



Practice Homework #7

ChatRoom - A Full-Stack Application

Due : 9pm, Monday, 2022-11-21

作業規定與建議步驟：

1. 這個作業是要大家使用 React/AntDesign/WebSocket/Express/Mongoose 框架，寫出一個簡單的全端應用 "ChatRoom App"。
2. 這個作業「基本要求」的前後端程式碼大多已經貼在上課的講義 (Lecture Note #8 & #9)，所以練習的重點在於看懂我們的 **reference code**、了解整體的架構與設計的邏輯後，再把前後端整合起來 <= 這也是未來大家在做專案時常常會遇到的前後端整合工作。但「進階要求」則會需要改動到前端 **UI/UX** 以及後端 **API + Database**, 歡迎挑戰，但請量力而為。
3. 目錄名稱、package.json 等規範，大致與 HW#6 同。而程式碼則請在詳讀 lecture notes 後，自行 copy and paste 到適當的 .js 檔案中。但，請注意，示範的 ref code 礙於投影片版面的關係，有時候會省略一些細節，因此，請在 copy & paste 以後還是要仔細的檢查一遍，以免遇到 compile error 而束手無策。
4. 執行本 app, 請在 "hw7" 目錄底下，先用 "yarn server" 開啟 server (at Port 4000), 然後在另外一個 terminal, 也是在 "hw7" 目錄底下，用 "yarn start" 開啟 frontend (at Port 3000) 網頁。
5. 這次作業的基本要求功能如下 (請參考封面示意圖)：
 - Sign in 時輸入名字即可開始使用 ChatRoom app (不用有登入密碼檢查機制)，並且跳轉至 ChatRoom 頁面，title 自動設為 sign in 時的名字 (如：Ric's Chat Room // 可忽略英文直接加 's 會不合文法的情形)。如果使用者沒有輸入名字就按送出，應跳出錯誤訊息，禁止其 sign in。
 - 前端重新整理 or 開新開啟視窗時會回到 Sign In 畫面，且 input box 中會自動顯示最後一次 sign in 時的名字 (by local storage)。
 - 在後端使用 MongoDB 儲存所有人的對話紀錄 (請參考講義中建議的 schema: user, message, chatbox)
 - Chat room 請用 Ant Design 的 Tabs 實現。
 - 每個人登入後看到的是空的 chat room, 使用者按下 '+' 會跳出一個 Modal 來輸入想要對話者 (下稱「對話者」、「朋友」 or "friend") 的名字，然後再把 (me, friend) name pair 傳到後端資料庫去檢查，把先前的對話紀錄以 [[name, message]] 的形式傳回來 (如果沒有對話紀錄，就傳回 empty array)，顯示在前端該朋友的 chat box 裡頭。我們只支援 1x1 對話，因此，可以用兩人的名字在排序後作為 chat box 的 unique ID (key) (例如：Ric 跟 Mary 的 chat box 其 key = "Mary_Ric")。
 - 在對話框 (chat box) 中，朋友的留言靠左對齊，而自己的留言靠右對齊，並且按照時間順序顯示，越早的留言排在越上面。
 - 使用者在下方 message input 可輸入想講的話，並且傳到後端資料庫，傳輸成功後，在前端更新最新對話 (在對話框的最下端)。
 - 別人可以在別的視窗使用此 chat room app, 並且選擇跟你對話 (i.e. 在他的 chat room 開啟你的名字的 chat box)。如果你跟他的 chatbox 目前有在你的 chat room 開啟的話，則對方的留言會立即顯示在你跟他的 chat box 上面，否則，就只是存在 DB, 待你下次開啟你們之間的 chat box 才會把對話紀錄載入。
 - 如果在不同的視窗開啟相同名字的 chat room、或者是相同 name pair 的 chat box, 則會在不同的視窗同步開啟相同對話內容，而不會 issue errors。

△ 不管是你 or 朋友的留言，新的留言都是在 chat box 的最下方，並且在超過視窗高度後，每次留言的變化都會自動驅動留言往上捲，確保最後的留言不會被切掉。

