

專題書面報告

點餐機網站

<https://menu-project-client.vercel.app/>

(成果如網址)

目錄

- 一、 摘要
- 二、 研究動機與目標
- 三、 背景知識介紹
- 四、 研究方法與步驟
- 五、 網站設計與流程
- 六、 結果與展示
- 七、 結論
- 八、 未來展望
- 九、 參考資料及網站
- 十、 Github URL

一、摘要

近年來由於地球環境影響，全世界不管政府或民間團體越來越重視綠能這方面的事，為了不要過度使用紙張而導致過度開發森林，許多團體開始推動無紙化行動，盡量都把內容資訊化，不要再出現紙本內容，一來可以節省經費(紙張費用以及影印費)，也能達成聯合國永續發展目標，而我們做點餐機，只要印出一張 QRcode，讓 user 透過掃描能夠使用自己手機的 browser 進行點餐，除了減少紙張使用，如果透過行動支付付費，在這疫情肆虐的時代，也能減少人與人之間的接觸。

二、研究動機與目標

(一) 為何要做此一專題與實驗

近年來比較流行熱門的程式語言主要有 2 種，一種是 python，另一種就是 javascript，雖然 2 種語言在我們大學課程都有，但我們對於 javascript 較能掌握，有鑑於此，所以我們決定透過 javascript 的 React 框架，製作一個有關點餐機的 web 頁面，運用我們在大學期間所學，來共同完成這個專題。

(二) 預計達到的效果

我們預計我們做出來的東西會有 3 個頁面，其一是菜單頁面，在這個頁面會詳細介紹餐點中英文名稱、價錢、餐點圖片以及查詢功能，查詢功能的部分總共有 2 個東西可供查詢，例如 user 今天單純想吃豬肉，就可以在搜尋欄打上豬肉，就會跳出所有有關豬肉的餐點，以及可以查詢肉是否為原肉或重組肉，在搜尋欄打上原肉，菜單就會自動排除是重組肉的餐點，顯示是原肉的餐點在上面，其二是點餐頁面，在這個頁面可以點選餐點數量(若 0 就不須點選)，或者是不想看到其中一個餐點可以直接點選 remove 鍵，讓該餐點直接消失在購物車選單裡，若餐點點好之後滑至下方按下送出，此時可以選擇付現或使用悠遊付支付，其三是抽選頁面，當 user 不曉得要吃什麼可以按下抽獎鍵，透過抽選的方式會提供給 user 一個餐點品項建議，user 可以透過這個建議在返回點餐頁面去點餐，總體來

說，我們預計做出一個有菜單、有點餐頁面以及能夠提供行動支付的一個點餐機系統。

三、背景知識介紹

此節會有簡短的程式語言介紹以及後續提到的專有名詞

JavaScript

JavaScript（通常縮寫為 **JS**）是一門基於原型和頭等函數的多範式進階直譯程式語言，它支援物件導向程式設計、指令式編程和函數式程式設計。它提供方法來操控文字、陣列、日期以及正規表示式等。不支援 I/O，比如網路、儲存和圖形等，但這些都可以由它的宿主環境提供支援。它由 **ECMA**（歐洲電腦製造商協會）透過 **ECMAScript** 實作語言的標準化。目前，它被世界上的絕大多數網站所使用，也被世界主流瀏覽器（**Chrome**、**IE**、**Firefox**、**Safari** 和 **Opera**）所支援。

JavaScript 與 **Java** 在名字和語法上都很相似，但這兩門程式語言從設計之初就有很大不同。**JavaScript** 在語言設計上主要受到了 **Self**（一種原型程式設計語言）和 **Scheme**（一門函數式程式設計語言）的影響，在語法結構上它和 **C** 語言很相似（如 **if** 條件語句、**switch** 語句、**while** 迴圈和 **do-while** 迴圈等）。

對於客戶端來說，**JavaScript** 通常被實作為一門解釋語言，但如今它已經可以被即時編譯（**JIT**）。隨著 **HTML5** 和 **CSS3** 語言標準的推行，它還可以用於遊戲、桌面和行動應用程式的開發，以及在伺服器端網路環境執行（如 **Node.js**）。

React

React（也稱為 **React.js** 或 **ReactJS**）是一個自由及開放原始碼的前端 **JavaScript** 工具庫，用於基於 **UI** 組件構建使用者介面。

