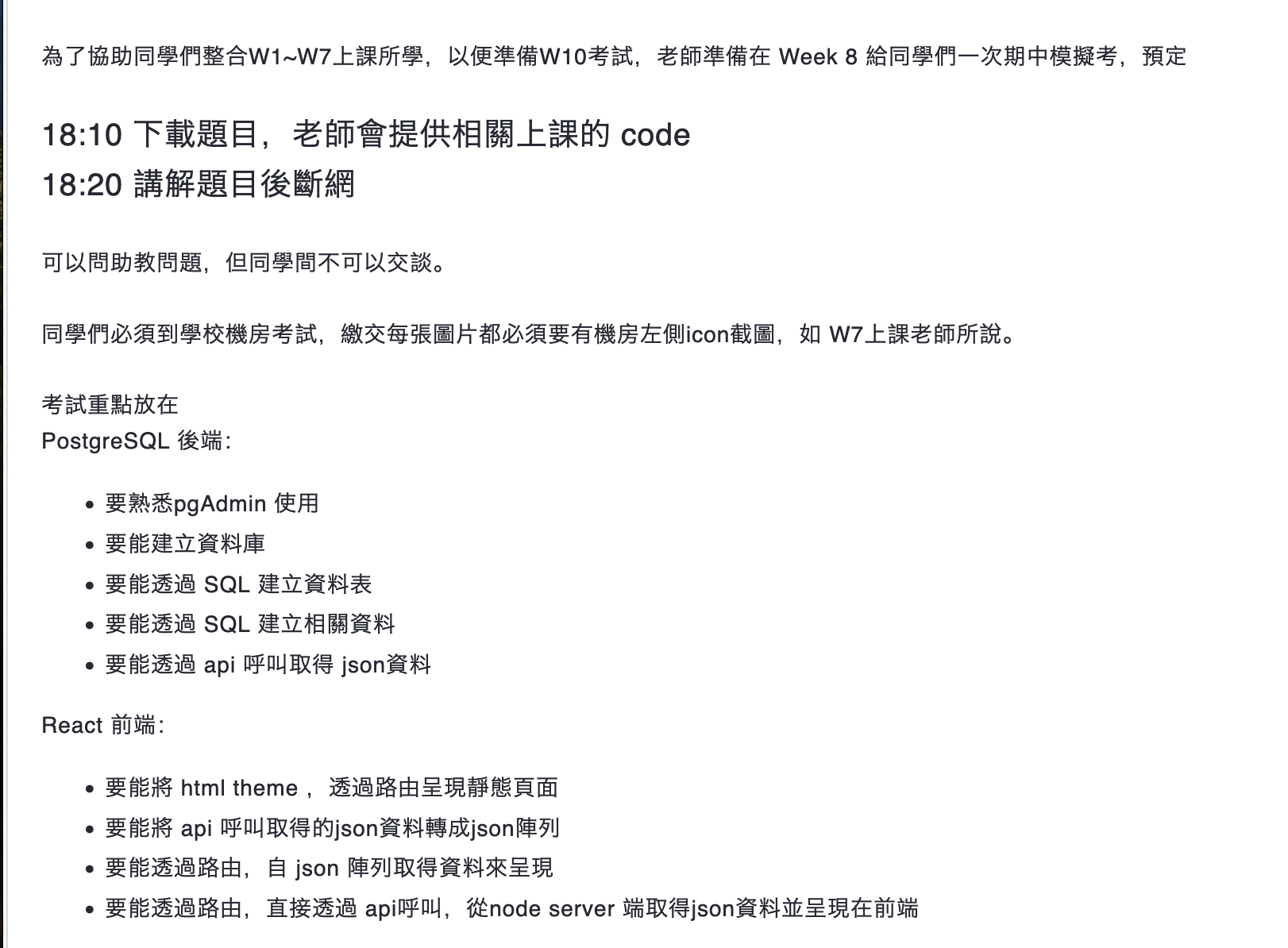
# Midterm Preparation 期中測驗準備

這份文件是為了給自己做複習，順便分享給你做做參考。有問題可以一起討論！

本期目標參考老師於Iclass上的敘述所使用，有可能與實際Code不符。還請諒解哈！

