嵌入式系統報告

組員 1:B02901108 吳省澤
組員 2:B02901108 朱晉德

一、你們在進行這個實驗、學習新技術、撰寫程式碼時，是否曾經遇過問題？

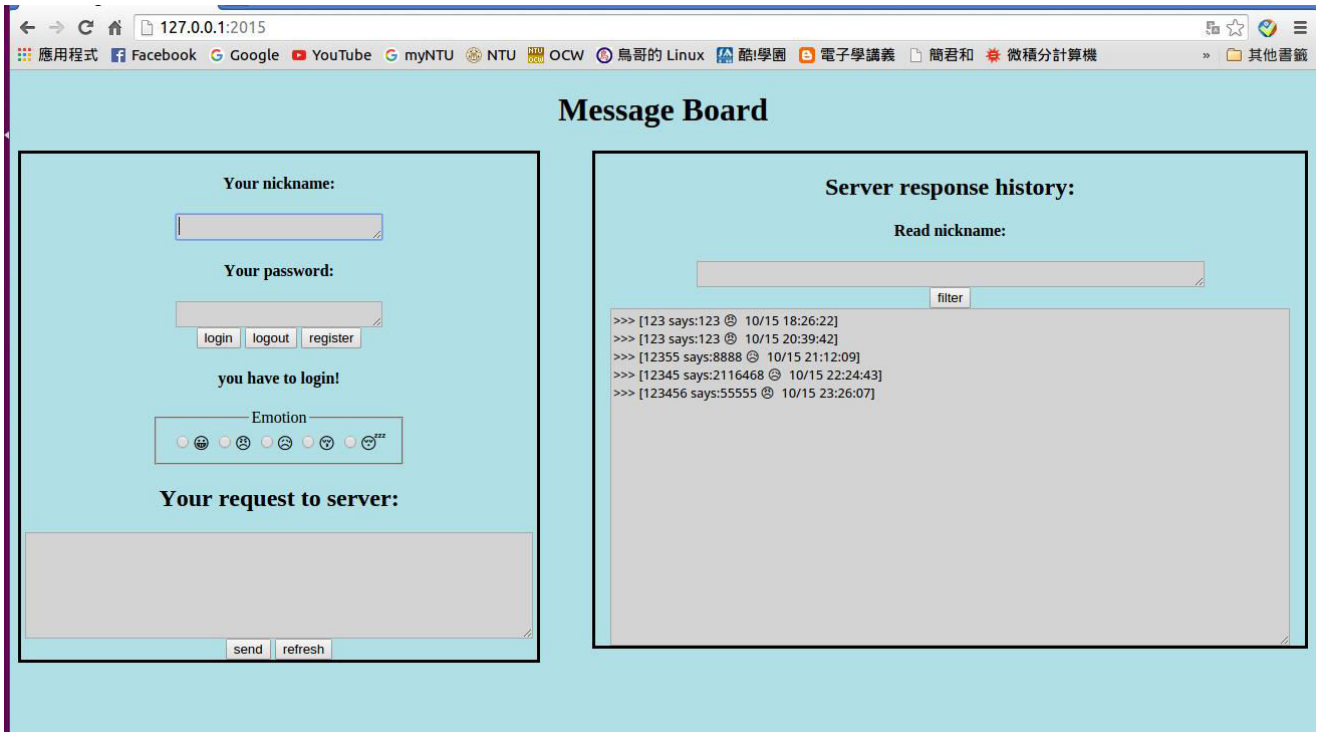
在完成作業的過程中，我們認為我們遇到的最大困難就是不熟悉 javascript 的語法和 nodejs 提供的 API，雖然很多基本的語法都和 C++很類似，但是 javascript 的某些語法觀念(比如說把 C++中每個資料型態當作是物件、function 可以被當作物件和參數來傳遞)，讓我們在理解助教提供的 code 的時候，遇到很多困難，像是 app.js 裡的 set_handler function，裡面其中一個參數就是 function，而這 function 的參數可能又是另一個 function，並且這些 function 在 app.js 和 httpserver.js 之間互相傳遞並定義，另外這份 code 因為是用 nodejs 跑的，所以裡面用到很多 nodejs 提供的 function，也讓我們花了不少時間上 google 查詢這些 function 的用法。