它由 **Meta**（前身為 **Facebook**）和由個人開發者和公司組成的社群維護。**React** 可用作開發具有 **Next.js** 等框架的單頁、手機或伺服器彩現應用程式的基礎。然而，**React** 只專注狀態管理和將狀態彩現到 **DOM**，因此建立 **React** 應用程式通常需要使用額外的工具庫來進行路由實作，以及某些客戶端功能。

Supabase

Supabase 是一個開源的後端服務平台，提供完全可擴展的 **PostgreSQL** 資料庫和 **RESTful API**。它結合了實時的 **WebSocket** 功能和角色認證，讓開發者能夠快速建立應用程式。**Supabase** 簡化了應用程式開發和數據管理，是現代 **Web** 應用開發的理想選擇。

簡單來說 **Supabase** 是一個整合後端程式、資料庫、用戶授權、訪問權限等後端即服務(**BaaS**)的平台，功能強大而且在有限的額度內免費使用，提供所有正在使用 **Firebase** 的人或是準備使用 **Firebase** 的人一個全新替代方案。

資料庫與 API 快速聯動

Supabase 是一個採用 **PostgreSQL** 作為資料庫的開發平台，先進入後台為你的專案設定好資料庫內的表格格式後，包含新增、刪除、修改等 **API** 即自動生成，而且可以無限制的請求。

PostgreSQL

PostgreSQL 是一個開源的物件-關聯式資料庫管理系統，在類似 **BSD** 授權與 **MIT** 授權的 **PostgreSQL** 授權下發行。

PostgreSQL 這個單字的拼法讓有些人覺得很難讀，特別是那些把 SQL 讀作 "sequel" 的人。PostgreSQL 開發者把它讀作 "post-gress-Q-L"。（Audio sample，5.6k MP3）。它也經常被簡稱為 "postgres"。

Git

git 是一個分散式版本控制軟體，最初由林納斯·托瓦茲創作，於 2005 年以 GPL 授權條款釋出。最初目的是為了更好地管理 Linux 核心開發而設計。應注意的是，這與 GNU Interactive Tools（一個類似 Norton Commander 介面的檔案管理器）不同。

git 最初的開發動力來自於 BitKeeper 和 Monotone。git 最初只是作為一個可以被其他前端（比如 Cogito 或 Stgit）包裝的後端而開發的，但後來 git 核心已經成熟到可以獨立地用作版本控制。很多被廣泛使用的軟體專案都使用 git 進行版本控制，其中包括 Linux 核心、X.Org 伺服器和 OLPC 核心等專案的開發流程。

Github

GitHub 是一個線上軟體原始碼代管服務平台，使用 **Git** 作為版本控制軟體，由開發者 Chris Wanstrath、P. J. Hyett 和湯姆·普雷斯頓·沃納使用 Ruby on Rails 編寫而成。在 2018 年，GitHub 被微軟公司收購。

GitHub 同時提供付費帳戶和免費帳戶。這兩種帳戶都可以建立公開或私有的代碼倉庫，但付費使用者擁有更多功能。根據在 2009 年的 Git 使用者調查，GitHub 是最流行的 Git 存取站點。除了允許個人和組織建立和存取保管中的代碼以外，它也提供了一些方便社會化共同軟體開發的功能，即一般人口中的社群功能，包括允許使用者追蹤其他使用者、組織、軟體庫的動態，對軟體代碼的改動和

bug 提出評論等。GitHub 也提供了圖表功能，用於概觀顯示開發者們怎樣在代碼庫上工作以及軟體的開發活躍程度。

截至 2022 年 6 月，GitHub 已經有超過 5700 萬註冊使用者和 1.9 億代碼庫（包括至少 2800 萬開原始碼庫），[\[7\]](#)事實上已經成為了世界上最大的代碼代管網站和開源社群。截至 2023 年 1 月 26 日，已經有超過 1 億開發人員使用 GitHub。

Vercel

Vercel Inc.（原名 **ZEIT**）是一家美國雲端平台即服務公司。該公司維護 Next.js Web 開發框架。

Vercel 的架構圍繞著 Jamstack 構建，並透過 Git 儲存庫處理部署。Vercel 是 MACH 聯盟的成員。

Vercel 的部署透過 Git 儲存庫進行處理，並支援 GitHub、GitLab 和 Bitbucket 儲存庫。儘管 Vercel 提供了對部署的自訂域的支持，但部署會自動在域下指定子網域