## Requirement



**思路**

先創立資料庫，設置好後使用ExpressJS與資料庫進行對接。

完成測試後，架設React前端畫面。

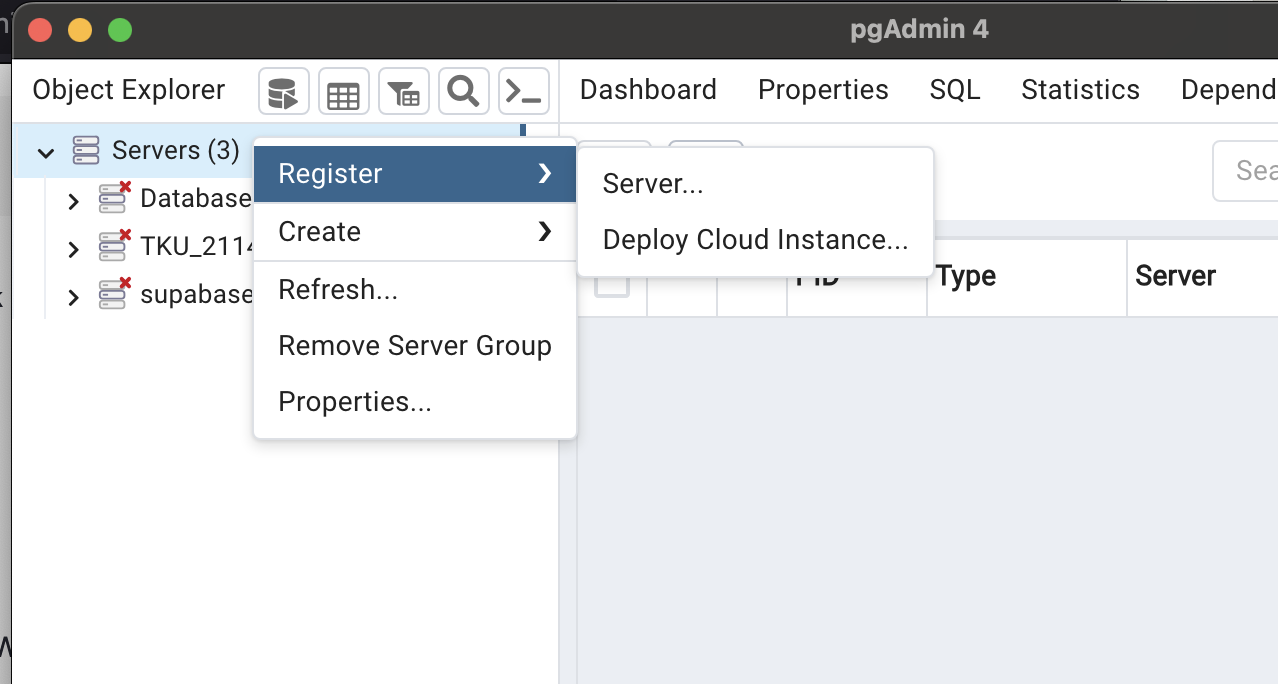
想法有了，咱們開始吧！

