【1308】

資料庫系統

公司財務與合作資訊資料庫

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 系級 | 座號 | 學號 | 姓名 | 備註 |
| 資訊二乙 | 69 | D0782956 | 平祖安 | 組長 |
| 資訊二乙 | 58 | D0746384 | 李筱文 | 組員 |
| 資訊二乙 | 57 | D0746353 | 岳靈 | 組員 |

1.主題:

公司財務與合作資訊資料庫

2.目的:

讓所有的合作清楚，且提供上司的檢核，方便進行財務報表，

提升公司確認工作進度的效率，而可提供短暫的員工證，讓合作廠商一同協作，根據權限來規範使用者可看到的內容，保障項目的機密性，促進未來的合作和諧度。

3.系統使用者: ※根據應用層面，可能所有合作皆可使用

* 管理員
* 老闆
* 經理
* 員工
* 外部使用者

4.需求功能：

* 所有使用者:
* 登入登出
* 管理員
* 增加、減少和更新使用者帳戶
* 老闆
* 查詢經理負責專案的資訊，有篩選功能
* 經理
* 查詢自己負責專案的資訊，有篩選功能
* 新增和更新專案及員工，且更新時檢查是否為帳戶所負責的專案
* 員工
* 查詢自己負責專案的資訊，有篩選功能
* 新增專案的檢測日期，且新增時檢查帳戶是否為專案的處理人
* 外部人員
* 可了解公司理念和財務情況

5.輸入和輸出

1.輸入帳戶及密碼

成功 導向至帳戶對應介面

失敗 顯示登入錯誤訊息

2.輸入欲新增、刪除或更新之帳戶(**管理者**)

1. 前端格式錯誤 顯示格式錯誤

前端格式正確 顯示表單成功送出

II.後端資料定義正確 顯示操作成功

後端資料定義錯誤 顯示操作失敗

3、4為進行查詢時的條件

3.輸入查詢篩選條件(**老闆、經理、員工**)

4.輸入顯示篩選條件(**老闆、經理、員工**)

* 3為查詢的條件，4為顯示時要出現那些欄位

進行資料庫查詢 顯示符合條件的結果

(篩選條件與顯示結果會因帳戶有所差異)

5.點擊詳細資訊按鈕(**老闆、經理、員工**)

點擊 打開提示框顯示詳細資料

6.新增一筆專案的資訊和專案員工(**經理**)

1. 前端表單輸入不完整 顯示錯誤訊息

前端表單輸入完整 顯示表單成功送出

II.後端資料定義正確 顯示操作成功

後端資料定義錯誤 顯示操作失敗

7.更新專案資訊及替換專案員工(**經理**)

1. 前端表單輸入不完整 顯示錯誤訊息

前端表單輸入完整 顯示表單成功送出

II.後端資料定義正確 顯示操作成功

後端資料定義錯誤 顯示操作失敗

8.新增專案的檢測日期(**員工**)

專案屬於此員工負責 顯示操作成功

專案並非此員工負責 顯示操作失敗

9.顯示公司資訊(僅輸出)

資料限制

EmployeeAccount

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 欄位名稱 | 欄位型態 | 是否唯一 | 能否NULL | 備註(PK,FK) |
| EmployeeNO | Email | YES | NO | PK |
| FirstName | Varchar(10) | NO | NO |  |
| LastName | Varchar(10) | NO | NO |  |
| PhoneNumber | Decimal | YES | YES |  |
| Authority | Char | NO | YES |  |
| FlagLead | boolean | NO | NO |  |

CooperateItem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 欄位名稱 | 欄位型態 | 是否唯一 | 能否NULL | 備註(PK,FK) |
| ProductName | Varchar(10) | YES | NO | PK |
| Price | Int | NO | YES |  |
| SignDate | Date | NO | NO |  |
| StartDate | Date | NO | YES |  |
| CompleteDate | Date | NO | YES |  |
| Content | text | NO | YES |  |

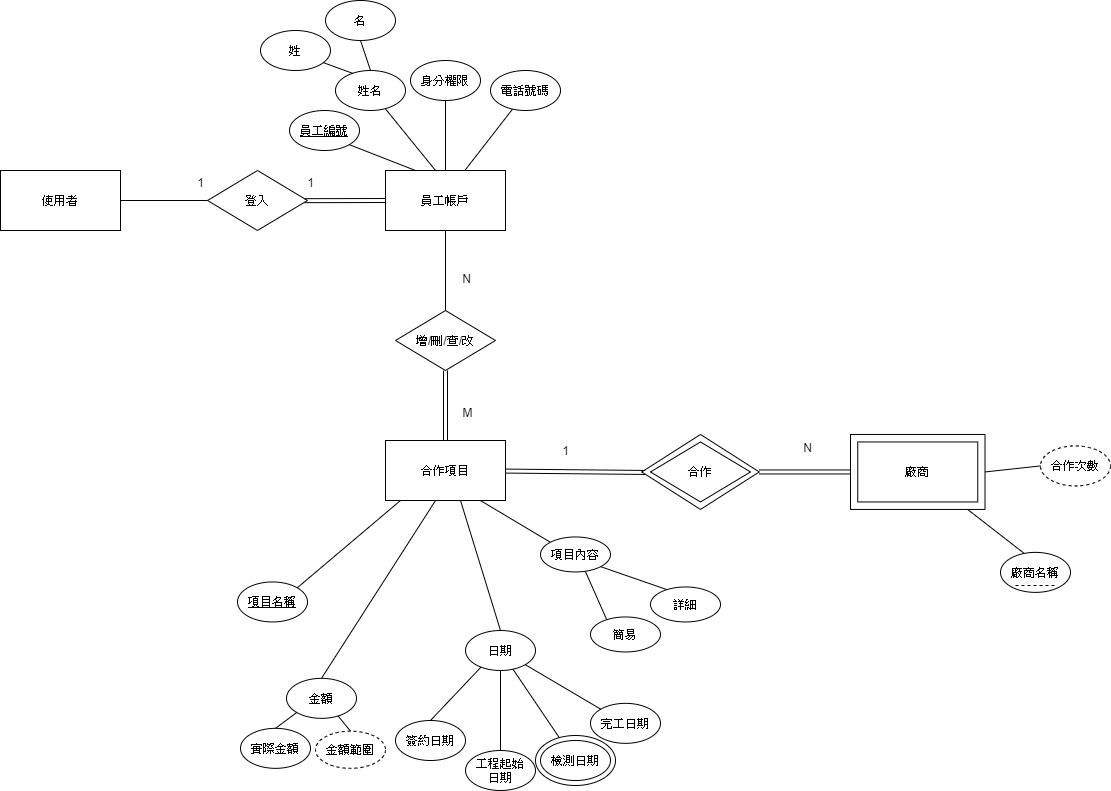
RelationDeal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 欄位名稱 | 欄位型態 | 是否唯一 | 能否NULL | 備註(PK,FK) |
| EmployeeNO | Email | NO | NO | FK (EmployeeAccount) |
| ProductName | Varchar(10) | NO | NO | FK  (CooperateItem) |

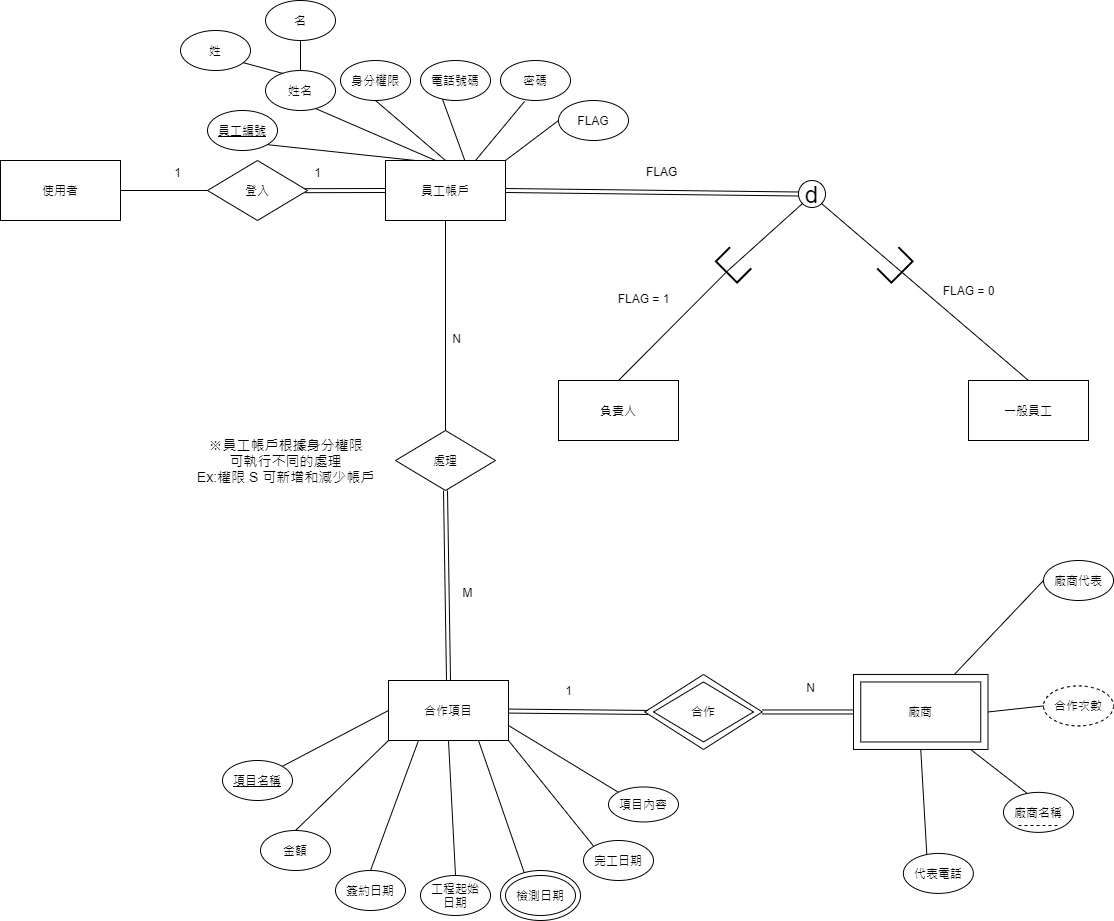
MultivalueDectDate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 欄位名稱 | 欄位型態 | 是否唯一 | 能否NULL | 備註(PK,FK) |
| DetectDate | Date | NO | NO |  |
| ProductName | Varchar(10) | NO | NO | FK  (CooperateItem) |