`vercel.app`，

Vercel 的基礎架構使用 Amazon Web Services 和 Cloudflare。

四、研究方法與步驟

（一）初步概念與規劃

我們一開始先把我們所想要呈現的功能想出來，在討論看看實作出來的可能性，經過討論我們決定先把菜單頁面做出來，並將後端菜單資訊透過 PostgreSQL 存放在 supabase 資料庫裡，再來實作點餐的頁面，以及進入點餐頁面前的 Homepage，然後在點餐頁面我們希望能夠至少加入一個行動支付的功能，但大部份的行動支付都需要商家驗證，執行上可能會遇上一些麻煩，但是到最後我們發現如果使用悠遊付好像可以解決這個問題，因為它不需要商家驗證，所以到最後

我們選了悠遊付來當成我們行動支付的服務。

(二) 網站設計

有了方向後，我們便開始著手處理，我們決定先把我們的 code 透過 git 版控上傳至 github，再透過 github 部署到 vercel，由於 vercel 支援 github 儲存庫，所以對我們來說就很方便，且 vercel 有提供免費的子網域，就不需要再花錢煩惱這些問題，對我們來說蠻理想的。

Vercel 有支援 javascript(包含 react)，於是我們的網頁使用 javascript 來開發，Vercel 是一家雲端服務提供商，專注於靜態網站部署和前端開發。它提供了一個托管靜態網站和應用的平台，開發者可以使用它來輕鬆部署他們的前端項目。Vercel 支援多種編程語言和框架，包括 React、Vue、Angular 等，同時也提供了自動部署、CDN 加速、SSL 支援等功能，使開發者能夠構建高性能的網站和應用。Vercel 也與 GitHub、GitLab、Bitbucket 等代碼托管平台集成，使開發者能夠通過版本控制工具輕鬆部署他們的項目。這使得開發團隊可以更快地交付產品，並實現持續集成和持續部署（CI/CD）流程。

總結來說，Vercel 是一個為前端開發者和團隊提供托管和部署服務的平台，旨在簡化開發工作流程，提供高性能和可靠性。

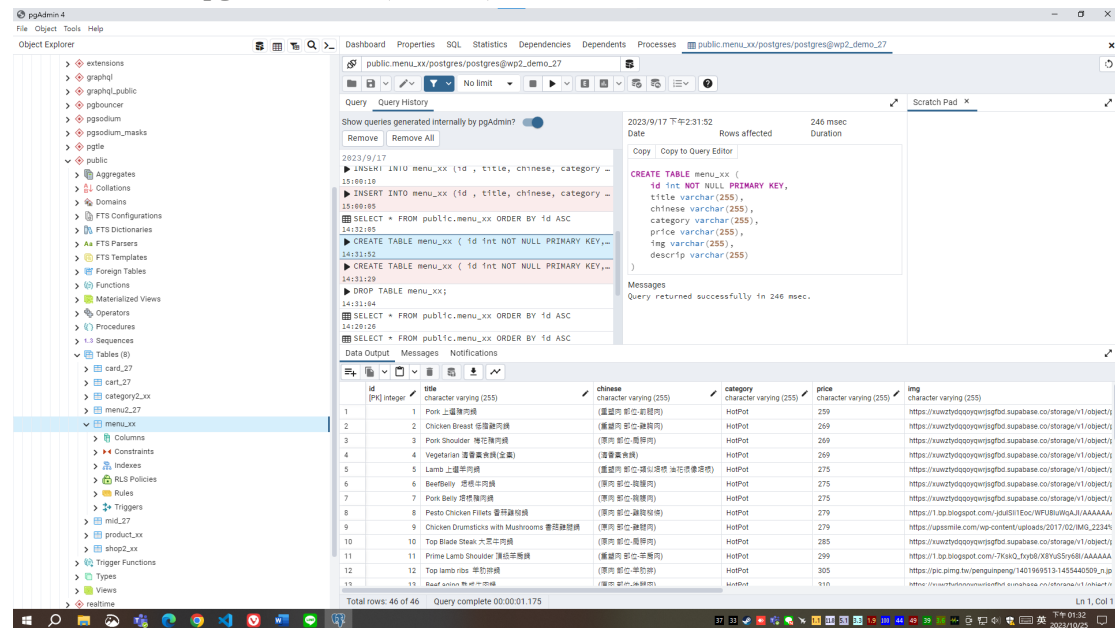
(三)進一步規劃和測試

在網站的各網頁和功能增加時，需要不斷的測試撰寫的新功能是否正常，以及外觀是否符合期望，css 有沒有跑掉，且不能只在 Internet Explorer 瀏覽器上測誼，也需要測試其它種類的瀏覽器，如 Mozilla FireFox 和 Google Chrome，其他瀏覽器在頁面上也需正確顯示其外觀和功能；如果功能不齊全或外觀差異過大，版面配置和網頁程式需重新規劃來改進。