# 後端設計

# **Postgres App使用**

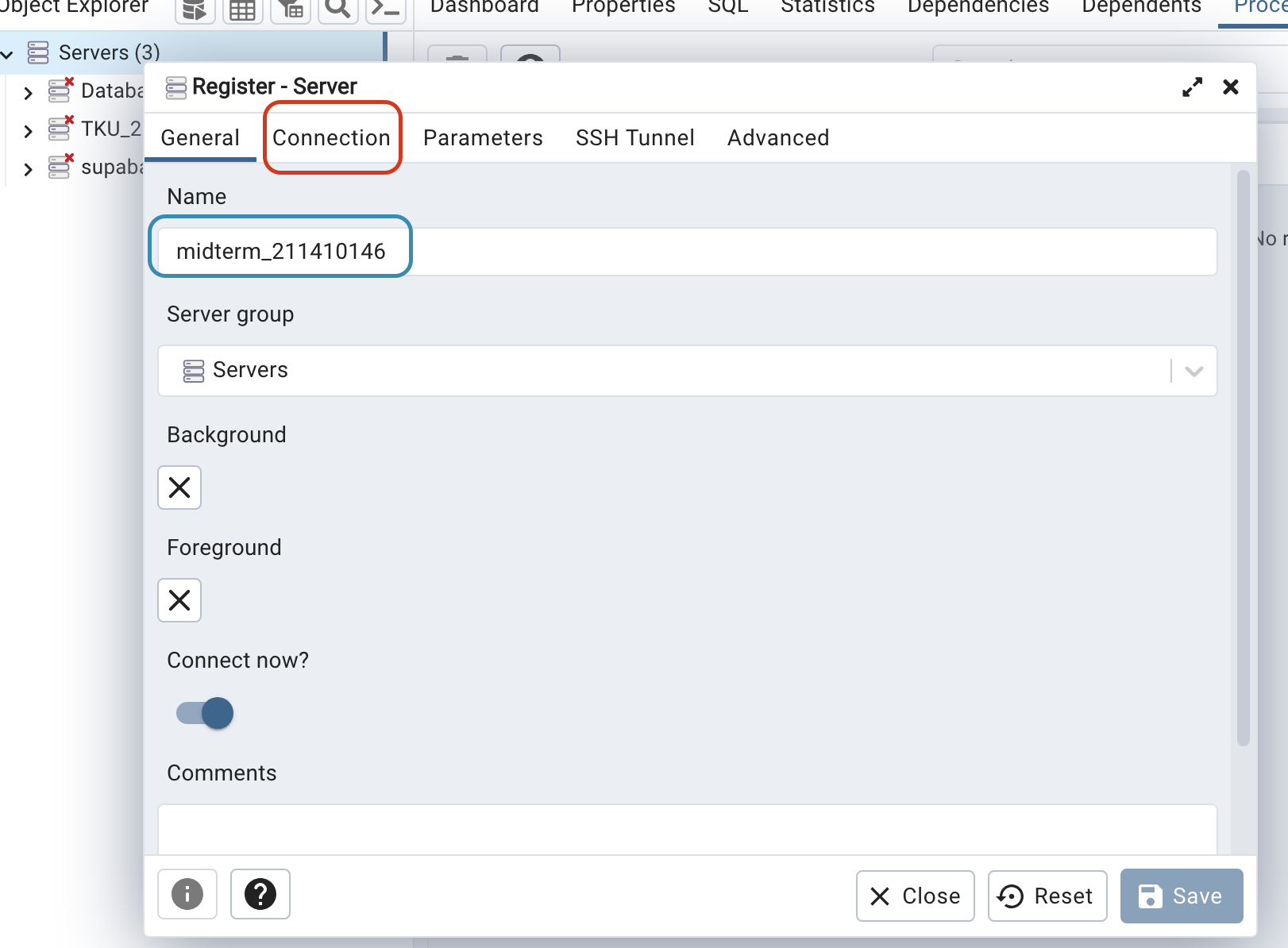
請注意，我的版本是使用Mac作為開發環境，可能創建過程會略為不同。但理論上應該相同。如果有問題直接跟我說，我再補一版Windows。