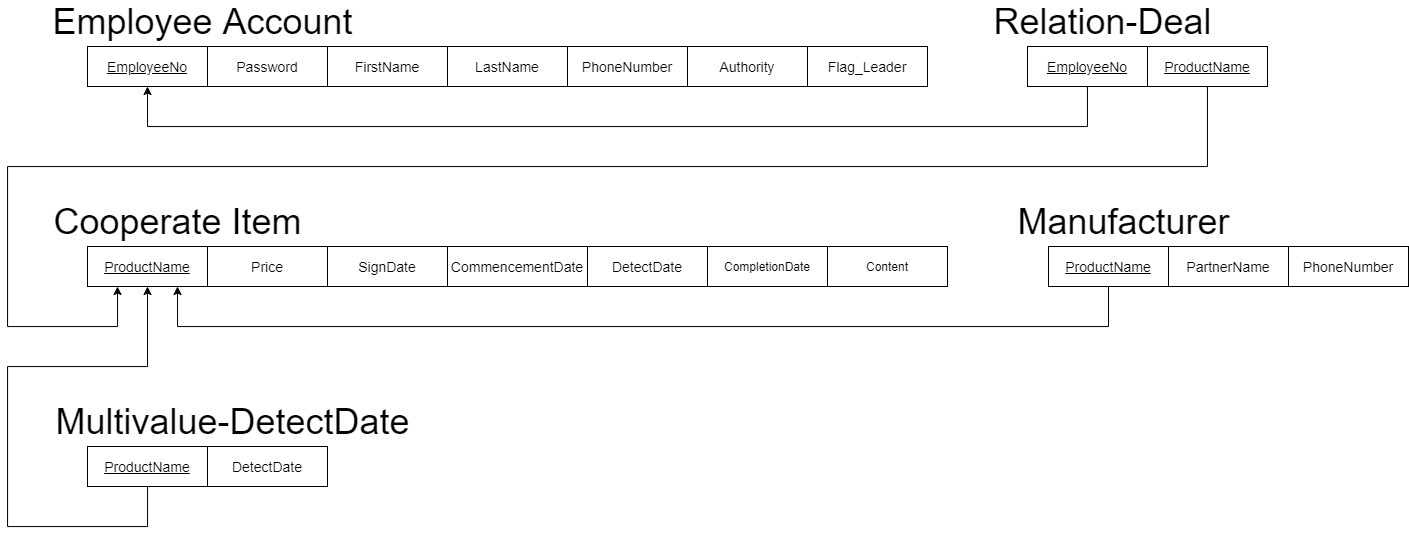
6. Entity-Relationship diagram:



7.EER-diagram



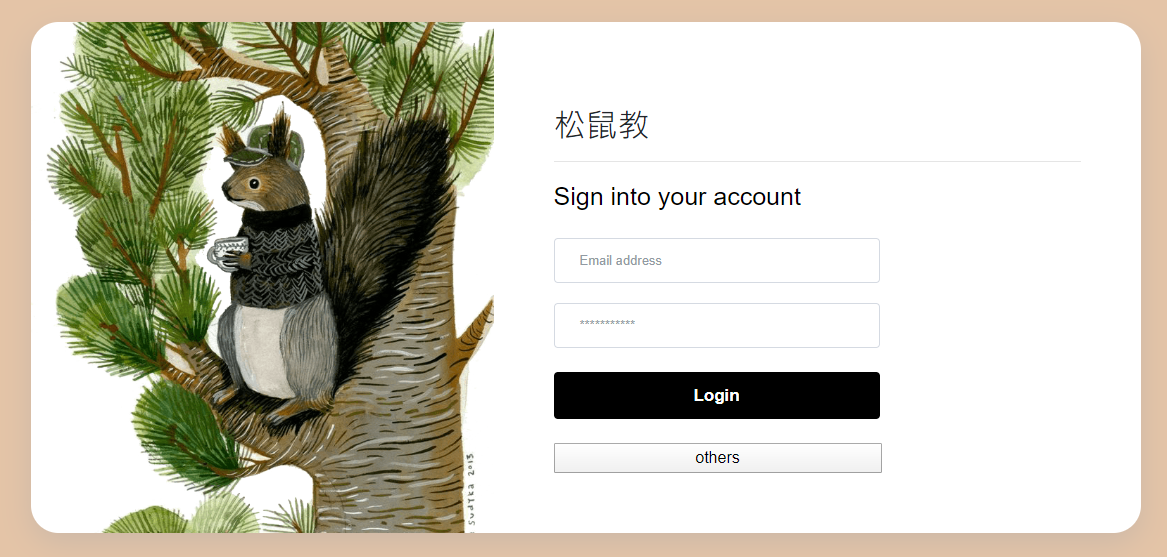
8.Relation EER diagram



10.系統功能：

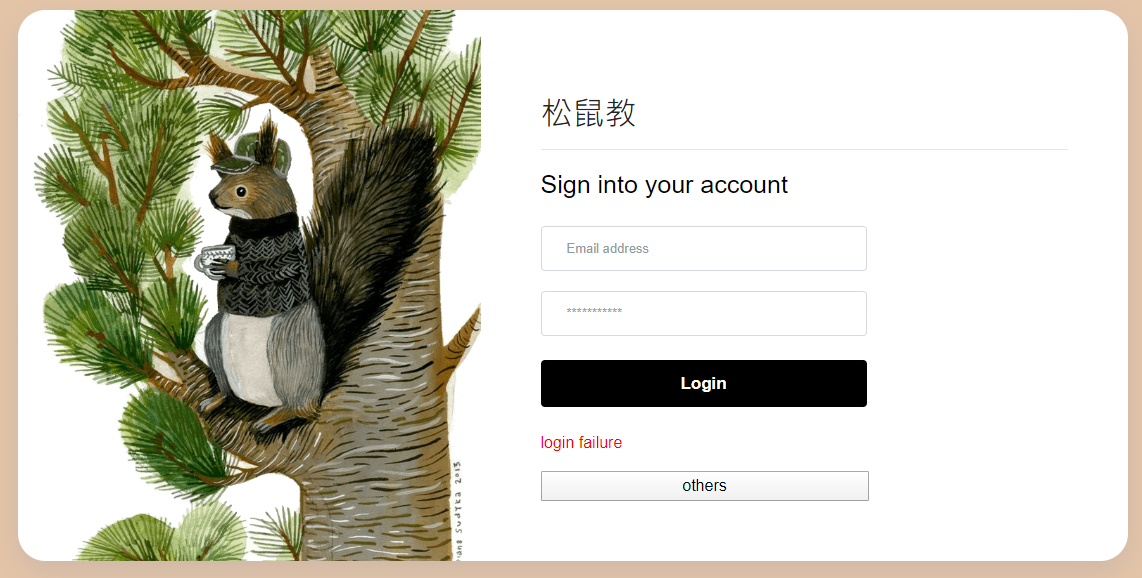
※輸入以紅色表示，輸出以藍色表示

1. 共同介面



1.當登入成功時，依照使用者權限，導向底下相對應的介面

2.當登入失敗時，顯示如下圖



登入失敗訊息

對應程式碼：



無符合的資料  
表示登入失敗

拿取資料庫  
符合的資料

依照使用者權限  
決定重新導向的網址

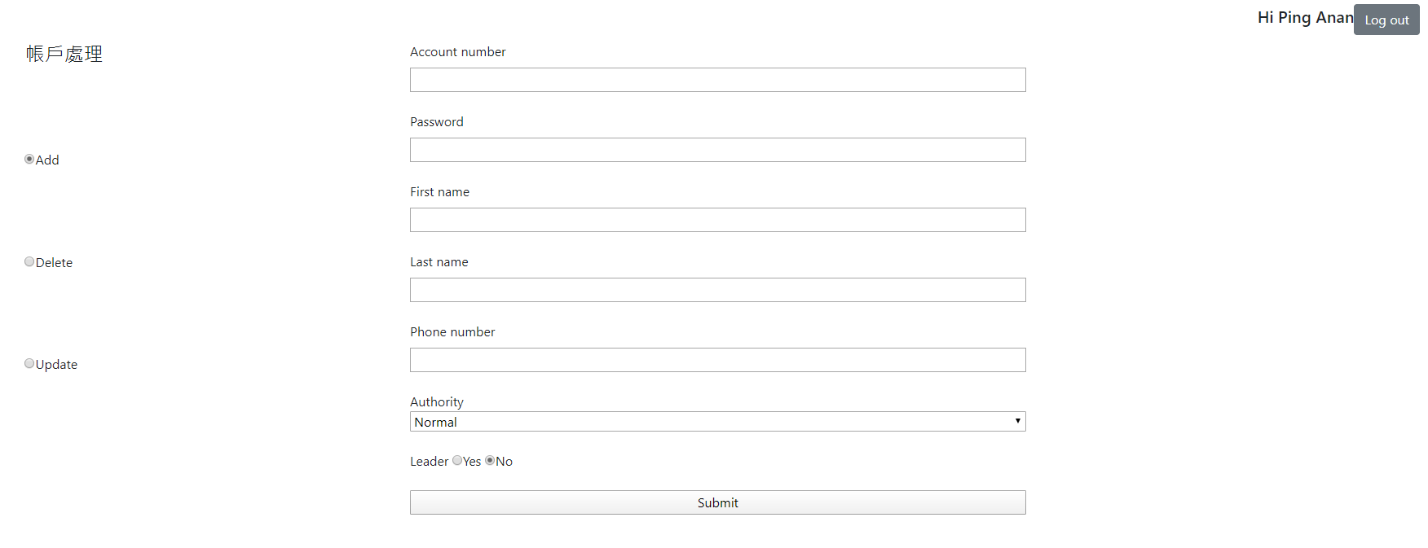
R代表Roor，管理者

B代表Boss，老闆

M代表Manager，經理

N代表Normal，一般員工

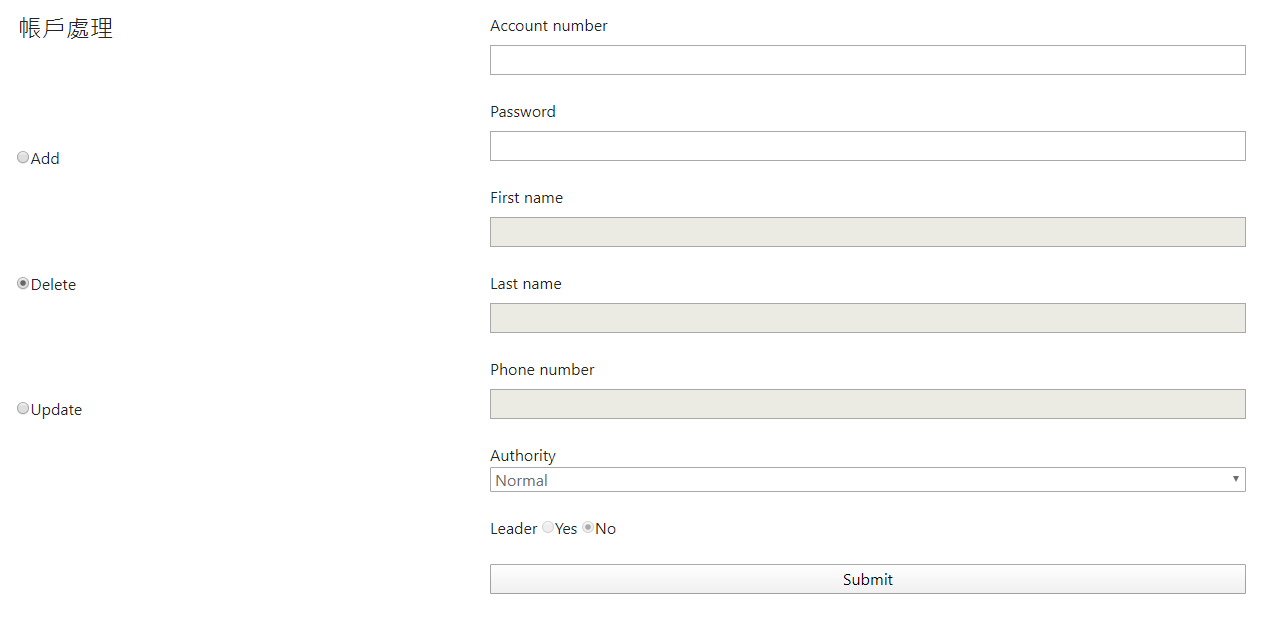
1. 管理員介面(僅一人使用，故陽春許多)



