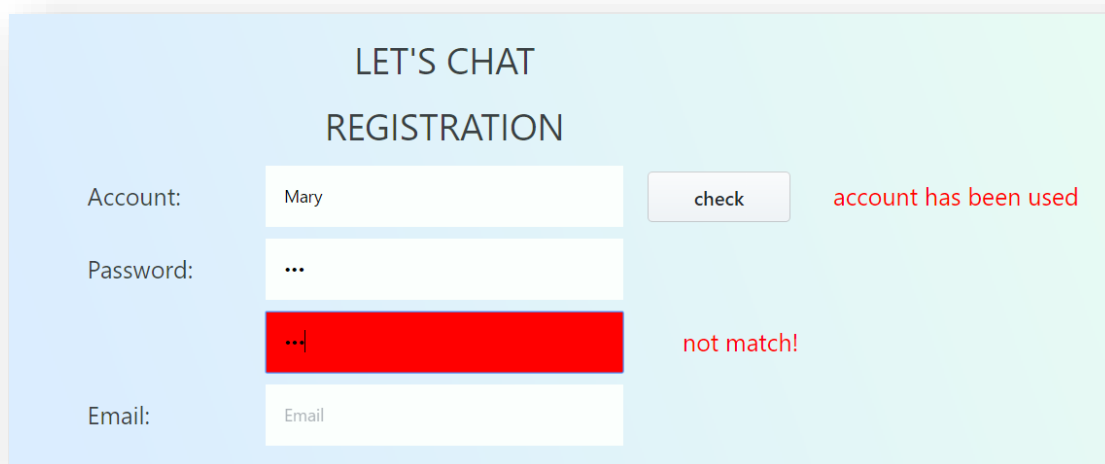
Account: account has been used

Password:

Email:

☐ I am a robot
☐ I am a robot
☐ I am not a robot
☐ I am a robot

如果第二次輸入的密碼不符合第一次輸入，則頁面會出現警訊。



LET'S CHAT
REGISTRATION

Account: account has been used

Password:
 not match!

Email:

驗證欄若勾選錯誤，是沒有辦法通過認證的。



☐ I am a robot

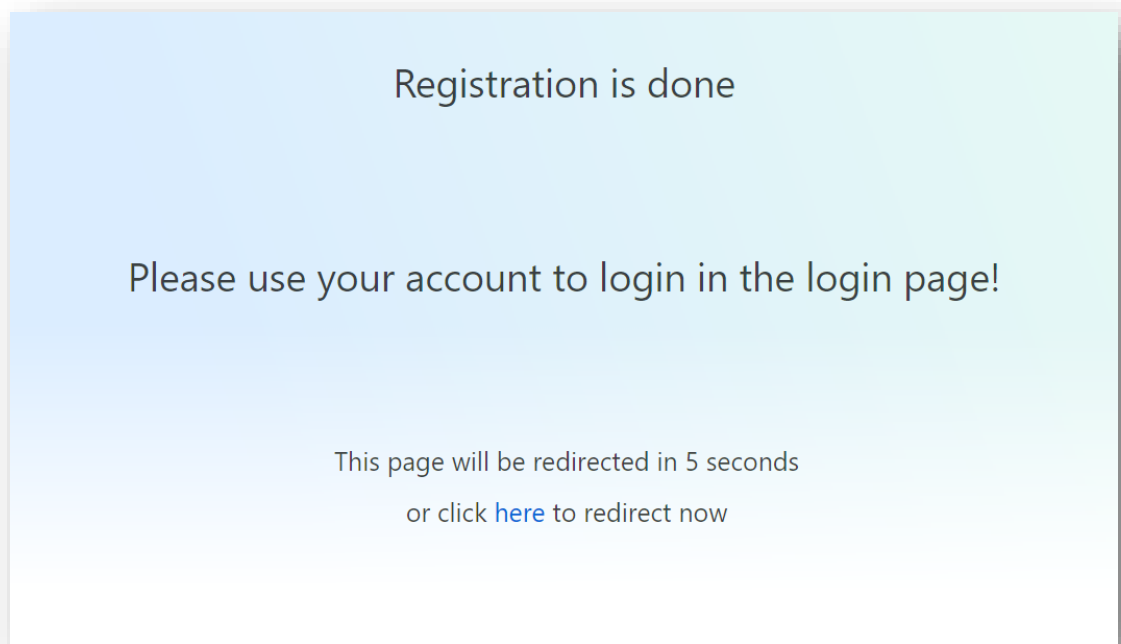
☐ I am a robot

☐ I am not a robot

☒ I am a robot

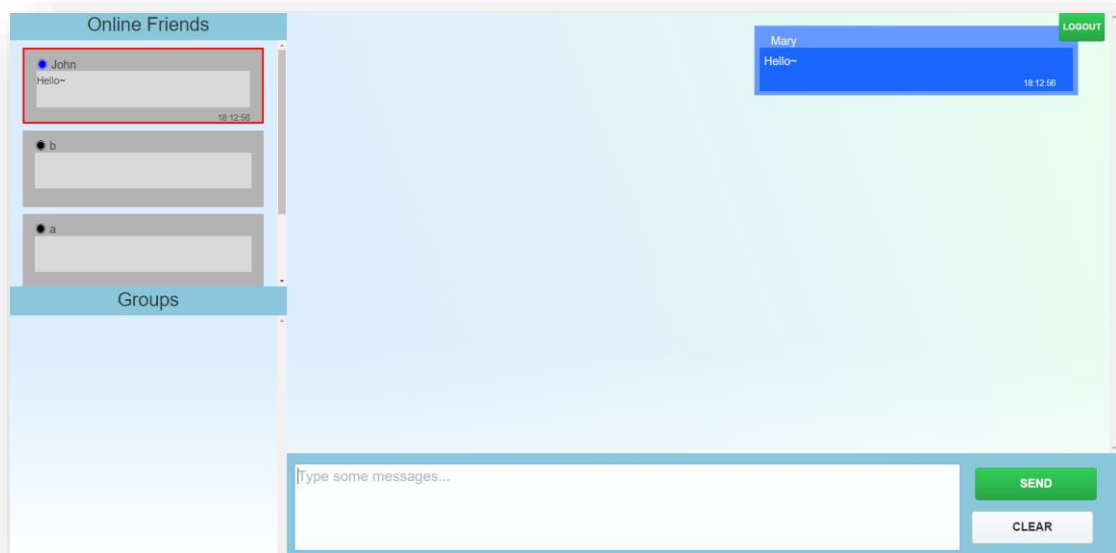
You are a robot!!!