這是PG Admin，也是一個GUI介面操作Postgres的資料庫。我們就從這裡開始吧！



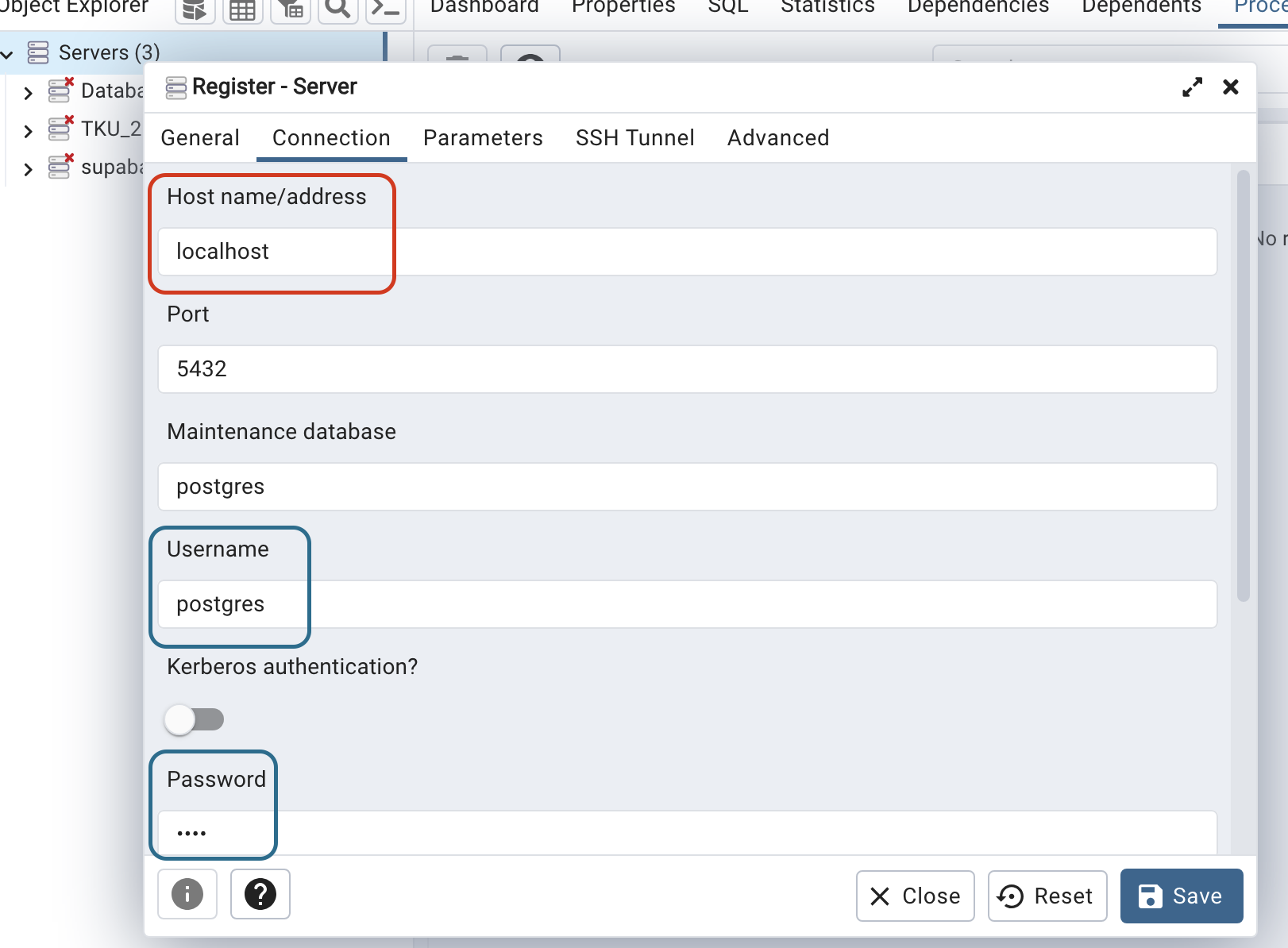
首先對著**Servers**按右鍵，進入**Register**，進入**Server**。

