

My Chat Room

Sign In

Ric's Chat Room

Clear

Ric Hello

Ric How are you doing?

Mary Yes?

Mary So?

Ric What are you going to do today?

Mary Don't know

Ric Really?

Mary Yeah...

Send

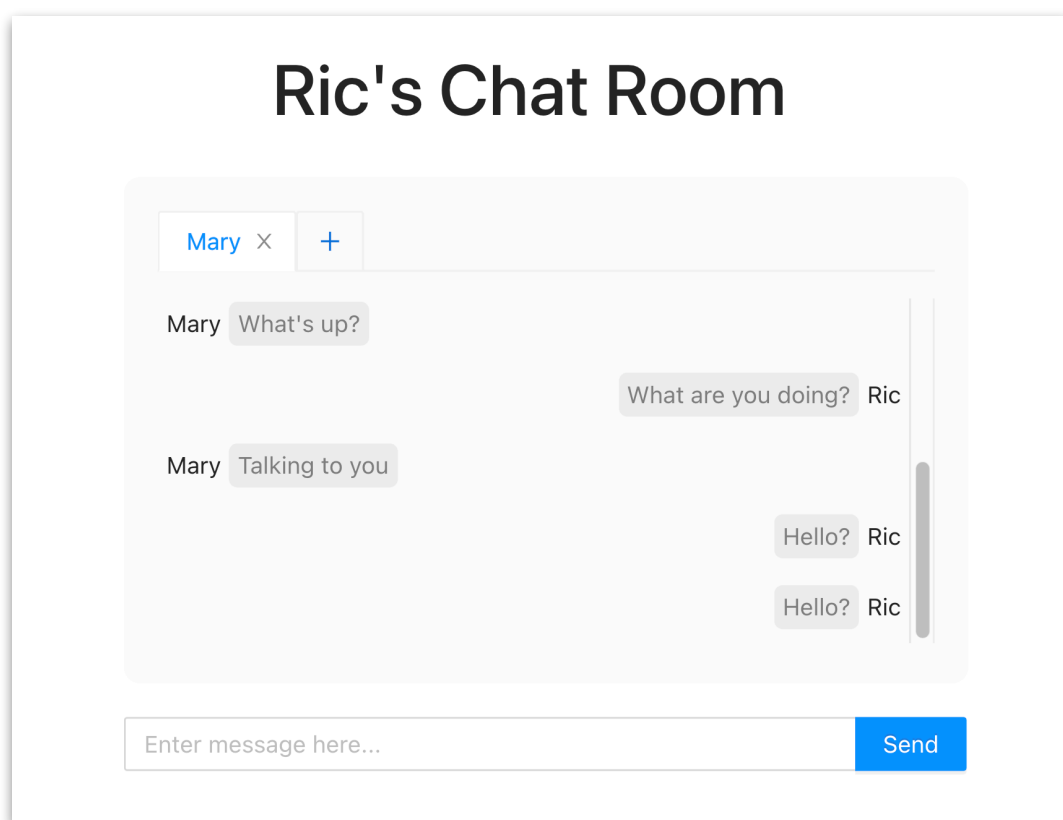
Practice Homework #8

ChatRoom - A Full-Stack Application

Due : 9pm, Monday, 2021-12-13

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」的前後端程式碼大都已經貼在上課的講義 (**Lecture Note #7 & #8 的前半部**)，所以練習的重點是在如何看懂我們的 **reference code** 後把前後端整合起來 <= 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#7 同。而程式碼則請再詳讀 lecture notes 後，自行 copy and paste 到適當的 .js 檔案中。
4. 執行本 app, 請在 “hw8” 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 “hw8” 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，並且跳轉至 ChatRoom 頁面，title 自動設為 sign in 時的名字。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 (i.e. 存到同一個 messages array)，每個人登入後會自動的從此 DB 把目前所有的對話紀錄 load 到前端，並且顯示出來。
 - 可同時開啟不同的 browser windows 測試多人群聊的情境 — 每個人的輸入對話會即時廣播到所有開啟的視窗。
 - 任何人按下 ChatRoom 視窗的 Title 旁邊的 “Clear” 按鈕會立即清除掉 DB 目前的對話紀錄，而所有人的對話視窗也會因此而同時被清空。
 - 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 (請參考下頁示意圖。有一定難度，請量力而為)：
 - 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
 - 將基本要求的「集體聊天室」改寫成可以「一對一對話的真。聊天室」(可不用保留集體聊天室功能):
 - ✓ 在一個瀏覽器視窗 sign in 進去 app 後，我們稱之為開啟一個 ChatRoom。而一個 ChatRoom 可以讓登入者 (i.e. 自己) 跟不同人開啟個別的 ChatBox 來進行一對一對話
 - ✓ 每個 ChatBox 就是一個 TabPane，上面標籤顯示對方對話者的名字
 - ✓ 剛登入時 ChatRoom 應為空的，即沒有任何的 ChatBox
 - ✓ 此外，除非電腦系統有限制，否則我們不限制 ChatRoom 或是 ChatBox 的個數。



- 在 Tabs 列表最右邊按下 '+' 後會跳出小視窗詢問欲新增對話對象的名字，待輸入對話對象的名字後，即新增自己與此人的對話視窗 (ChatBox)：
 - ✓ 如為第一次與此人對話，則會在 DB 產生一個新的 ChatBox, 否則，之前與這個人的對話記錄會自動從 DB 載入並呈現在此對話框裡
 - ✓ 對方的對話靠左邊 (<名字> <訊息>)，自己的回話靠右邊 (<訊息> <名字>)
 - ✓ 如果對話的兩個人的 ChatRoom 都有打開此彼此的 ChatBox (e.g. Ric to Mary, and Mary to Ric)，則任何一方輸入對話訊息後會在兩個人的視窗同時更新（但一個人是發話者，另外一個人是收話者），且會同時更新至 DB 的兩人對話記錄中
- 針對某一個已開啟的 ChatBox (TabPane) 按下 '-' 後可以關閉此對話視窗（但不會把對話記錄刪掉），但如果透過上一步再開啟，則對話記錄應該還是要原封不動的存在。
- 可點選 ChatRoom 上方的 Tab 切換 ChatBox。
- 同一個人在不同瀏覽器視窗執行此 App 所開啟的是獨立的 ChatRoom, 即各自 ChatRoom 可以有各自開啟/新增的 ChatBoxes。
- 處理輸入過長時 "Line Wrapping" 的問題
- 自己可以跟自己對話 (i.e. 開啟 ChatBox 時名字寫自己的名字)，但所有對話訊息會都在右邊。
- 盡可能 cover 各種錯誤操作，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)

- (Optional) 使用 React Router，讓切換不同的 ChatBox 同時也會切換不同的 URL (i.e. Client-Side Routing)，因此可以使用瀏覽器的「回上一頁/到下一頁」來切換對話視窗
 - (Optional) 使用 useContext 來管理一些適合搬到 global storage 的 states
 - (Optional) 為了效能起見，只更新目前顯示的 ChatBox 的對話訊息，而其他的 ChatBox 則等到被 bring up 時再更新對話訊息即可
7. 如果你 hw8 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, 2021-12-13) 之前把你的作業 (i.e. hw8 整個目錄) push 到你 GitHub 上 wp1101 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 hw8 (i.e. 把整個 hw8 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#8 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，如果因為彼此的 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。