- 當 chat room 有多個 chat boxes 的時候，使用者可以點擊上面的 tab label 以切換不同的 chat box (i.e. 改變 Tabs 的 activeKey). 而使用者點擊任何一個 chat box (i.e. 不一定要是 active) 的 'x' 時，會關閉該 chat box (但不會清掉在 DB 的對話紀錄)。關掉 chat box 請適當的更新 activeKey (i.e. 會換成關掉的 chat box 的右邊的那個 chat box, 而如果關掉的是最右邊的 chat box, 則換成關掉的 chat box 的左邊的那個 chat box)
- 自己可以跟自己對話 (i.e. 開啟 chat box 時名字寫自己的名字)，但所有對話訊息會都在右邊。
- 除非電腦系統有限制，否則我們不限制 ChatRoom 或是 ChatBox 的個數。

△ 當以上的動作發生時，用 “displayStatus” function 在對話框的上端適當地顯示出通知訊息。

6. 這次作業的進階要求功能建議如下 (有一定難度，請量力而為，完成兩項即可被判定為 Perfect)：

- 在 Sign In 頁面加上註冊與檢查密碼的功能：
 - ✓ 註冊資訊以及密碼是否正確應在後端檢查
 - ✓ 密碼請不要用明碼傳到後端
 - ✓ 與基本要求同，下次再進入 Sign In 頁面時應直接帶入上次的登入名字 (but not password)
 - ✓ 如果沒有輸入名字或是密碼，應跳出適當的錯誤訊息
- 將基本要求中只能一對一對話的限制放寬，讓一個 chat box 也可以容納多人來聊天 (i.e. 多於兩人的群聊)：
 - ✓ 為了簡化起見，可以限制在剛開啟多人對話的 chat box 的時候就得要決定參與者。事後如果要增減參與者，應該就是開啟不同的 chat box.
 - ✓ 其他人的留言都靠左放，但應加上留言者的名字，以示區別。
 - ✓ 需要重新定義 DB schema
- 處理各種錯誤，並給予適當的提示訊息，至少包括：
 - ✓ 未輸入名字就按 “Sign In”
 - ✓ 未開啟任何 ChatBox 就送出訊息
 - ✓ 創建 ChatBox 時沒有輸入對話者名字，或者是與對話者之 ChatBox 已開啟 (i.e. 不可重複開啟同一個 ChatBox)
 - ✓ 處理輸入過長時 “Line Wrapping” 的問題
 - ✓ 後端沒有正確回應或是開啟
- 如果有訊息來到不是 active 的 chat box, 則在該 chat box 上方顯示紅色的圈圈，並累計未讀留言數。

7. 如果你 hw7 資料夾中有些檔案是沒有需要 push 到 GitHub 上去的，請記得把他們的檔名加到 .gitignore 去。**Note: 請不要把 node_modules 以及 .env 上傳到 GitHub**

8. 以上說明，如有不夠詳盡之處，請自行判斷與定義，不用問我們「這樣可不可以」、「這個要不要做」之類的問題，再次聲明 practice homework 不會有詳細的規格，「定義規格」也是 practice 的一部分，但為了讓 reviewer 能夠正確地使用你的 app, 請儘可能地在 README.md 中提供必要的說明。

繳交

請在 deadline (9pm, Monday, 2022-11-21) 之前把你的作業 (i.e. hw7 整個目錄) push 到你 GitHub 上 wp1111 這個 repo 上面 (Note: 我們不接受遲交哦!)。你應該會用到的指令有 `git add hw7` (i.e. 把整個 hw7 一次加進去), `git commit -m "你想留的 comment"`, 以及 `git push`. 你也可以用 `git status` 來檢查一下檔案是否都已經成功 push 到 GitHub 上面, 或者是登入 GitHub 去檢查。

如有任何的問題, 請到 FB 社團的 "HW#7 Q&As" 貼文底下發問, 或者是寫信到老師/助教信箱: eewebprogramming@googlegroups.com。

Reviewer 注意事項

批改時請使用自己的 DB, 換句話說, 請把自己的 .env copy 到 clone 下來被 review 的人的目錄中。不過請注意, 為了避免因為彼此的 DB schema 不同而造成混用 DB 所發生的 crash, 請到 MongoDB 管理介面把該 collection 清掉後再重新執行 app, 請勿未檢查就把對方的作業評為屍體。