此外 javascript 原本就是一個在網路環境中執行的語言，而且通常是和 html、css 一起執行，所以他也會有許多 C++所沒有的概念，像是 callback、server and client，然後在作業中也牽涉到了 http 協定的觀念，所以我們在了解 fetch API、request 和 response 之間的關係的時候，也花了一些時間。

在這次的作業中我們確實學到很多東西，但我覺得學會怎麼寫 javascript 只是其次，學習如何快速掌握一個沒接觸過的技術才是這個作業帶給我們最棒的經驗，在開始寫這份作業時，完全想像不到我們會如何去理解現有的 code，並如何把附加功能實做出來，一開始我們可能會希望去全盤了解 javascript 和 nodejs API 在開始寫程式，但後來發現這樣太緩慢，所以我們就照著對現有 code 的一知半解和網路上的資料來一步步建構我們認知，並逐步驗證我們的想法，這和閱讀教科書的學習方式感受很不一樣，有種逐步探索的新奇感，此外我們還學到了如何對症下藥，如何提出重要的問題並尋找有用的資料。

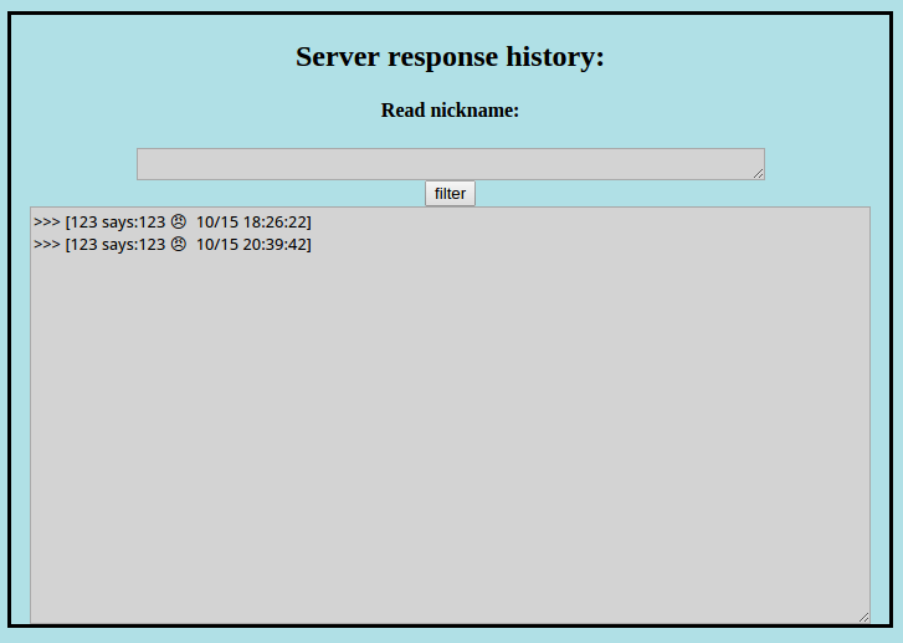
二、請簡述你們的程式要怎麼使用。

1. 先在 terminal 進入 msg 目錄，然後打上指令 node app.js。
2. 開啟瀏覽器，網址列輸入 127.0.0.1:2015



3. 如果要 creat 一個新帳號，可以先打上自己要的 nickname 和 password，然後按下 register。
4. 在留言時要先 login，如果我 login 的 nickname 是 lalala，就會在表情列上方的文字顯示 your id is lalala，如下圖

5. 接著就可以在留言板上留言了，只要選定表情符號，然後打上想要的 message，然後按下 send，你的留言內容就會顯示在留言板上。
6. 我們的留言板還有提供進階功能 refresh 和 filter，如果你想要看某一個 id 所顯示的留言，只要在 Read nickname 下的欄位輸入想要看的 id，然後按下 filter，下圖就是輸入 123 然後按下 filter 後的結果。



