亞大生活圈

指導老師: **陳永欽** 教授

組員: 謝孟和 (106021014)、陳宣佑 (106021102)、林永隆 (106021384)

摘要

亞大生活圈是社群網站與電子商務網站結合的平台，用戶可以在這裡發文、購物。專案內容可以簡單分為前台與後台，前台是給使用者瀏覽、操作的介面，後台是給網站管理者使用的操作介面。

技術上，前台使用了 Vue.js 來打造單頁式網頁(Single Page Application; SPA)，讓用戶在頁面的轉換不會感受到重新載入網頁的時間；搭配 Vuetify 套件來製作具有 Material Design 風格的使用者介面；使用 PWA(Progressive Web Application) 讓網頁能像 APP 一樣能被安裝在手機上。線上交易的部分則是選擇了綠界金流來實現。後台則使用 PHP 簡單好上手的框架 CodeIgniter 3(以下簡稱 CI3)，來快速打造基本的管理頁面。

在開發過程上，為了能快速開發、簡化資料庫結構更改的處理流程，並建立大量亂數資料進資料庫，則使用了 PHP 的 Phinx 套件來完成。

網站目前架設在 Azure 的 CentOS 8.2 虛擬機上，網頁伺服器為 Apache，資料庫為 MariaDB。進入 https://byjt.servehttp.com 能瀏覽專題成果。

1. 前言

在新生季，最常出現在社群平台上的文章可粗略分為以下兩類: 大學資訊詢問以及二手書買賣。據我們觀察，通常這些資訊會發布在臉書社團或 Dcard 亞洲大學校版。Dcard 有規定禁止交換個資的規則，因此在 Dcard 發二手買賣約面交時間地點等留下個資的行為是不符規定的。臉書社團有人工審核社員的問題，不像 Dcard，註冊時是用學校信箱，能保證大多數成員身分是大學生，而臉書上的個人資料是可以偽造的，對於身分辨別正確度不如 Dcard 高。於是我們就想創造一個在二手買賣和發文不受太大限制的平台。

2. 專題內容

我們在 Azure 上開了一台 CentOS 8.2 的虛擬機，並把我們的專題架設在該虛擬機上。至於為何選用 CentOS 來當作伺服器，而不選用使用者社群龐大且學習資源相

對較多的 Ubuntu，是因為 CentOS 的更新頻率緩慢，且預設就有 SELinux 的保護，所以相較之下 CentOS 可能更適合拿來當作伺服器的作業系統。

資料庫則是採用 MariaDB。這是因為我們剛開始開發專案時，所選擇的 XAMPP 版本，預設就是搭配 MariaDB。雖然 MariaDB 會盡量與 MySQL 保持高度相容，但為了避免遇到未知的麻煩，所以最終還是選擇了 MariaDB。

後台是給網站管理者使用的操作介面。我們使用 PHP 過去曾風光一時的 CI3 框架進行開發，該框架的一大優點就是易於上手，且能透過簡單的步驟就設定好 CSRF、XSS 等攻擊的防護。

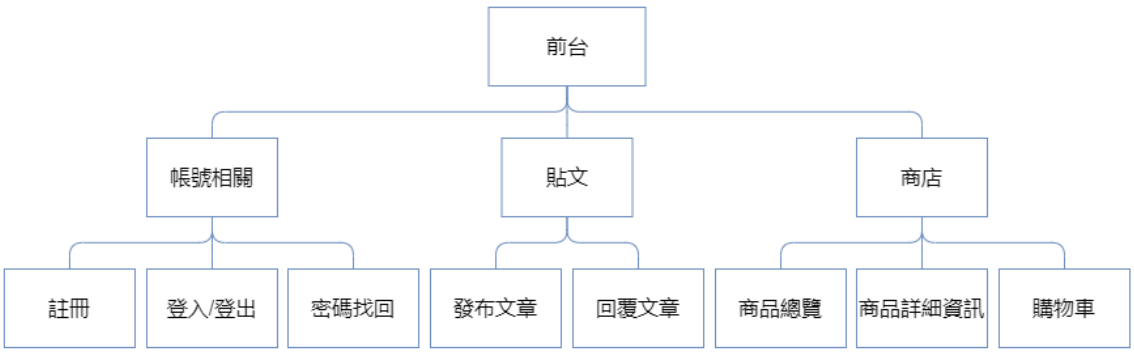
為了加速開發的速度，我們使用 Phinx 這個 PHP 套件，使資料庫的更動變得更簡單、快速，需要變更資料表結構、新增假資料時，只需要下簡短的一兩行指令，就能輕鬆的達到。