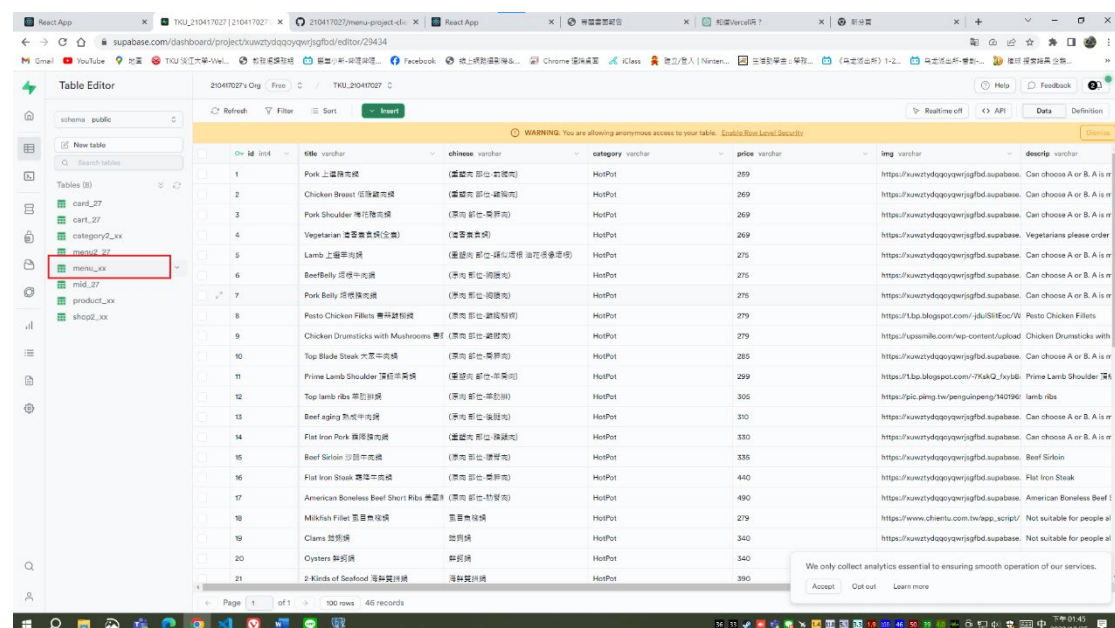
資料庫分面我們選擇使用 Supabase 來作為我們的資料庫，透過 pgadmin 連結到 supabase 的 host，在 pgadmin 下資料庫的指令，讓他上傳至 supabase 的 host，除了檢查其上傳資料是否正確進入指定的欄位，每當網站功能增加或改變，資料

庫的種類和各自欄位都需經過適當的修正，增加新的欄位或改變欄位的名稱，以免和新的需求不相符造成頁面上顯示錯誤。

1.將指令打在 pgadmin 上面(如下圖)



2.透過 pgadmin 把資料上傳至 supabase(如下圖)



(四)最後修正與美觀

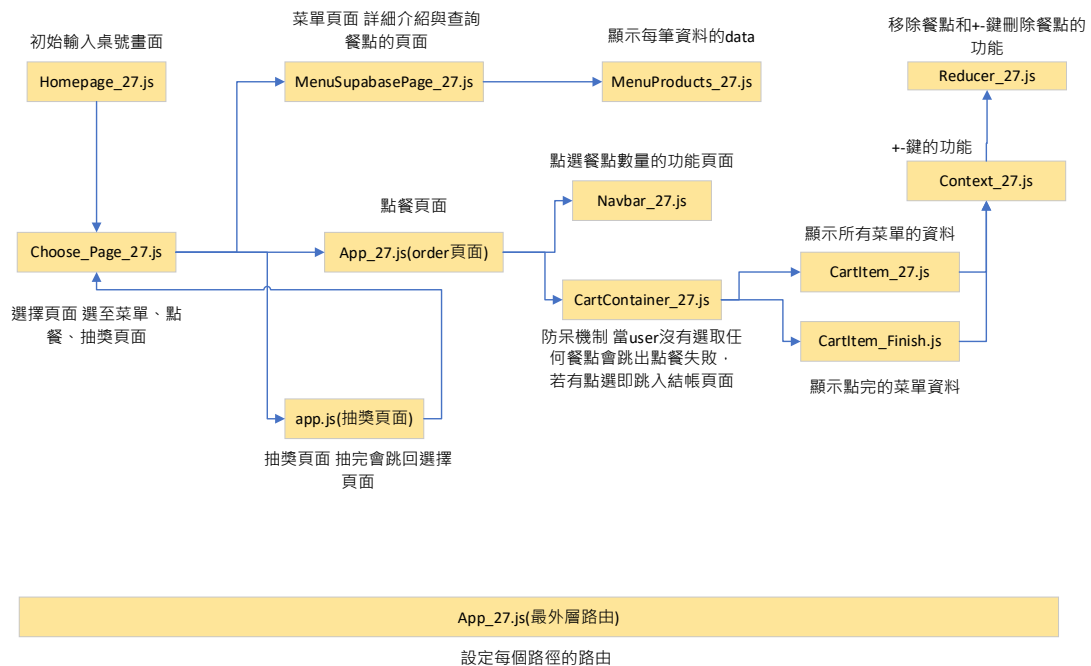
以上工作都完成後，網站的雛型已大致完成，剩下工作只需修正網站各網頁的功能增強、版面配置和外觀美化，以達到我們認為的最佳情形。