進來後先取名，通常老師會使用學號作為區隔，請注意老師要求。



然後點進去Connection。

點進來會看見一份琳琅滿目，主要需要關注的在這框起來的就好。

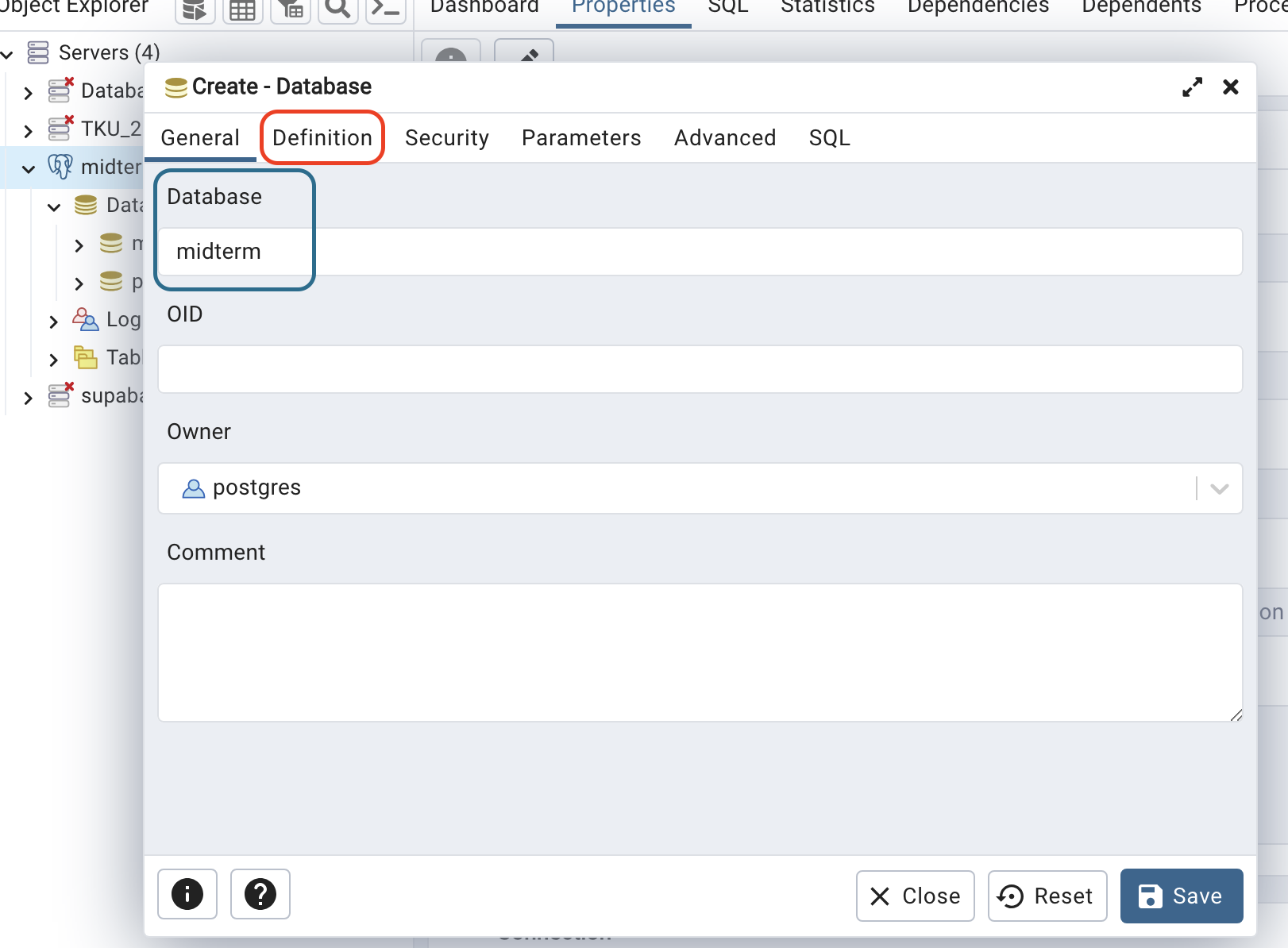


**Host Name**那輸入**localhost**或**127.0.0.1**。無線網路老師告訴我們那是代表本機的意思，意味著我們要在這台電腦上設置Postgres Database。

**Username**的部分請記好，Express的時候有大用；**Password**我忘了他會不會預設，如果沒有那就好，如果有請記得密碼。我目前是隨便設**0000**。

緊接著就來到創建**Database**跟**Table**了。

右鍵點你剛才創立好的Server，然後選擇Create，選擇Database。名字設好理論上就ok；如果遇到一個Local Provider的問題，點選Definition，選擇Local Provider後換一個即可。