成功註冊後，畫面會引導你前往 login 的頁面，如果停留時間超過 5 秒，系統也會自動導向。

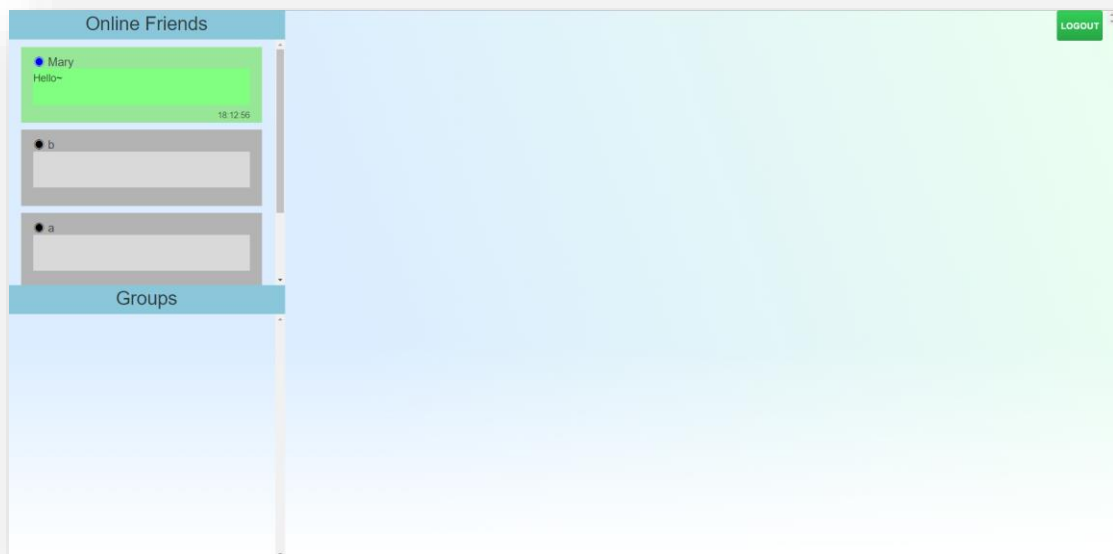


b. 一對一聊天

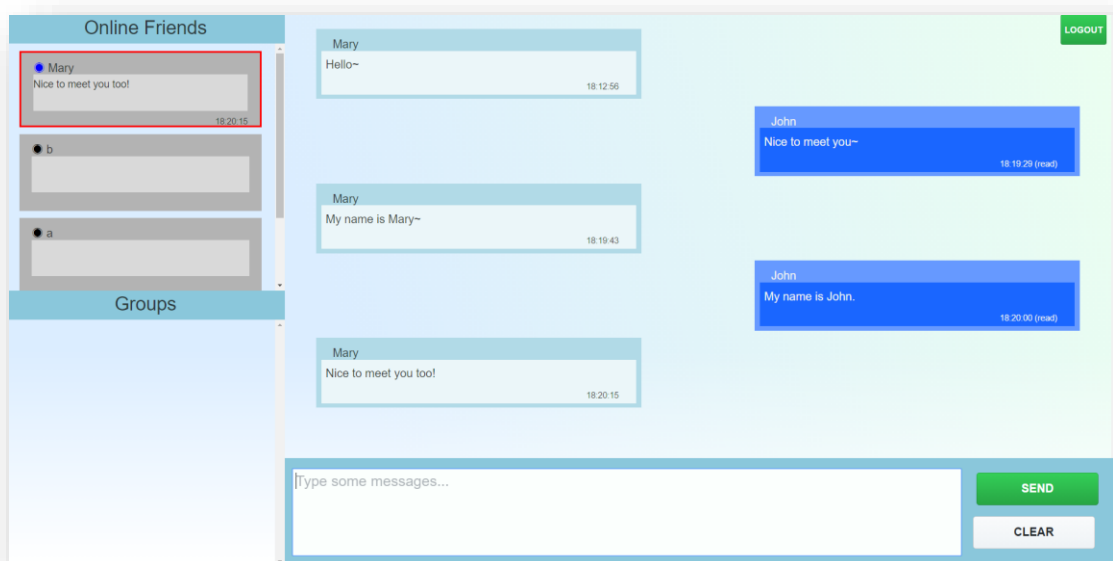
成功登入後會進入聊天畫面。



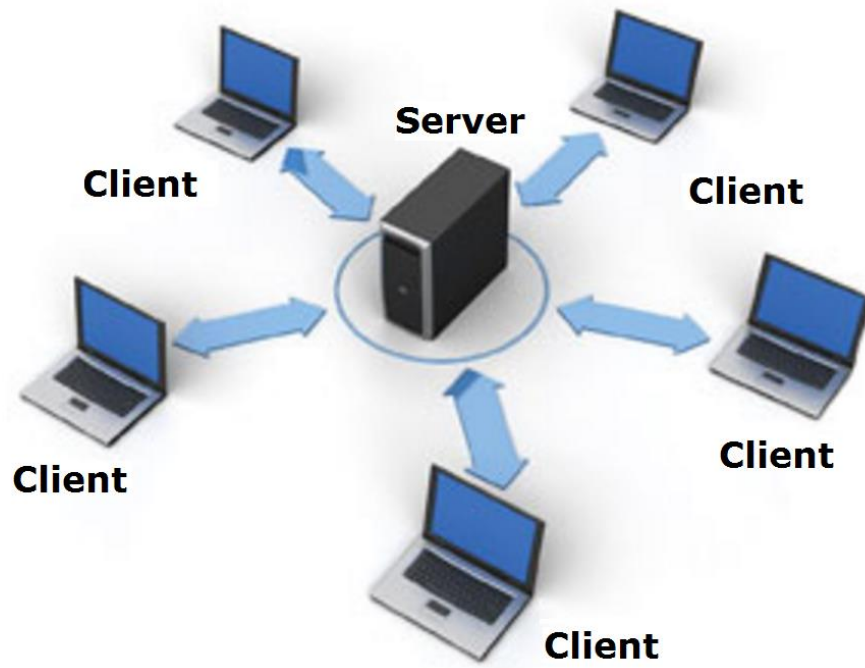
左上角是目前在線上的人有誰，藍色代表目前上線中，黑色代表離線，上圖中用 Mary 的帳戶向 John 傳送了一則訊息，John 的頁面會看到一則新通知，如下，



綠色代表有新訊息通知，點開後就可以開始聊天了。



架構:



在 login 頁面時，首先 client 端需先提交一份 account 與 password 給 server，server 在確認可以使用後會回傳一份相同的 account 與專屬的 key 給 client，並將資料備份在 chat.html 的 sessionStorage 裡，此目的是不想讓 client 輸入的密碼外流，只有 server 端才能存取，當帳號申辦好後，server 會在之後每個開啟的新聊天室窗架上所有的聊天帳號，每新增加一個帳號，server 會自動更新 chatlog 裡面的清單，並隨時顯示於每個聊天室窗的左上角，例如說現在已經有三個帳號被註冊，且三個帳號對話框同時開啟，此時新增頁面申辦第四個帳號，並於註冊完後開啟第四個對話框時，所有之前註冊好的帳號會全部出現在第四個對話框的左上角，且其他三個對話框也會即時新增剛剛第四個申辦的帳號，如果其中兩個帳號開啟私聊，server 會將兩個 client 的帳號與對話紀錄 update 在 data 的筆記本裡，讓下次登入時 server 可以直接讀取上次的對話確保聊天紀錄可以延續，開啟私聊時，socket 會分別記錄每個聊天室窗的已讀進度，在開啟視窗的同時，client 會確認你已經讀取全部訊息並回傳告知 server，server 會轉告與你聊天的另一個 client 並在頁面的聊天框框裡顯示 read，表示已讀，最後，logout 的部分，socket 會退出，並將頁面轉回至一開時的登入主畫面，在輸入帳號與密碼後即可再進入聊天室開始聊天。