1.點擊右上log out按鈕，回到登入介面，所有使用者都有此功能，後續不在重複敘述

2.選擇帳戶處理(Add, Delete, Update)

Add & Update如上圖，Delete如下

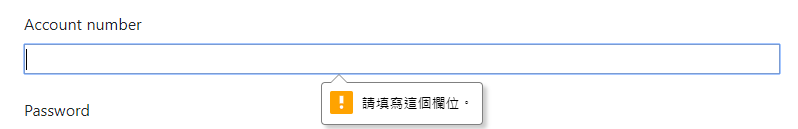


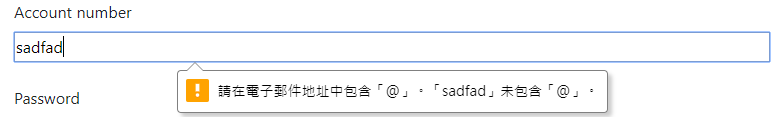
若選擇Delete，只需輸入  
Account和 password，  
其他欄位會封鎖

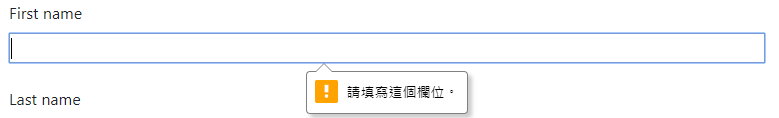
此處可選擇要  
add、delete或是update

3.輸入資料，格式錯誤(前端)，顯示如下圖

帳戶需email格式，密碼，號碼前端無限制，姓名不可空，其餘為選擇，不會錯誤





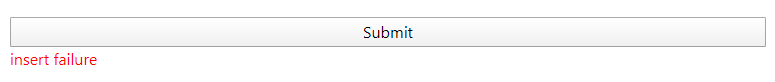




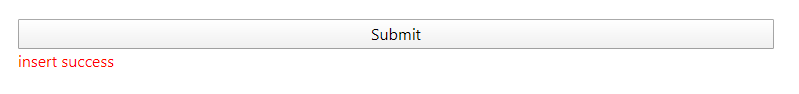
4.輸入資料，格式正確，顯示如下圖



5.輸入資料，前端成功送出，後端判定錯誤(違反限制，如key重複或資料定義錯誤)，如下圖

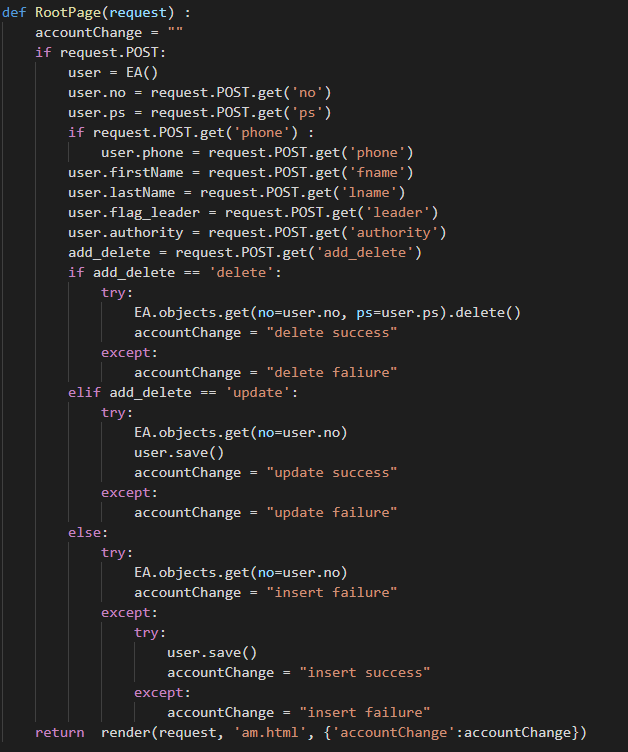


6.輸入資料，前後端皆成功，顯示如下圖



※刪除與更新顯示相似，不在贅述

對應程式碼：



Delete:

刪除對應的資料

可能成功或失敗

Update:

拿取資料庫對應的資料，並更新

可能成功或失敗

Add:

若拿到對應的資料代表此帳戶已存在

->失敗

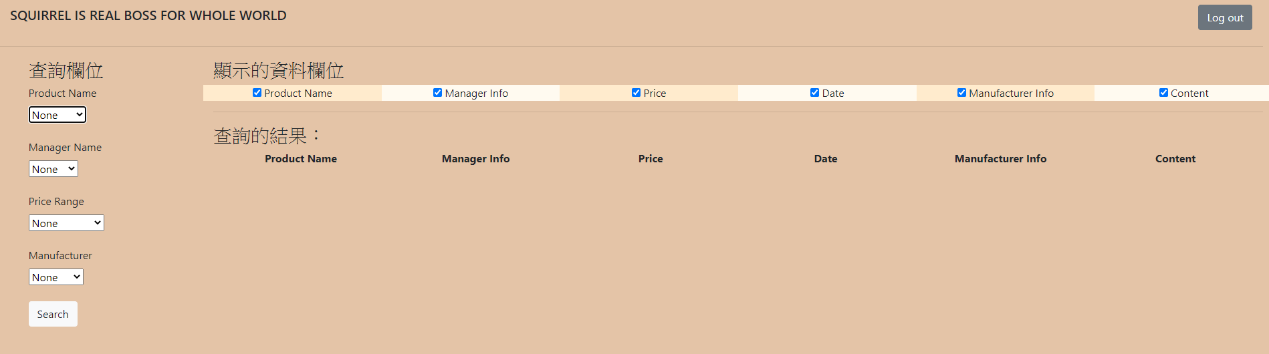
未拿到代表可新增

->成功

判斷使用者選擇

add、delete  
還是update

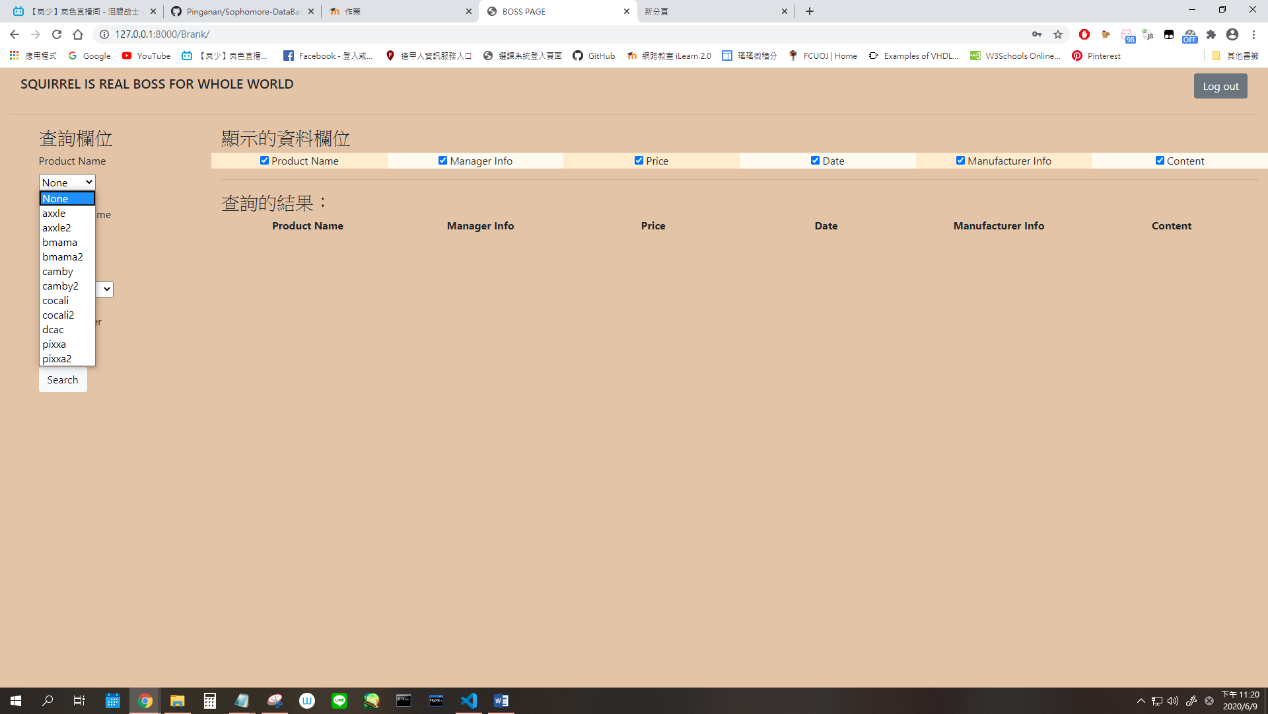
1. 老闆介面



選擇顯示條件

依照使用者的勾選

決定查詢結果的欄位

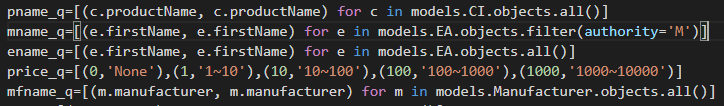


選擇查詢條件

依照資料庫現有的資料

顯示下拉式選單

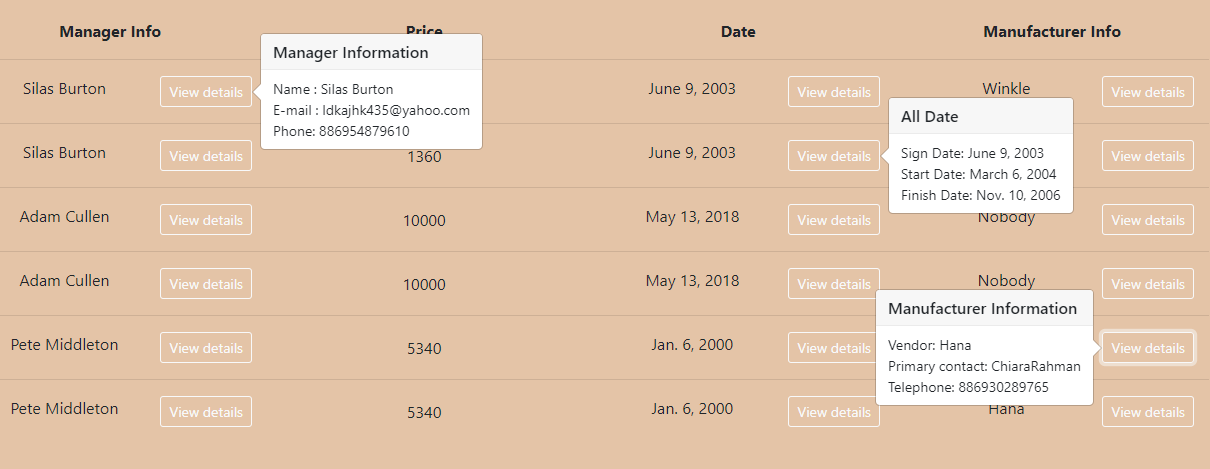
下面是對應程式碼



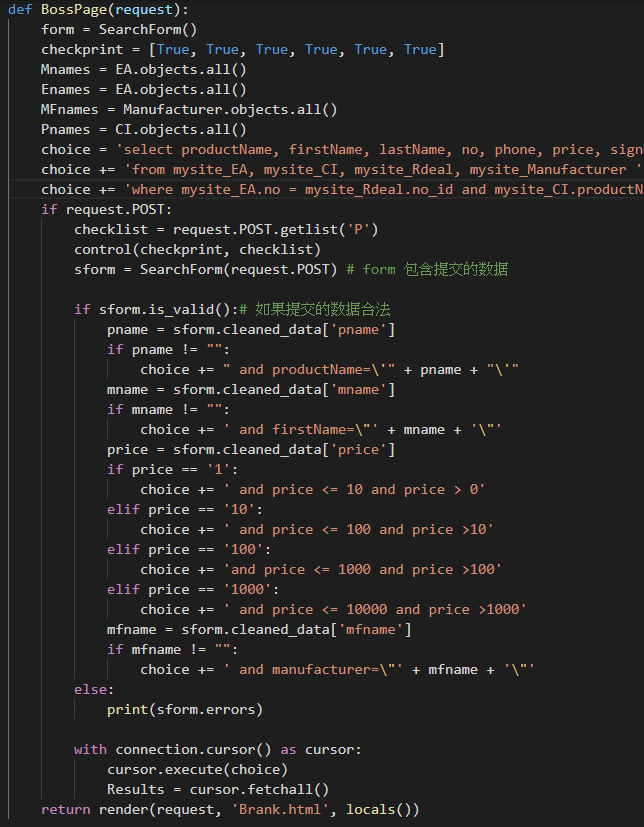
1. 由左方查詢資料欄，篩選資料內容(沒有選表示不規定)，及上方顯示資料欄，選擇要看到的資料，search按鈕按下，結果如下圖