7. 因為留言板可能同時被多人使用，所以內容隨時都可能被更新，如果你想看留言版上最新的內容，只要按下 fresh，server 就會傳回資料庫中最新的所有內容。

三、 請嘗試著去描述這份 web application 程式碼實際運作時，它包含的各個元件的角色是什麼，實際上又會發生什麼樣的互動。

我們的程式碼提供了數個 http 介面，POST/echo、POST/submit、POST/read_all、POST/login、POST/register:

1. POST/echo、POST/submit:
   主要負責留言 check 部份，當我們在傳送留言出去之後，系統會幫我們把 nickname、emoji、message 包成 json 的格式，以 POST/echo 或 POST/submit 的形式傳送 request 交由後端處理，後端會檢查 request 的格式，若正確則會將留言內容輸進 data.db，然後傳回 OK 的 response，如果錯誤則會傳回具有錯誤訊息的 response。
2. POST/read_all:
   當我們提出 POST/read_all 的 request，後端會傳回 data.db 的所有資料，交給前端的 code 去處理，主要是用來 load 或是 update 網頁上 messageboard 的內容。

3. POST/login、POST/register:
   主要是在處理登入系統的 id check 部分，當我們想要註冊，按下 register 的時候，我們把 nickname 和 password 包成 json 的格式，以 POST/register 的形式傳送出去，然後後端收到 request 時會搜尋 member.db 的資料，查看是否為已註冊的 id，如果不是，則會將此 nickname 和 password 記錄在 member.db 中，傳回 OK 的 response，反之則不會執行任何動作，只傳回 error 的 response。
   而我們在登入時，nickname 和 password 會以 POST/login 的形式傳送出去，後端會搜尋 member.db，看是否帳號密碼是否相符，若相符則會傳回 OK 的 response，反之則傳回 error 的 response。

而在 main.js，除了 html 中的每個元素，我們還多加了一些變數 user_nickname、othername、read，user_nickname 的作用就是儲存當下 login 的使用者，othername 和 read 則是在 filter 功能中扮演重要的角色，執行 filter 功能的時候，我們會先將 read 設為 1，代表啟動 filter mode，othername 則是儲存我們輸入的 read name，接著提交 POST/read_all，當 filter mode 啟動時，我們不會把 response 的內容全部 load 到留言板上，而是逐一比對每一個留言的 nickname 是否和 othername 相符，以達到 filter 的功能。

在 main.js 中我們也提供很多 function，`data_from_server_callback` 的功能是處理 POST/echo、POST/submit 的 response，如果是 OK 的 response 就交給 function refresh()，refresh()會提出 POST/read_all 來 update 的留言板內容。`load_log_textarea` 會處理 POST/read_all 的 response，根據變數 read 決定是否要篩選 filter 的內容，`id_display` 會處理 POST/login 傳回的 response，如果是 OK 的 response，則會將 id 顯示在 `id_display_elm` 中，也就是表情符號列上方的文字 your id is ……。此外我們還增加了一個新功能 `setInterval( refresh() , 30000)`，在每半分鐘就會 `refresh` 一次留言板的內容。

四、果你們有挑戰加分項目：對於每一個加分項目，請描述你們做了什麼事情。

1. 我們做了一個簡易的登入系統，我們提供了三個按鈕 register、login、logout，register 可以註冊想要的 nickname 和 password，但前提是不能註冊已註冊過的姓名，然後 login 之後再按鈕下方會顯示你的 id，之後你發出的訊息都是以此 id 為身份，如果之後想要以另一個身分登入，就必須先按下 logout，才能繼續登入。
2. 我們增加了一個 filter 的功能，可以讓留言板只顯示出某個 id 發過的留言。
3. 在 send 旁邊有一個 refresh 的按鈕，按下按鈕可以 update 最新的留言板內容