例如要把菜單跟點餐頁面弄成 2 個頁面還是要整合，如果弄成 2 個頁面要怎麼弄，如果菜單直接接點餐機頁面會不會讓一些熟客 user 感到麻煩，還是要多做一個頁面讓 user 直接選取要看菜單還是直接點餐，這都是我們考慮到的點，我們後來決定會做一個選擇的頁面，這個頁面可以讓 user 決定要看菜單，還是直接點餐，或者是說當 user 沒想法時進入到一個隨機抽選的頁面，用隨機抽選的方式來決定要點什麼。

其他方面目前我們在登錄畫面增加了一些特效，讓頁面看起來不要那麼單調，令使用者登入網站瀏覽時在視覺上有較鮮明的印象，除此之外我們在點餐頁面的最後新增了行動支付的功能，我們選擇使用了悠遊付(因其他行動支付都需要商家實名認證，故選擇悠遊付)，讓 user 在結帳時避免透過現金接觸，拿手機出來就可以付款了。

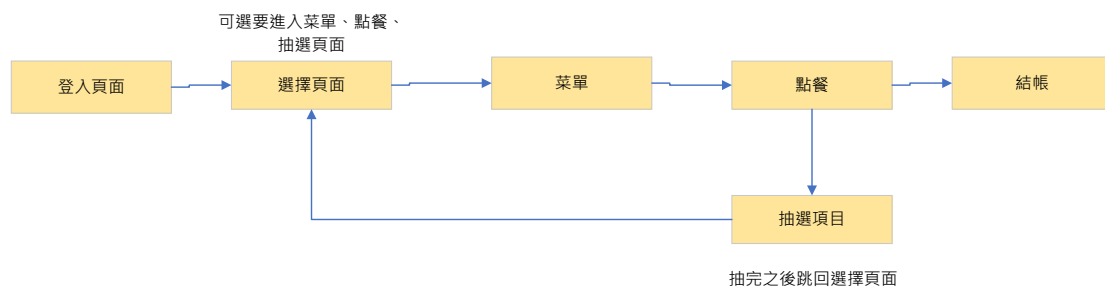
五、網站設計與流程

(一)整體網站架構



(二)網站簡易流程

登入頁面→選擇頁面→菜單→點餐→結帳(現金 or 行動支付)



六、結果與展示

(一)首頁:輸入桌號的頁面(是一個會動的動態頁面)