1. 點view details，獲得更多資訊，如下圖



對應程式碼：



choice為用來存SQL指令的字串

此處擷取不完整

完整字串在功能最後

根據前端拿到的查詢資料，把條件加到choice字串的後面

利用cursor使用SQL指令，

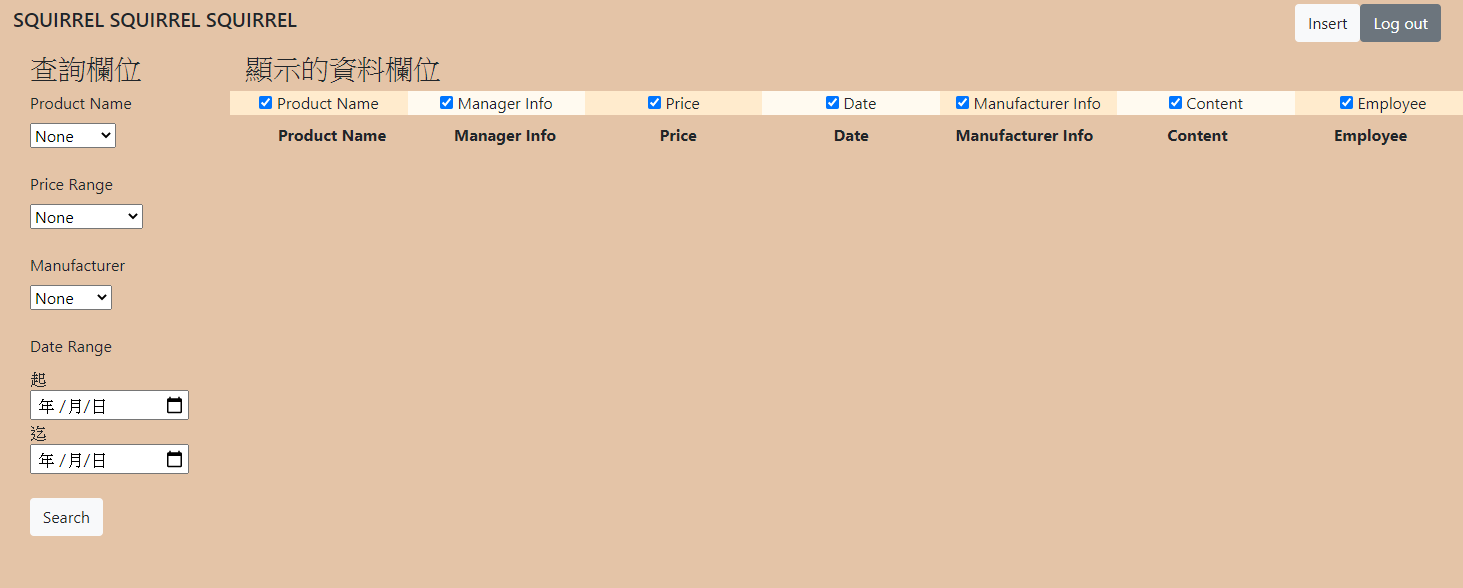
將剛剛的指令字串輸入進去

Results接結果的list

1. 經理介面

經理有兩個頁面，按下Insert按鈕

會跳至新增與更新專案的頁面



與老闆頁面類似，

但經理多了兩個欄位

Employee和Detect Date

稍後解釋

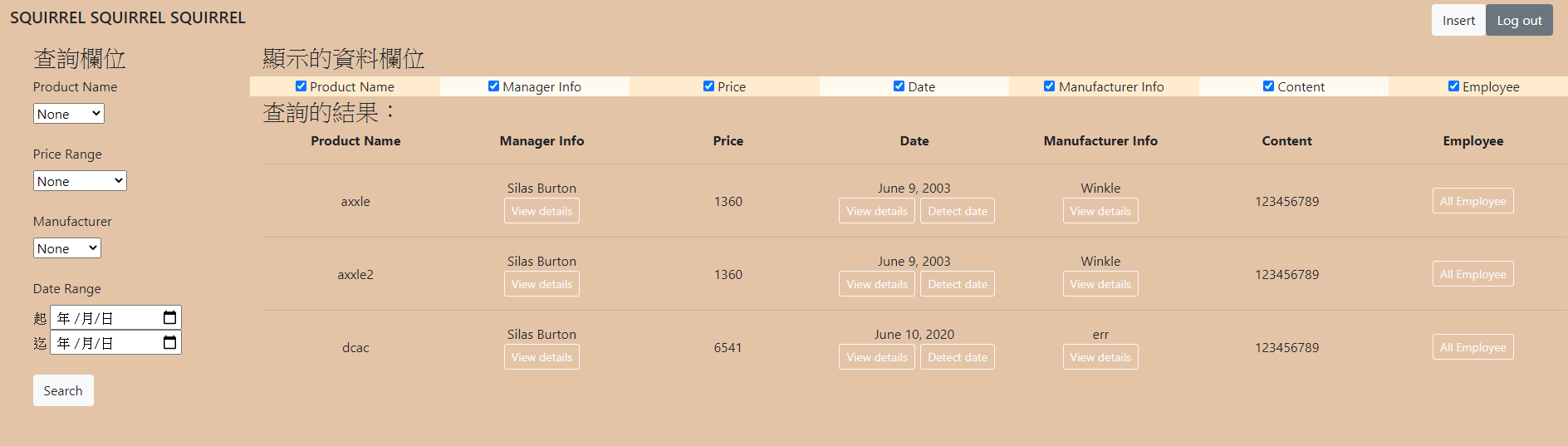
與老闆的頁面類似，

但多了日期條件可以篩選

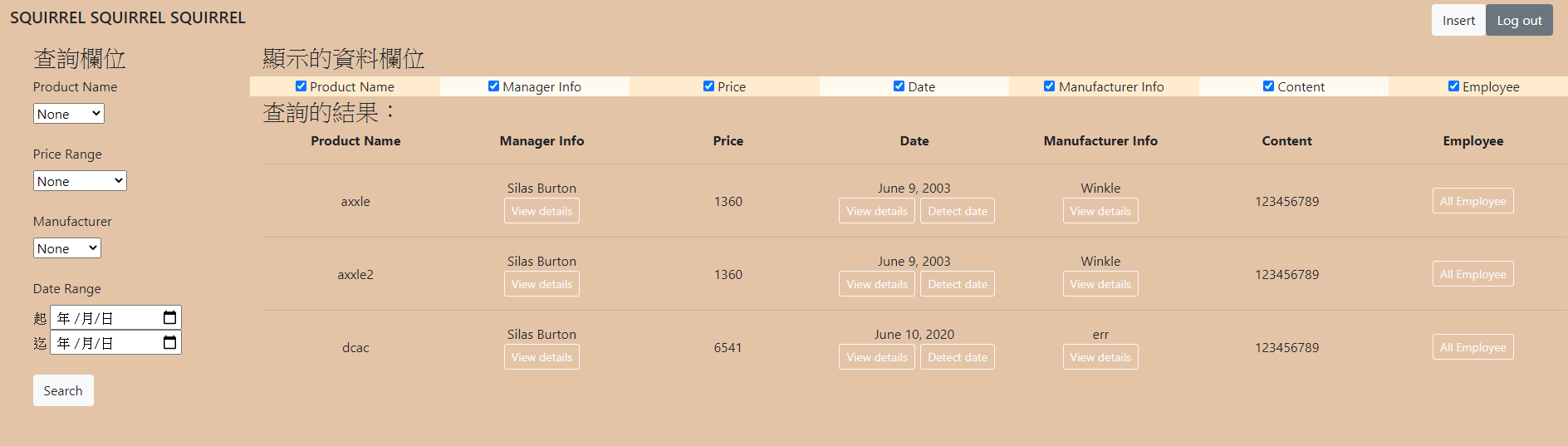
並且有隱藏的篩選條件：

只會顯示此帳戶負責的專案

1. 由左方查詢資料欄，篩選資料內容(沒有選表示不規定)，及上方顯示資料欄，選擇要看到的資料，search按鈕按下，結果如下圖

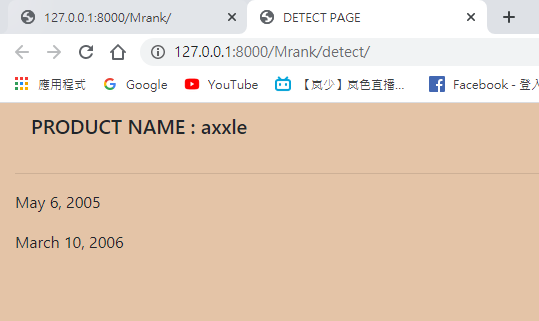


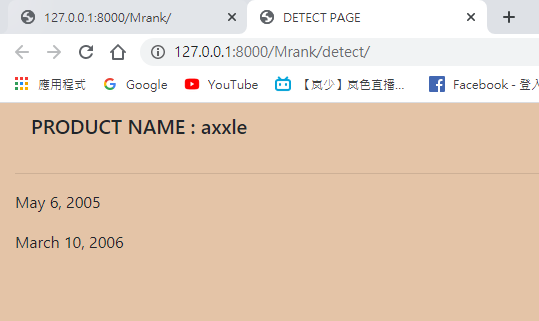
1. 點view details，獲得更多資訊，這部分和老闆介面一樣，這裡示範DetectDate和Employee按鈕