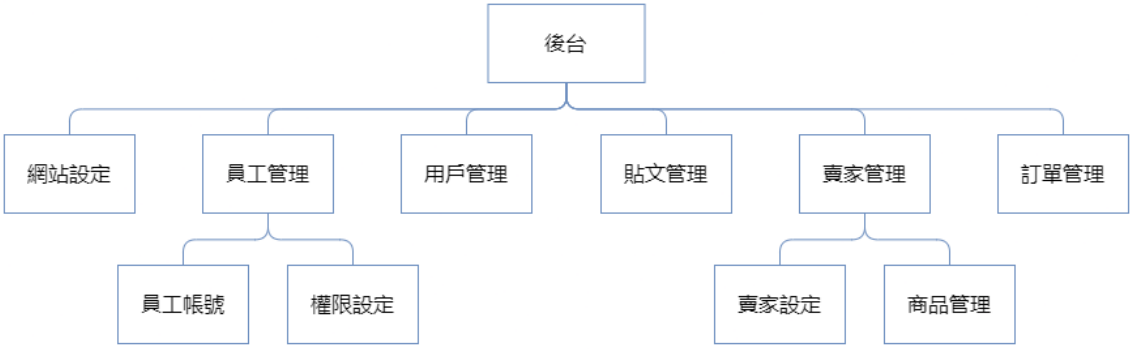
前台，給使用者瀏覽、操作的介面。使用了現在廣泛流行的技術 Vue.js，並搭配 Vuetify 套件，目的在於給使用者良好的操作體驗及介面。舉例來說，在不同頁面的跳轉，不會感受到整個頁面重新載入，而只有網頁的某些部分有重新被載入。網路上有人將這種行為稱作 SPA，因為伺服器從頭到尾只會給使用者一份 HTML 檔，接著 Vue.js 會根據網址的不同，向伺服器要求資料，得到伺服器回應的 JSON 資料後，Vue.js 再根據 JSON 資料呈現出不同的頁面。

第三方金流服務選擇了綠界科技 ECPay，因為我們認為它提供的串接文件是最容易瞭解的，而且官方也提供了許多的範例程式。

3. 成果

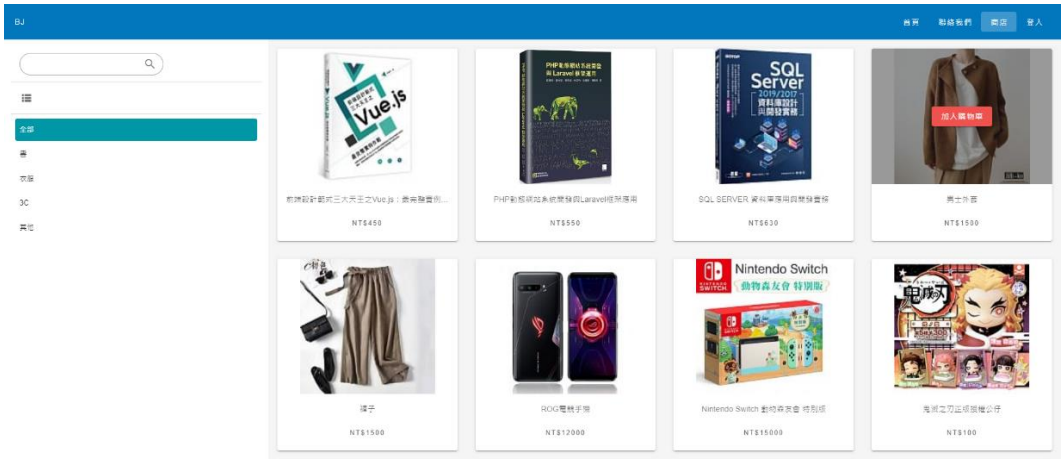
專案內容可以簡單分為前台與後台，前台是給使用者瀏覽、操作的介面，網址為 https://byjt.servehttp.com；後台是給網站管理者使用的操作介面，網址為 https://byjt.servehttp.com/BJ/2f09ce64。網站架構圖大致如下:



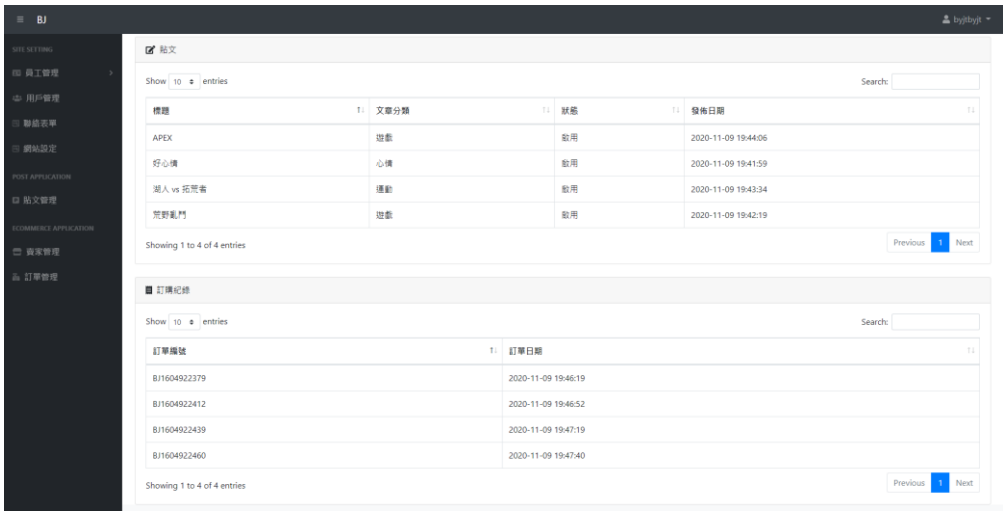


在前台的部分，貼文的功能需要登入才會顯示，商店的部分雖然可以瀏覽，但一樣需要登入才能加入購物車，然後做結帳的動作。貼文與商店的介面呈現方式也不太一樣，貼文頁面不斷往下滑，就會不斷地載入其他貼文；商店頁面則採用分頁的方式，一頁有 16 項商品。

發佈文章的部分，套用 CKEditor 線上編輯器，讓使用者不是只能夠打打文字而已，還能插入圖片並放在想要的位置。至於商店的部分，點選商品查看商品資訊與選擇購買數量。



後台的部分，在大量的頁面都提供了排序、搜尋、分頁等功能來過濾資料，以幫助網站管理人員更快速的找到資料。



4. 結論

各個成員對專題注重的面向不太一樣，有人注重使用者體驗，有人注重資料庫的設計，有人則注重專案的可維護性等等。再加上專題的內容有點太多太雜了，導致我們沒有辦法好好地去解決任何開發時遇到的問題。

先以 Chrome 近期對第三方 Cookie 制定的新規則來說，從綠界結帳頁面跳轉回到我們的網頁時，會發生 Cookie 不見而導致帳號被登出的問題，但因為發現問題的當下，不是所有成員都專注在串接金流 API 上，所以這問題並沒有被及時的解決，甚至一度還以為是框架的 BUG，但其實這問題只需要設定 Cookie 的 Samesite 屬性與 Secure 屬性就能解決。

還有因為我們對於 Vue.js 的不熟練與綠界金流的不熟悉，所以在綠界結帳與選擇超商的部分，其實並沒有做到 SPA，使用者在這部分還是會需要頁面的跳轉，在這部分其實有違當初想要提升使用者體驗的初衷。

對於專案進度的規劃、掌握，其實也不太理想。因為隨著專案的開發，更多的想法也隨之出現:

1. 根據使用者瀏覽、購買的商品，進行分析，然後顯示使用者可能會感興趣的商品。
2. 利用 websocket 技術，達到貼文、留言的即時顯示。
3. 使用 Redis 記憶體資料庫，當作資料庫的快取，進而減少 PHP 向 MariaDB 直接做存取的機會。

所以許多的細節就被忽略、淡忘了:

1. 用戶可以上傳自己的頭像，用於貼文發表文章與留言互動顯示頭像。
2. 對於訂單的狀態沒有良好的規劃，導致訂單整體狀態與訂單內的商品出貨狀態可能會不一致。
3. 部分頁面稍嫌陽春。
4. 新增網站推播通知功能，讓使用者得知貼文新的回應或新的優惠商品等第一手訊息。

造成專題進度控制的不好，很大的一部分也許是因為「練功」的心態，大過於「完成專題」的心態。

雖然專題沒有完全達到先前規劃的所有目標，但是我們在製作專題的過程中，彼此教學相長，分享彼此的開發技巧、觀念，像是如何有效率地對 PHP 進行 debug、Git 的一些小技巧等等。所以彼此在只做專題的過程中，得到的收穫其實比預期來得多許多。