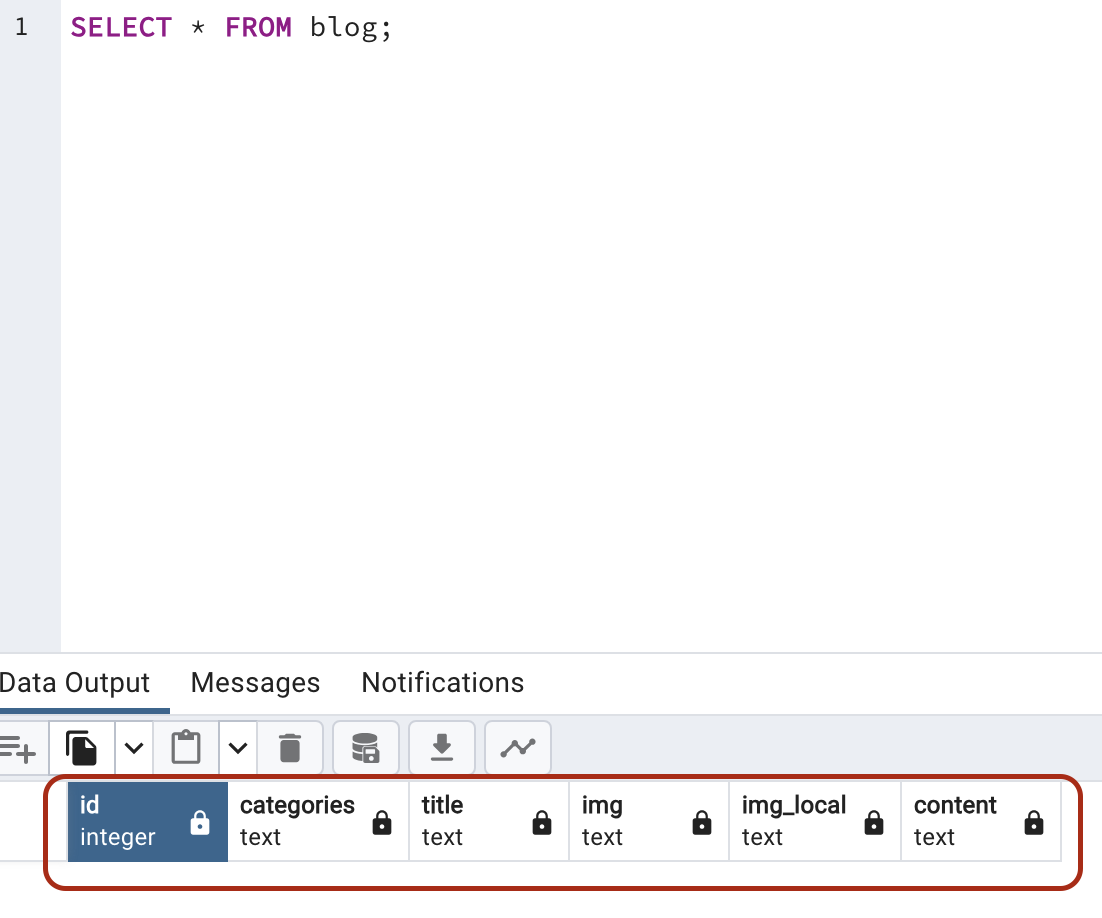


快速進入Table的部分。

在創立的Database那右鍵，選取Query tool。

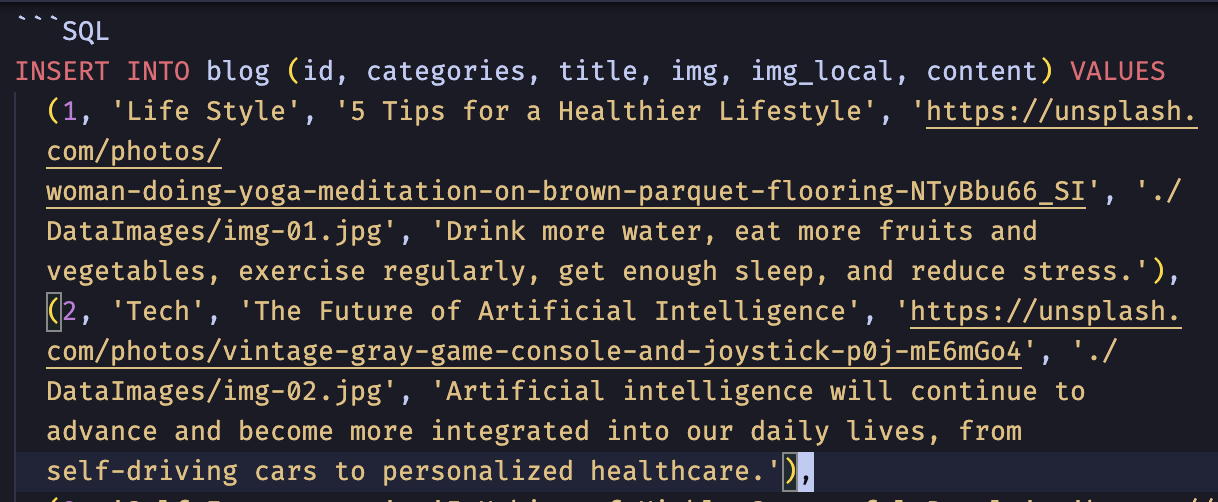
由於我預先使用CoPilot產生了好幾筆測資，格式也仿造老師的格式。

我把上面的SQL Command輸入進去Query tool內，建立了一個Schema。我懶得算字數，所以不選擇varchar或char，你可以有其他選擇。

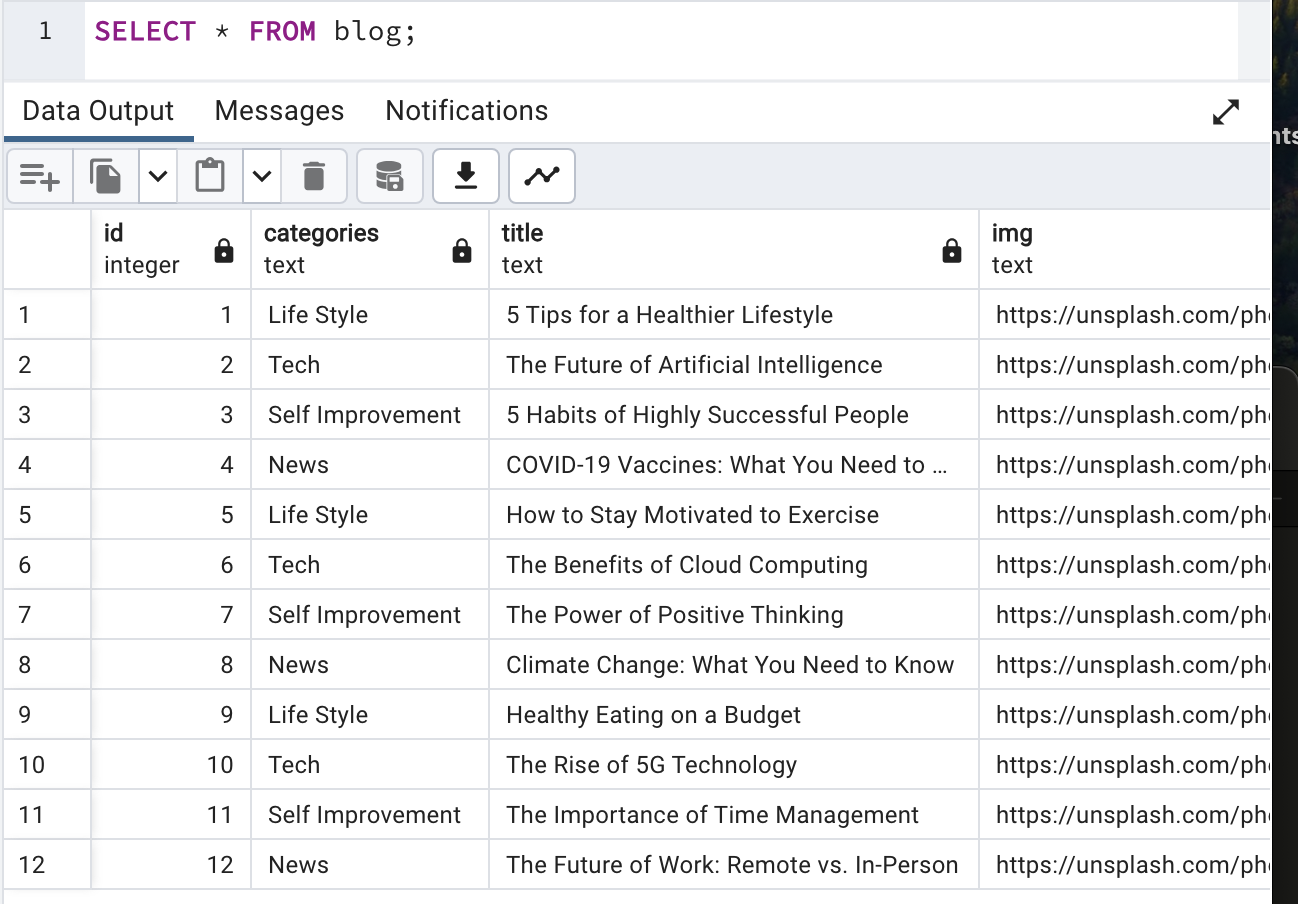


接下來用SQL指令輸入剛才產生的測資。

這裡只將幾筆資料塞進去，剩下的我也開了一個Markdown，給你參考一下！



內容塞進去了、資料庫健全了。接下來該著手進行Express的部分了。



## Express JS的路由

在Express JS，我們要建立起路由，讓我們能有一個安全的方式對資料庫進行操作。