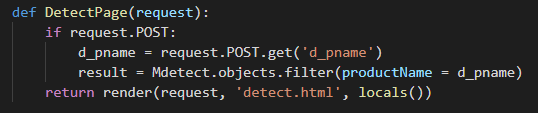
1. 按下Detect Date按鈕會打開新視窗，顯示此專

案的所有檢測日期



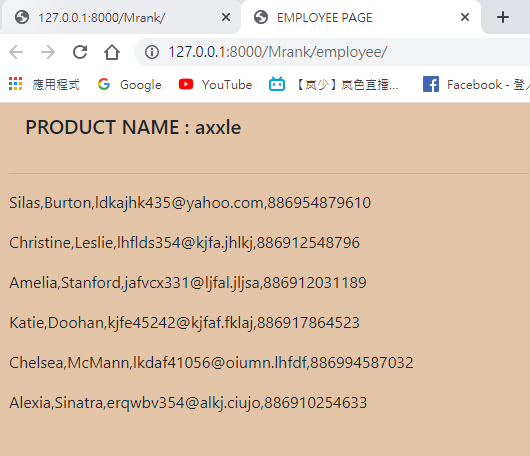


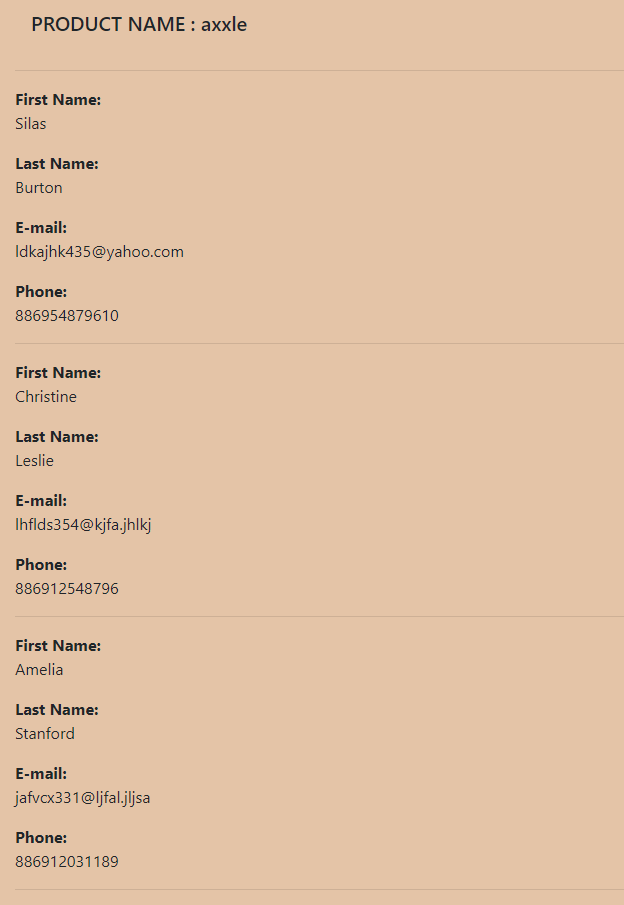
對應程式碼：



1. 按下Employee按鈕會打開新視窗，顯示參與此

專案所有員工的名字，E-mail，電話

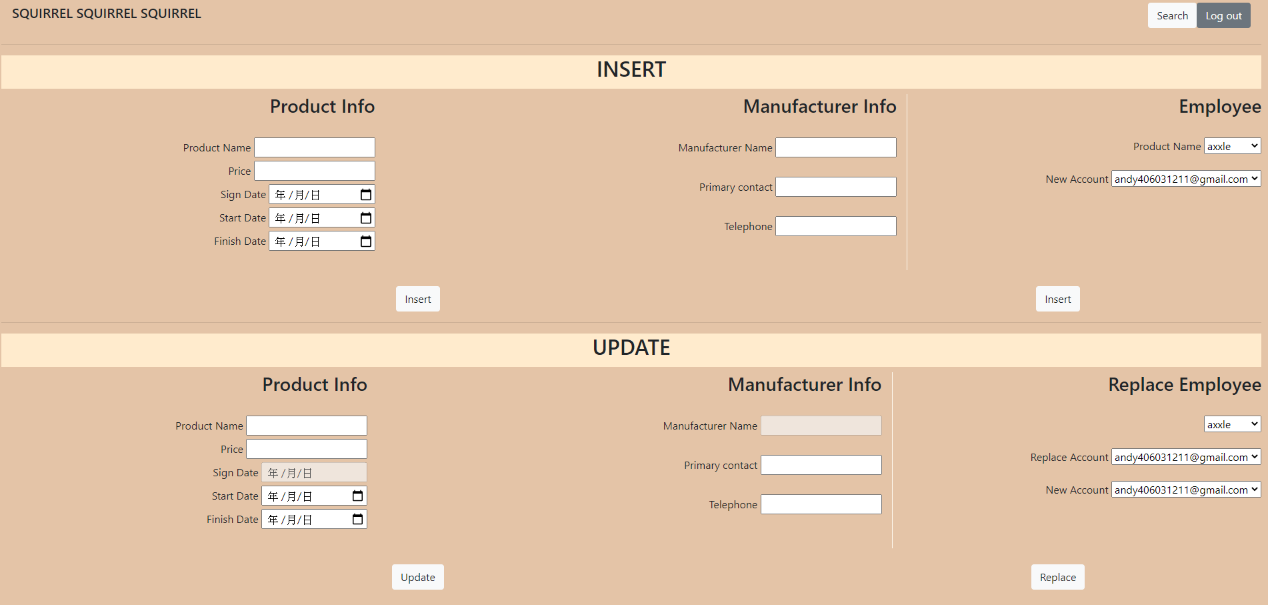




1. 按下Insert按鈕，跳至另一個頁面

按下Search按鈕

會跳至上面的查詢頁面



紅色框分別代表四個表單

左上為新增一筆新的專案

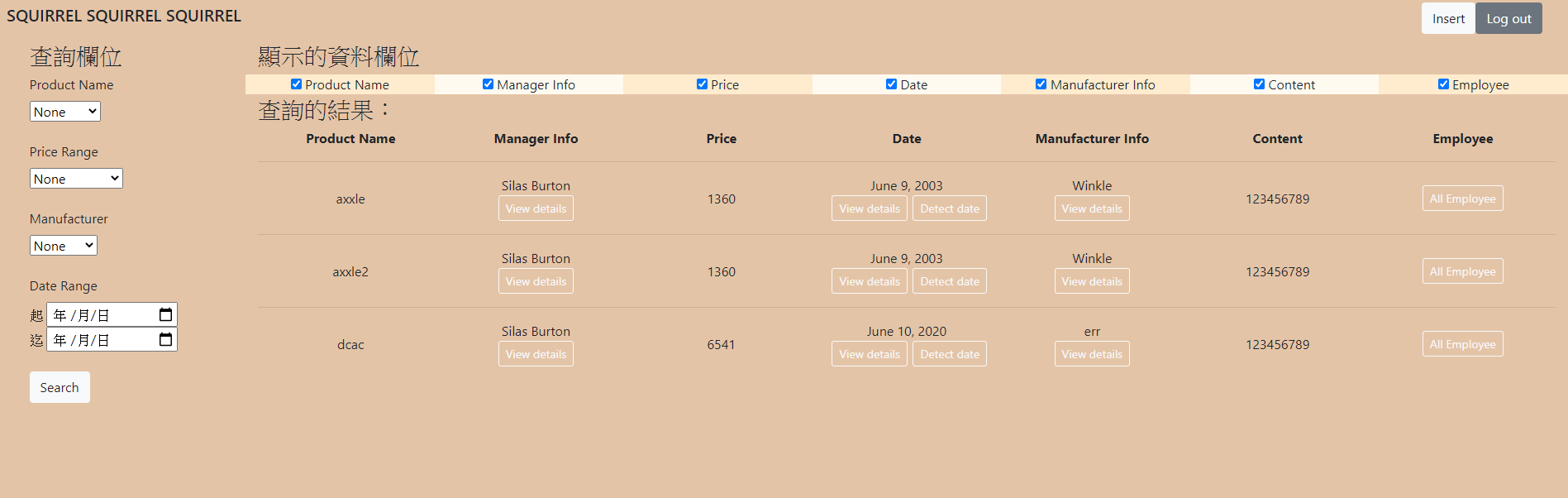
右上為選取專案並新增參與的員工

左下為更新某個專案的資訊

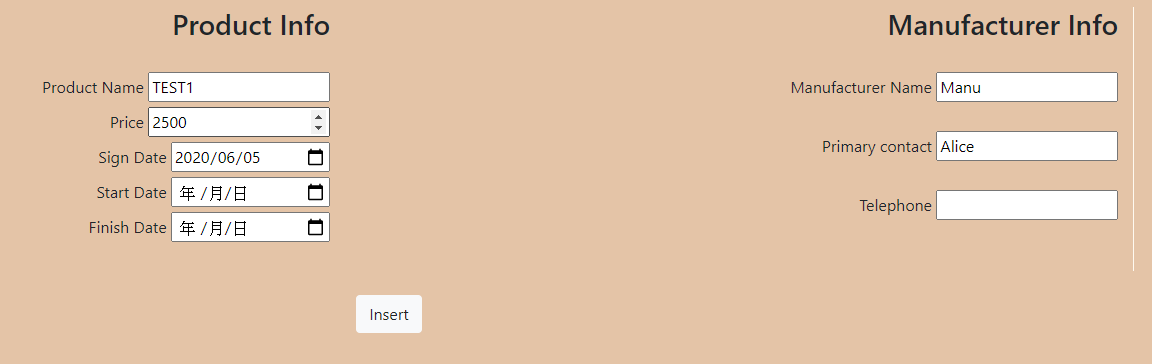
右下為選取一個專案，替換參與的員工

與查詢頁面類似的下拉式選單

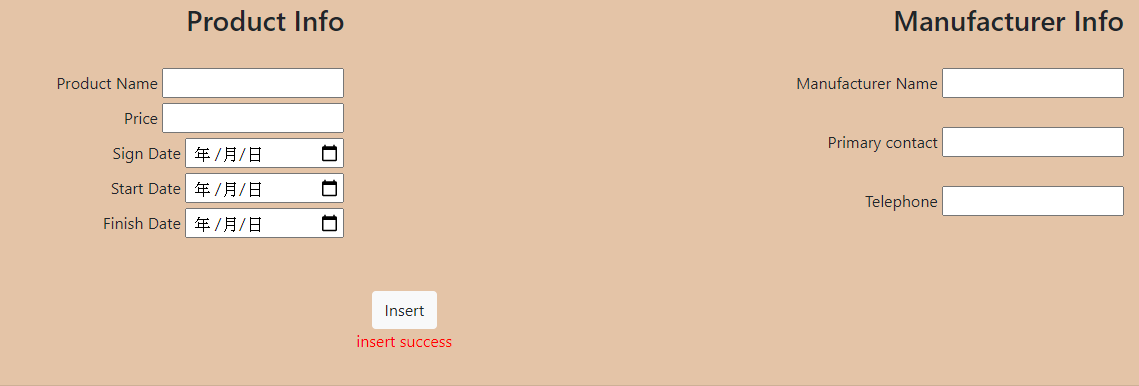
1. 使用者可在左上的表單填入資料，並按下Insert按鈕，按鈕下方會提示成功或失敗。



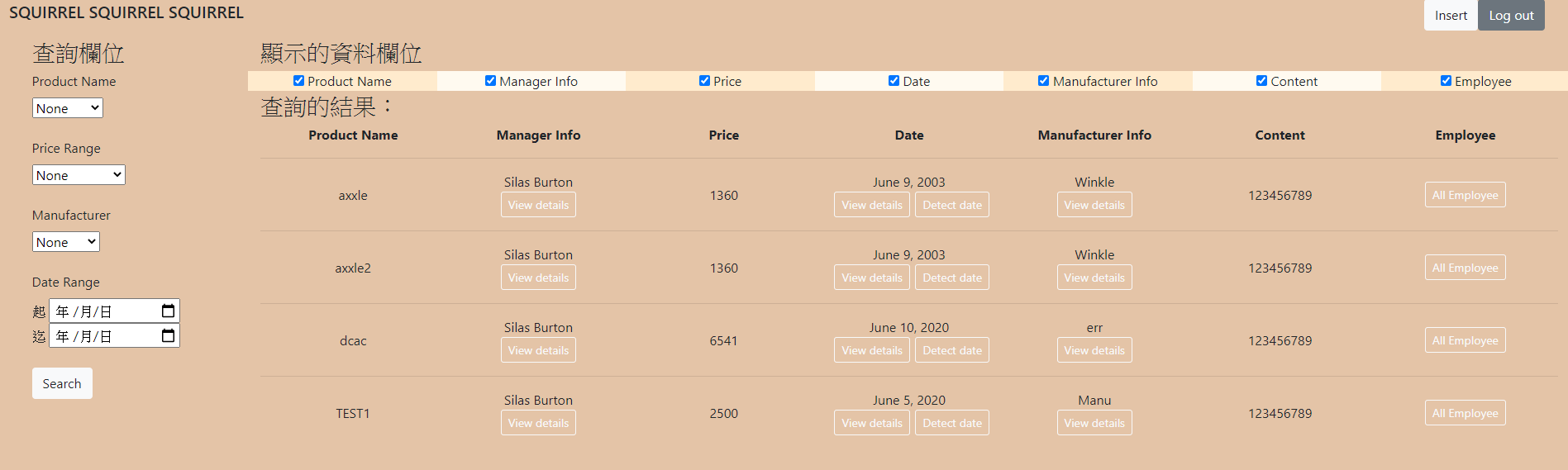
這個帳戶原本共負責三個專案



新增一筆專案，此圖空著的欄位為可填可不填，點擊按鈕

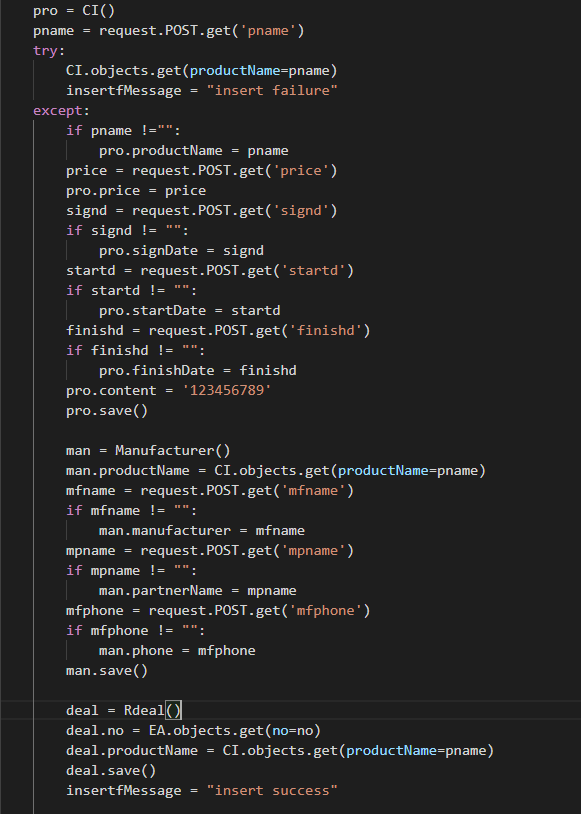


提示 insert success



回到search檢查，新增成功

對應程式碼：



取使用者輸入的值，

並存進CI table

取新增至CI的資料

把此帳戶加入至Rdeal table

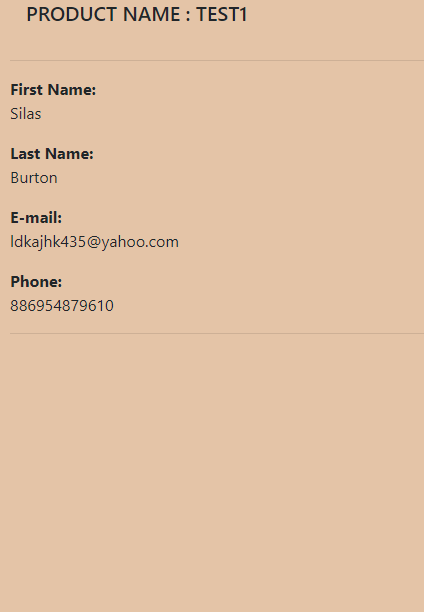
代表他參與此專案

取剛剛新增至CI的資料

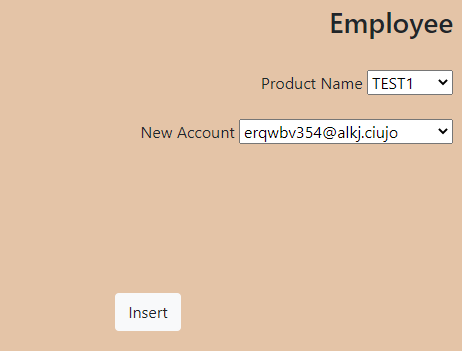
將使用者輸入的值存至Manufacturer table

確認資料庫是否已有這筆專案

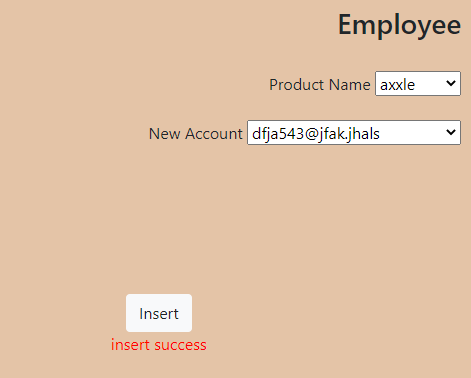
1. 使用者可在右上的表單選擇資料輸入，並按下Insert按鈕，按鈕下方會提示成功或失敗。



