

ScoreCard DB

ClearAdd☐ Name ☒ SubjectQuery

Practice Homework #6

React + Axios + Express + Mongoose

Due : 9pm, Thursday, 2021-05-20

WEB PROGRAMMING 109-2

作業規定與建議步驟：

1. 這個作業是要大家使用 React/Axios/Express/Mongoose 框架，寫出一個簡單的全端應用 “Score Card DB”。
 2. 為了讓同學有機會自己挑戰獨立完成全端應用，這個作業的 **ref code** 將會稍晚 (下週二) 釋出。大家可以選擇自己寫 (只要符合下面的規範即可)，或者是在我們提供的 **ref code** 上面寫。
 3. 請先到 wp1092 目錄下，創建一個 hw6 的子目錄，並在其底下建立 “own” 目錄：

```
cd wp1092
mkdir hw6
cd hw6
mkdir own
```
 4. 到 own 目錄底下，用 “yarn init -y” 建立一個專案，並在 “package.json” 裡頭加入這幾行：

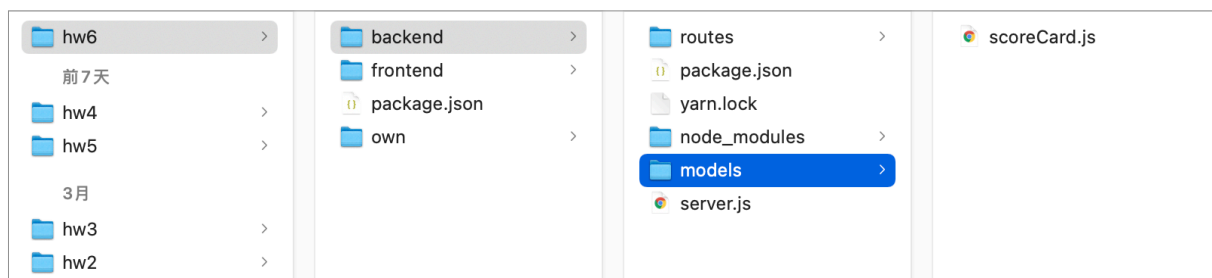
```
"scripts": {
  "start": "cd frontend && yarn start",
  "server": "cd backend && yarn server"
}
```
 5. 同樣在 own 目錄底下，用 “create-react-app” 建立 “frontend” 目錄：

```
create-react-app frontend
```
 6. 同樣在 own 目錄底下，建立 “backend” 目錄：

```
mkdir backend
```
 7. 到 “backend” 目錄，用 “yarn init -y” 建立一個專案，並在 “package.json” 裡頭加入這幾行：

```
"type": "module",
"scripts": {
  "server": "nodemon server.js"
}, backend
```
 8. 安裝一些需要用到的套件 (如果在執行 app 時遇到缺少套件的 error, 請自行增加)：

```
yarn add mongoose dotenv-defaults --save
yarn add -D nodemon
```
 9. 請參考 Lecture Note #8 的範例, 將 server 的入口檔案寫在 “server.js” 裡頭，然後將 MongoDB schema，以及 backend 的各 endpoint APIs，分別寫在 “models” 以及 “routes” 子目錄。
 10. 請將前端應用寫在 frontend 目錄，請盡量 follow 以前作業的檔案與目錄安排。
 11. (Optional) 幾天後，如果你要參考 ref code, 請至 Ceiba 公布欄下載 “hw6.tgz”。解壓縮後，你應該會看到部分的 frontend 以及 backend 的 reference code。請注意：由於解壓縮可能會蓋掉你原先同名的目錄或是檔案，建議你先在別的目錄解壓縮，再手動一一把你需要的檔案 copy 或者是 merge 過來。
 12. 這次作業的檔案目錄長得像這樣：
-



13. 執行本 app, 請在 “own” 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 “own” 目錄底下, 用 “yarn start” 開啟 frontend ScordCard 網頁。

14. 這次作業的基本要求功能如下 (請參考封面示意圖) :

- 當 "Clear" 按鈕被按下去後, 會送出 DELETE request 把 DB 清空, 然後在下方空白處印出 “Database cleared”。
- 當 "Name", "Subject" 以及 "Score" 三個輸入框都被輸入值之後, 按下 "Add" 按鈕, 會把資料寫到 DB :
 - ✓ 如果 {Name, Subject} paired value 已經存在 DB, 則取代原先在 DB 的這筆資料, 然後在下方空白處印出 "Updating (Name, Subject, Score)" // Name, Subject, Score 請代換成輸入的值
 - ✓ 否則, 在 DB 新增一筆資料, 就印出 "Adding (Name, Subject, Score)"。
- 當 Query string 不為空字串的情況下, 當 Query 按鈕按下去後, 會根據 Name/Subject 二選一的 radio button 選擇 Name or Subject 作為 query 條件, 然後根據輸入 Query String 的內容去 DB 把符合條件的資料搜尋回來, 在下方空白處印出搜尋出來的結果 (一筆一行, 格式不限)。如果找不到, 就印出 “QueryType (QueryString) not found!” // 例如: “Name (Ric) not found!”
- 以上所有動作在印出訊息/資料時會先把原先空白處的顯示清掉, 如果在 query 時有任何的錯誤訊息, 也是一樣印在這邊。
- 請注意, 封面的 UI 只是示意圖, 大家不管是把它刻得很陽春或是很 fancy, 都不在這次作業的基本要求裡頭。不過, ref code 會使用第三方的 UI 套件, 屆時也歡迎大家參考。

15. 這次作業的進階要求功能建議如下 :

- 新增更複雜的搜尋、修改功能。例如: 可以同時下 Name、Subject、Score 等多重搜尋條件, 搜尋條件可以有不同的邏輯 (AND, OR, compare (如: Score >= 60))... 等。
- 搜尋回來的資料一次只顯示一部分 (例如: 前十筆), 然後下方提供如: 下一頁、第幾筆等功能。
- 可以根據條件排序搜尋回來的資料 (可以在前端完成就好)
- 任何 UI/UX 的改進 (請在 README.md 中說明)

16. 如果你 hw6 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。

17. 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, Thursday, 2021-05-20) 之前把你的作業 (i.e. hw6 整個目錄) push 到你 GitHub 上 wp1092 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 hw6` (i.e. 把整個 hw6 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。請注意, 你不用建立 "review" 這個資料夾, 助教會在指派 review 作業時再幫你加進去你的 repo。

如有任何的問題, 請到 FB 社團的 "HW#5 Q&As" 貼文底下發問, 或者是寫信到老師/助教信箱: eewebprogramming@googlegroups.com。