首先進到終端機輸入

npm install –g express-generator

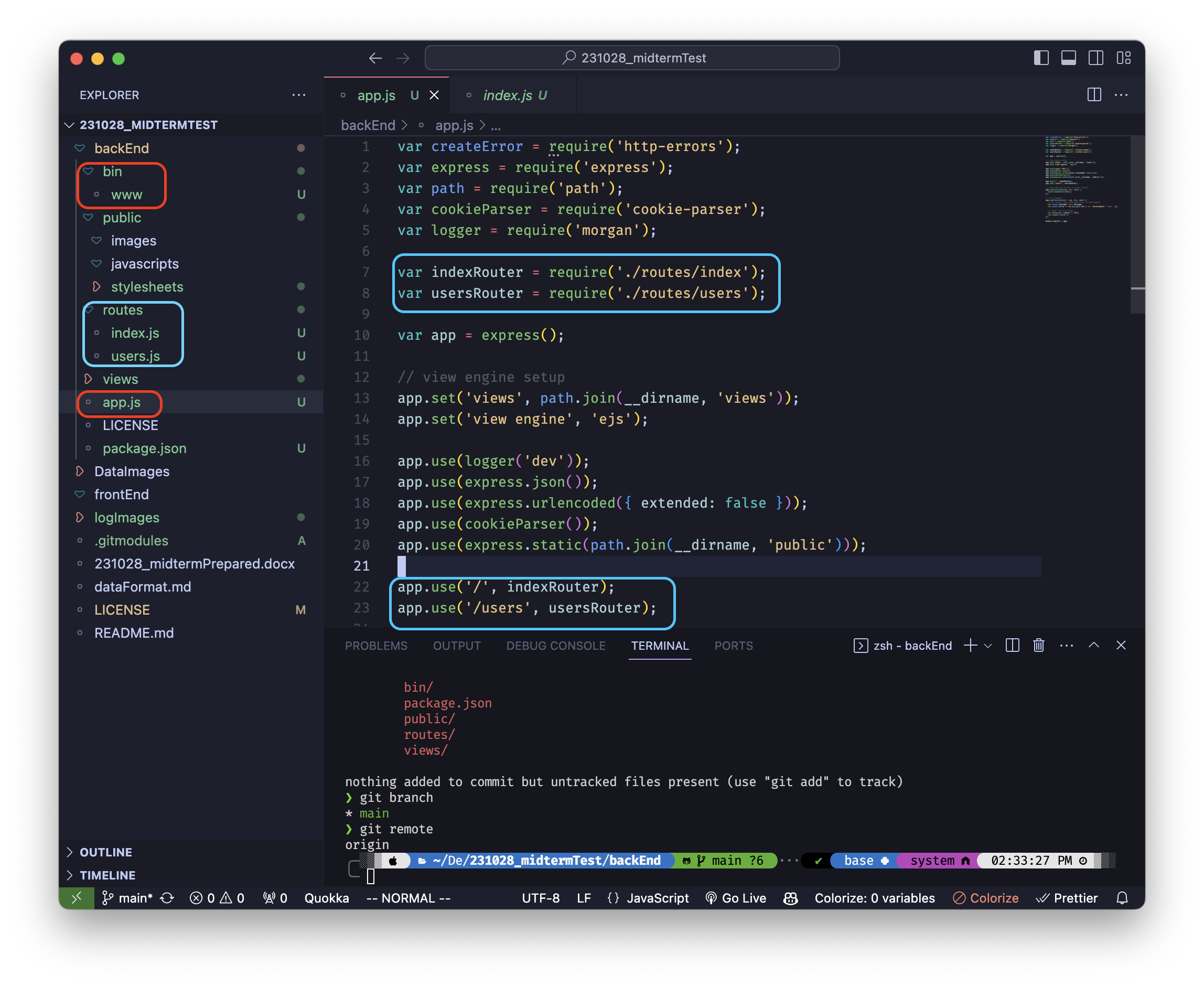
這是老師課上使用的方式，用來幫你設訂預設的程式框架出來。我們接下來輸入

express --view=ejs backEnd

backEnd 是我要將後端程式放置的位置，可以替換成其他路徑（如當前資料夾 . ）。

然後我們就得到以下內容。

紅色框是掌管程式運行的地方，我們不太需要動。



藍色框則是要關注的地方。Routes資料夾內的檔案是express公司為了方便你管理而放置的地方，但他**沒有**特別功能。App.js內的藍色框則是讓我們能在輸入網址時進行溝通的橋樑。如我們輸入localhost/他會回傳indexRouter裡面的內容。

我們接下來要做三個部分：創建新檔案、模組導入app.js、介接資料庫。

我在route資料夾下創立了一個card.js檔，並且在根目錄（最外層）的app.js中加了這段。



我要求程式當伺服器接收到關於**localhost/cards**請求時，會執行cardRouter中（包含底下分支）的所有動作。所以如果輸入localhost/cards/title/MarxIsHandsome，那一樣是歸類在cardRouter的業務範疇內。

來到card.js中，我先寫了一段Code來測試程式運行正常。router擔任起這個頁面底下分類、回傳的動作。也是回傳的內容。（有回傳才會動作！）



router.get（紅色箭頭）意思是獲取這個位置的內容（詳情請見[這裡](https://developer.mozilla.org/zh-TW/docs/Web/HTTP/Methods)），請求對象是“/”這個地方。我們目標是在這個位置中將所有的卡片以JSON格式給展示出來，因此我們在Response的地方寫了以JSON方式回傳（藍色箭頭）。