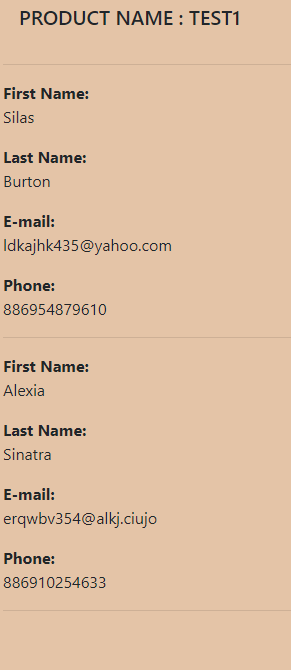
提示 insert success



選取好資料，點擊按鈕

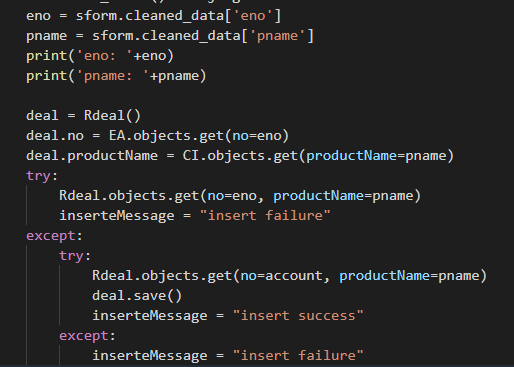


提示 insert success



多了一個參與的員工

對應程式碼：



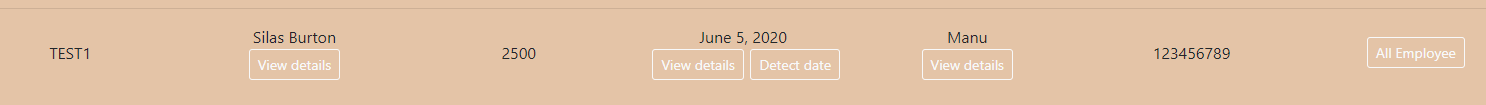
確認資料庫是否已有

使用者輸入的資料

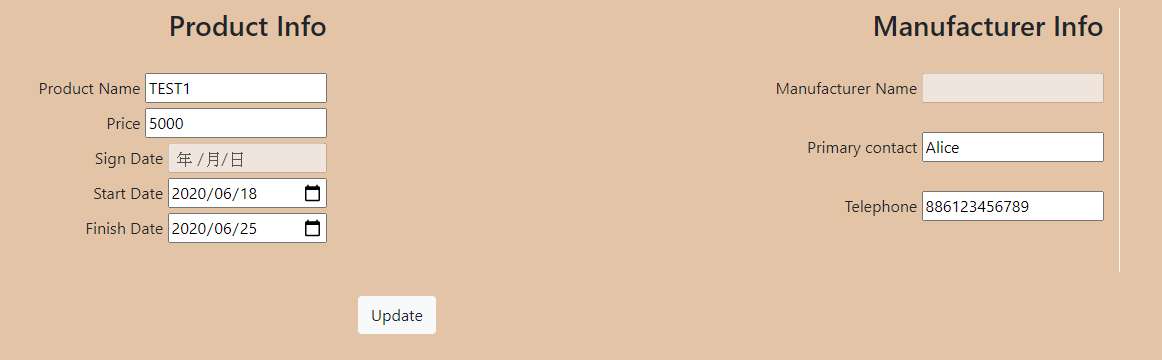
確認操作的專案

負責的經理為此帳戶

1. 使用者可在左下的表單輸入資料，並按下Update按鈕，按鈕下方會提示成功或失敗。



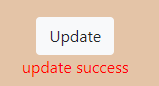
原本TEST1的price為2500



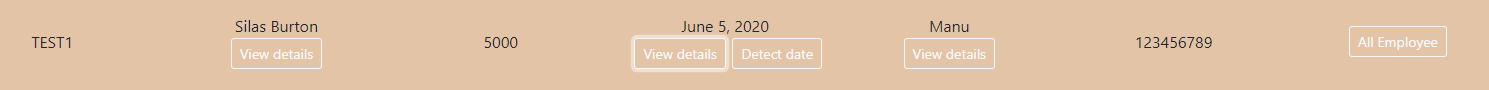
填入更新的資料

price為5000

點擊按鈕



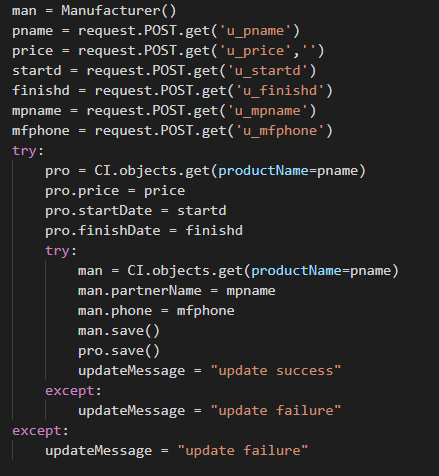
提示 update success



price為5000

更新成功

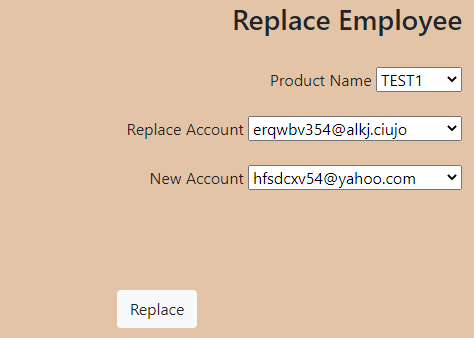
對應程式碼：



把更新的值存到Manufacturer table

確認這筆專案是否存在

1. 使用者可在右下的表單選擇資料輸入，並按下Replace按鈕，按鈕下方會提示成功或失敗。

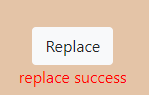


此圖的資料代表要把TEST1專案其中一個參與的員工換掉，被替換的帳號為

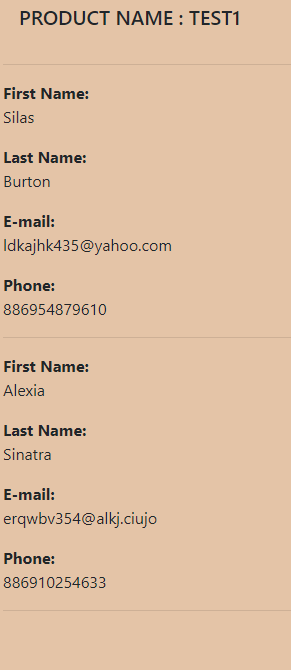
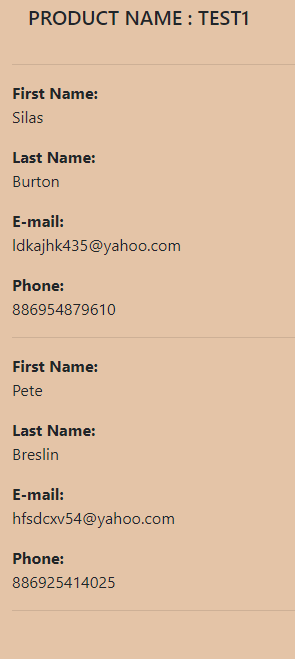
Replace Account

新的員工帳號為New Account

點擊按鈕



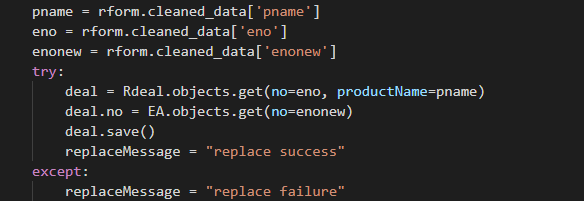
提示 replace success

原本的TEST1

查看新的TEST1，替換成功

對應程式碼：

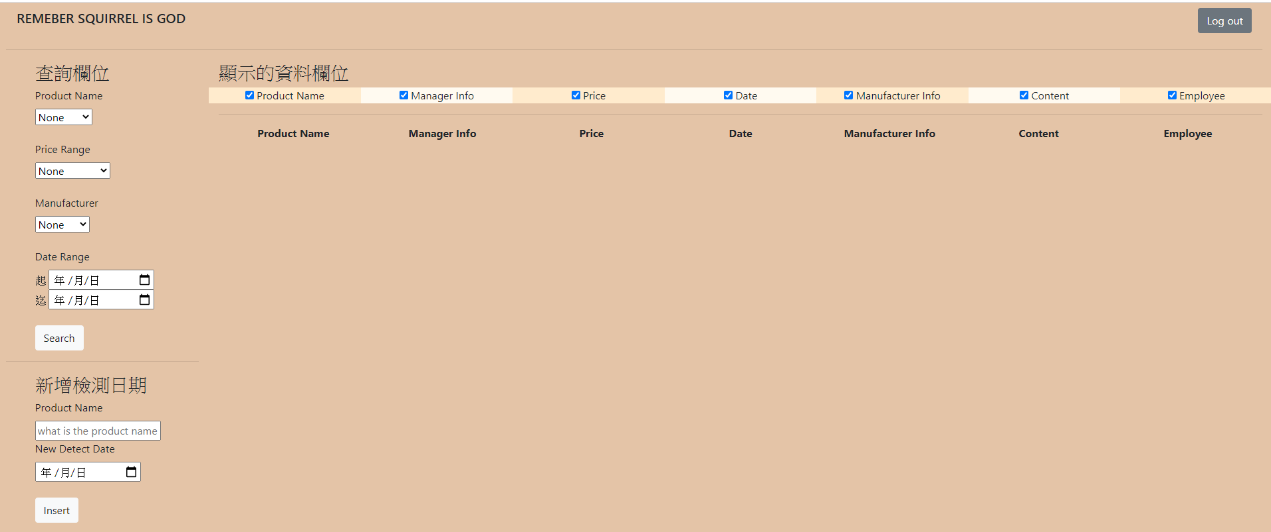


Rdeal描述員工和專案的關係

先取得使用者想替換的資料

再將no替換成新帳號並儲存

4. 員工介面



與經理頁面相同，

擁有Detect Date和  
Employee欄位

與老闆介面的查詢條件相同，

但有隱藏的篩選條件：

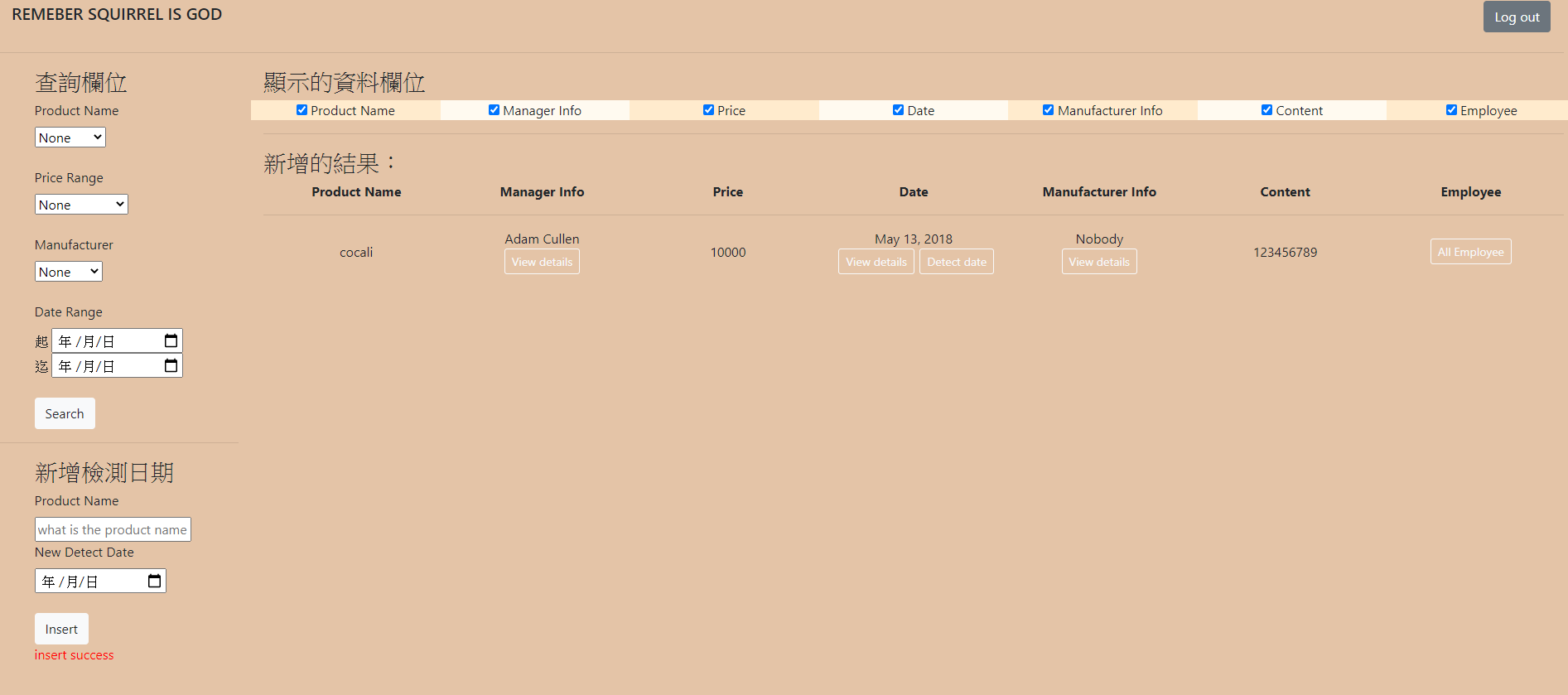
只會出現此帳戶負責的專案

員工可以新增專案的檢測日期(當他有參與此專案)

