學習心得

當初報到時，對於未來要將要上的課程有許多都已經接觸過了，但還是將自己回歸到最初的學習心態，跟著課程一起重頭開始學習。期間發現這樣選擇是對的，因為當有設計過的經驗再次複習基礎時可以更清楚理解語法為何要這樣使用且還可以跟之前使用過的方式相互對照，加強自己的基礎觀念。在基礎的部分也有一些自己完全沒注意過的使用方法，例如字串使用雙引號時變數是可以直接放入使用的，陣列型的變數只需要用{}就可以使用，之前習慣都是用點的方式串接字串，因為用得很習慣，所以也沒發現還會有其他的使用方式。當字串非常的冗長時也可以使用三個<<<接自訂的名稱，結束時在接一次自訂的名稱，但必須在該行的一開頭。這樣重頭認真的再學習一次，可以學習到其他不同的使用方式，讓自己開發時視情況可以有更多不同的選擇。

再學習新事物上，未使用的Git前，開發前總是先備份要修改的檔案，再三確認後沒有遺漏任何檔案後，才開始開發，但在有Git後，可以另開一個分支使用，當沒問題時再合併回主軸，或是有問題時也可以返回開發前的狀態，並且在Github上也可以看見每次commit修改的差異，清楚看到開發的歷程。使用錢老師的MVC，讓我發現原來網站的架構設計也可以做得如此神奇，讓整個網站方便擴充新功能，而且在同一套規範制度下，每個人都知道如何新增一個功能，不用像之前修改網站時必須要先研究整個網站的架構，清楚完整流程後才能開始規劃並修改或增加功能。

在研發單位各處的指導下，每完成一個題目都學習到許多。在MVC修改上，讓我更清楚知道Views是顯示，所以在controllers不能有echo，需要將json的資料顯示在專門的Views頁，如此MVC才能分割的更清楚。尋找最大的區塊學會在如此多重複動作下可以使用遞迴的方式重複執行。在活動報名系統限制人數的狀況下要使用FOR UPDATE，避免超過預期人數。

在簡易銀行系統中，學習到寫作方式統一，在團隊開發下，寫作方式統一，才不會相互看對方的排版不順眼，並學習到除了FOR UPDATE外的其他鎖表鎖行的方式。期間最大的收穫是想到一個不同的方式達成更好的效果，其方式與版本方式類似，用一個欄位儲存目前使用狀態(0與1)，表示是否使用。並利用壓力測試的方式同時同帳戶的存取款動作，驗證壓力測試後的明細資料與實際金額是相符的。在有想法後實踐，並通過驗證，證明可行，是滿有成就感的。

學習健壯性，瞭解一個重要的觀念是『永遠不要相信使用者』，且PHP是弱型別語言，所以更需要做好防範，且盡量將可能的發生結果考量進去，並且防範錯誤發生，必要時可以使用強制轉換型態，避免程式運行上的錯誤發生。學習到CURL可以存網站的cookie存成檔案方式，並且使用來通過檢查取得資料。Redis的set設定名稱與get指定設定的名稱來存取資料。研究RESTful，發現網站設計規範與MVC概念很像，用. htaccess檔設定網址的規範，並替換成Get的方式到Controller決定要new哪個物件執行結果，然後再判斷請求的是要html、XML或json的回傳格式，再回覆給使用者。

製作踩地雷遊戲時，讓我們充分玩遊戲體驗遊戲規則，在設計時就會很自然地把這些規則也一併的納入考量。在使用API前畫流程圖使我們建立觀念，請求過程中永遠只能少錢不能多錢。在實作API前，先玩別人的API，理解請求的規範，來操作系統達成使用的目的，如此設計時就能規範自己的使用要求，並撰寫API文件，讓使用者可以操作自己設計的API。在這個過程中，我們是先透使用觀察回饋的結果並嘗試理解後設計出類似的程式，所以在設計開發前，是需要透過不斷的嘗試，蒐集回饋資訊。