1. 填入Product Name以及新增的日期，並按下Insert按鈕，網頁會提示成功或失敗，若是成功，新增結果處會出現剛剛更動的專案資訊

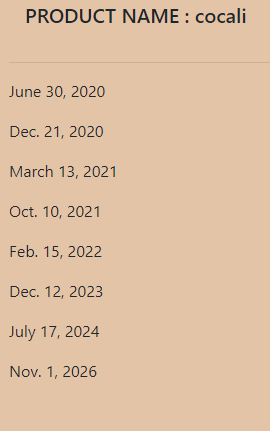
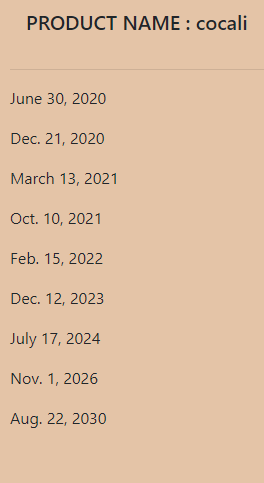


填入資料，點擊按鈕



出現剛剛新增日期的專案

提示insert success

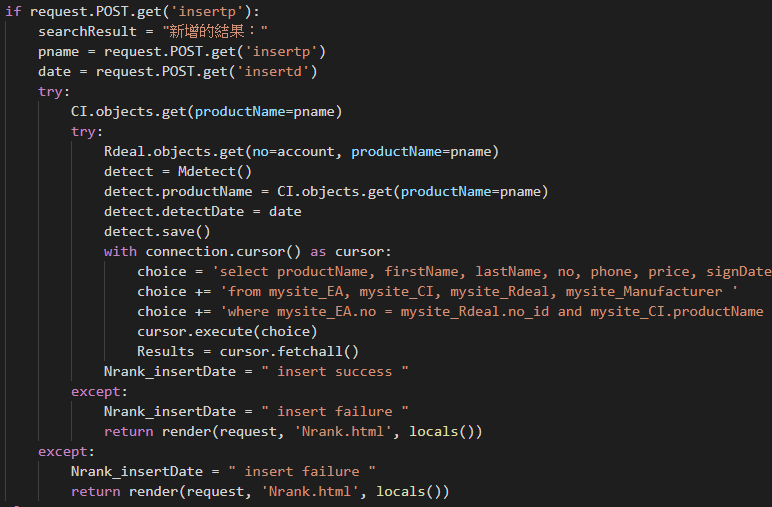


原本的檢測日期

新的日期新增成功

對應程式碼：

確認這筆專案是否存在



用SQL指令把剛剛操作的專案資料查出來

用Result傳給前端

把新增的日期存至Mdetect table

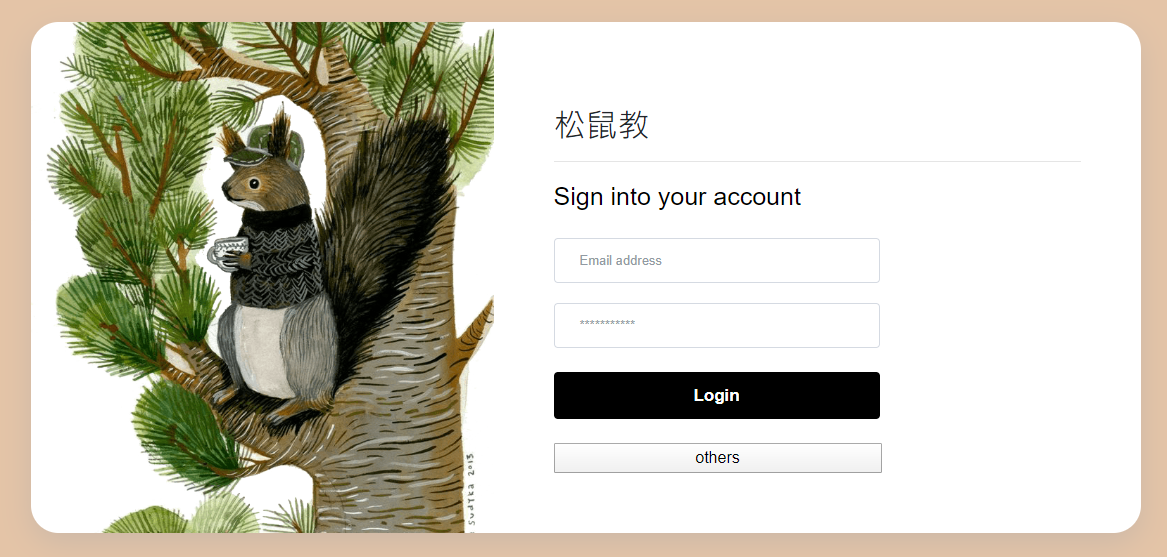
補充： choice字串全文

choice = 'select productName, firstName, lastName, no, phone, price, signDate, startDate, finishDate, manufacturer, partnerName, mphone, content '

choice += 'from mysite\_EA, mysite\_CI, mysite\_Rdeal, mysite\_Manufacturer '

choice += 'where mysite\_EA.no = mysite\_Rdeal.no\_id and mysite\_CI.productName = mysite\_Rdeal.productName\_id and mysite\_Manufacturer.productName\_id = mysite\_CI.productName and mysite\_EA.authority = "M" '

補充介面



點擊進入

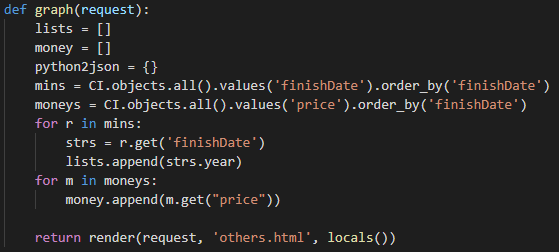


畫經濟曲線圖(程式解釋這塊)

重要成員(可以寫個董事之類的)

公司宗旨(中和英) ，純文字不多說明

對應程式碼：



抓finishDate(x軸)及price(y軸)，並以時間軸排列

11.心得

D0782956 平祖安：

本次專題，問題重重，一開始要做時，過沒有兩三天，我的電腦螢幕就忽然壞掉，於是只好學校申請進出資電館的權限，以方便我使用資電235不然我能用報告的時間就只能到晚上九點，然，資電的電腦沒有那麼的配合，當我嘗試使用課堂提供的資源時，偏偏只有教師機可以成功開啟，使我處理時一直處於無解的狀況，後來才知道是權限問題，然，當你跨過問題時，只會有更多問題，後面我們嘗試使用django做ROM架構，但遇到了資料庫成功建立，卻連接不了的窘境，設定檔跟網路上比對多次，依然無法解決，最後發現，是版本相容性問題，過此，終於成功將資料庫和網頁結合，但，問題接踵而至，建立table時，django的模型，不允許的限制許多，而且預設參數也都要一個個確認，路徑方面，考慮到一些資安的問題，所有也費了不少苦心，例如回傳路徑方式跟回傳網頁方式的差異性，且html的參數如何跟django的函式共同使用方式，這點由隊友協助解決，後來，因為參數的名稱必須相同，為此還小有爭執，還有因為我後來嫌棄前端介面醜陋，導致功能的程式被放到美觀之後，這點是我這次感覺最不可取之處，資料的增刪查改，因為是關聯性資料，所以增刪改時，必須考慮值的限制，和參考問題，

查的時候，因為django的joint table能力有點低落，於是使用了cursor，以讓我們使用SQL語法，但是，html跟django對於值的放入截然不同，一個是list一個是array，於是產生許多傳遞問題，處理方式是前後端都對值做一次篩選，以便拿到所要的內容，到此技術上的問題大多解決。

心境上的問題，我稍微計算一下，做三層式架構比SQL指令的最多多20分，但是這20分也不過佔總成績5(20\*0.25)分，如果報告是不考慮的話，就僅僅3分的價值，個人對於此配分方式，備感失落，因為當我得知此時，已經是快完成的階段，進也不是退也不是，希望看完這裏可以有些許的改變，或未來做更動。

失去的部分是不可勝數的時間，和腦內的糖分，雖然成績上可能不會有比其他人有更多的亮眼成績，但我個人也是透過此次機會，專心的研究了網站建立和資料庫操作和連結，有些老師也會提供一些網路管理的產學合作，根據此次的經驗，雖仍不足，但我們應該有一定的門檻，接觸一些外界的運作模式，所以這堂課最好的發揮價值，我個人覺得未來會慢慢體現，相比於此，或許成績的兩三分也不是如此重要。

D0746353 岳 靈：

我覺得這次的專題讓我體會到什麼是分工和自學的重要，因為需要在有一點概念的情形下去做出一個龐大的東西，並且需要跟夥伴們合作分工實踐出共同的想法。先不論分工光是要理解、釐清應該先做什麼就是個不小的問題了，因為如果不知道要從哪開始也就不知道該怎麼分配，這時候就是要聚集並討論資料庫如何實做出東西，說真的雖然老師有給補充的東西但如果沒有解釋真的看不懂，不知道順序、流程或是應該載哪些東西來用，就是”摸索”，我不知道其他人，我們這組是去借圖書館的書來看，需要甚麼就翻一翻書，再不然就去詢問google。

還記得有天晚上在資電235待到11點才走，因為在忙著搞專題，幾乎天天都可以搞到那麼晚，甚至有天有人搞到1、2點，真的是專題專題再專題，唯一的想法是這些都結束後需要好好放個假。

D0746384 李筱文：

這次的專題我主要負責前端和後端的資料傳遞，發現Django真的提供了很多功能，只是要多看各種文件，例如表單驗證可以做到，幫資料庫篩掉不符格式的資料，很可惜這次沒有用到Django的登入登出，這樣會比現在的版本安全一些。完成專題的途中，最讓我挫折的部分，是發現部分html的表單格式無法在Django後台讀取，因為基本的text類型可以讀取，導致直到整個前端頁面差不多寫好了，才發現這嚴重的問題，還好使用Django提供的表單渲染就可以解決了。

這次的專題我們在設置環境遇到的困難比較多，畢竟第一次用Django，再加上要和MySQL連線，還記得其中一個error是因為python版本沒有使用64位元，從error資訊根本看不出來，經過好幾次的重建專案，現在已經能順利地架起環境了。