(二)選擇頁面:選擇進去菜單頁面、點餐頁面或抽獎頁面



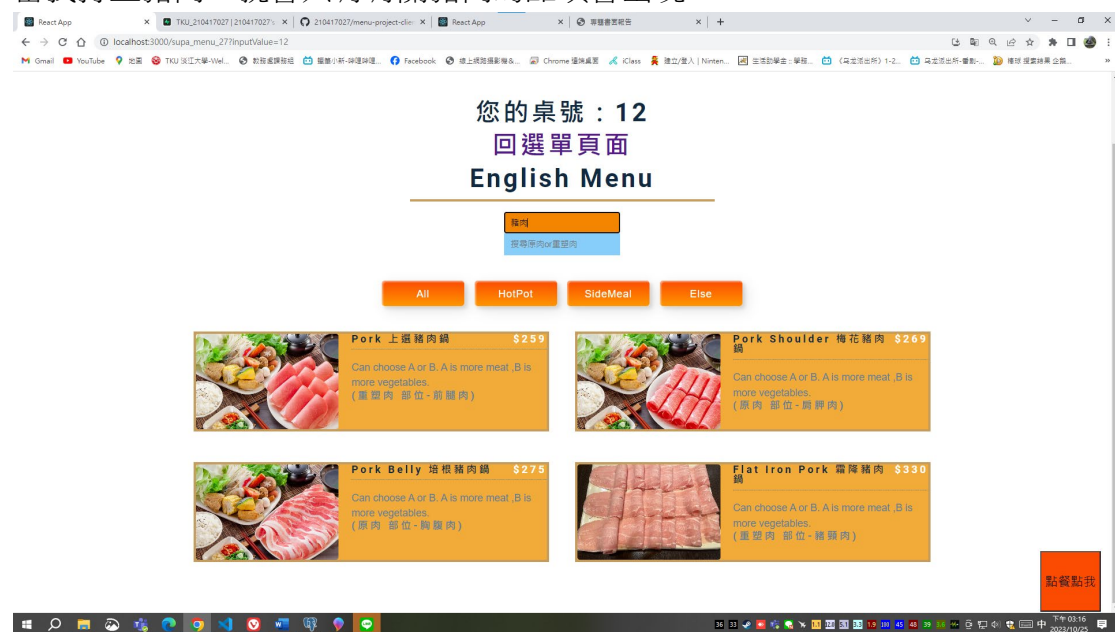
(三)菜單介紹頁面:包含查詢主餐、附餐、小菜功能以及搜尋單一肉品以及肉品是否為原肉或重組肉

1.下圖是菜單頁面一進來的樣子



2.搜尋欄功能(橘色一查詢肉品)

當我打上豬肉，就會只有有關豬肉的品項會出現



其他肉也可以，當搜尋欄打上牛，就會出現牛肉的結果(如下圖)



3.搜尋欄功能(藍色一查詢是原肉還是重組肉)

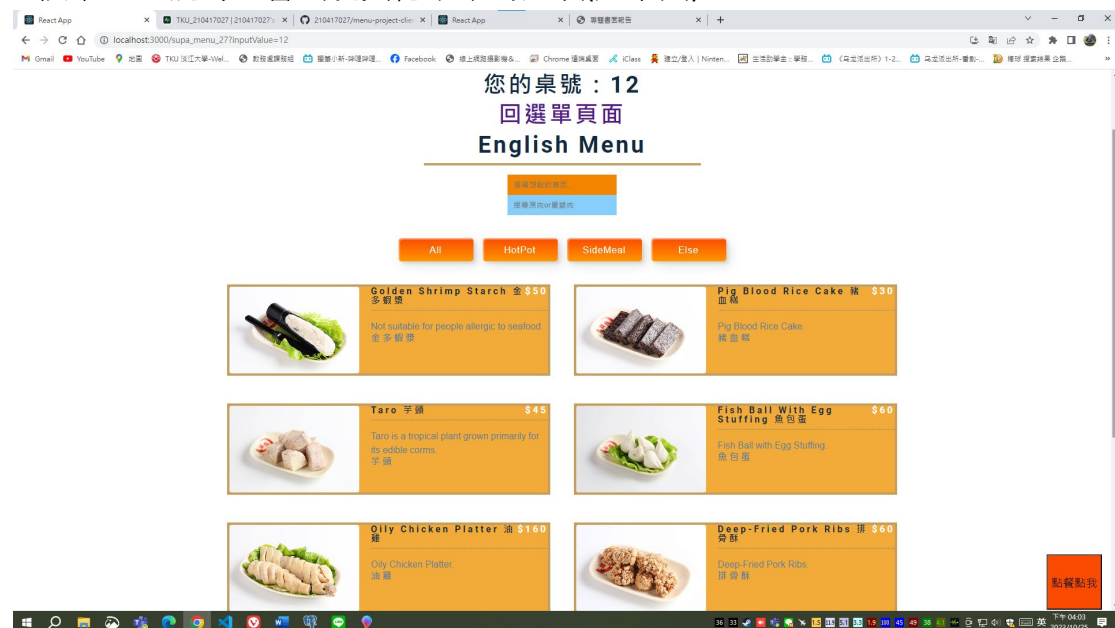
當我在藍色搜尋欄打上原肉，就會排除掉重組肉的選項，顯示原肉的選項(如下圖)



5.按下 sidemeal 鍵時，會出現所有附餐的選項(如下圖)



6.按下 Else 鍵時，會出現其他小菜的選項(如下圖)



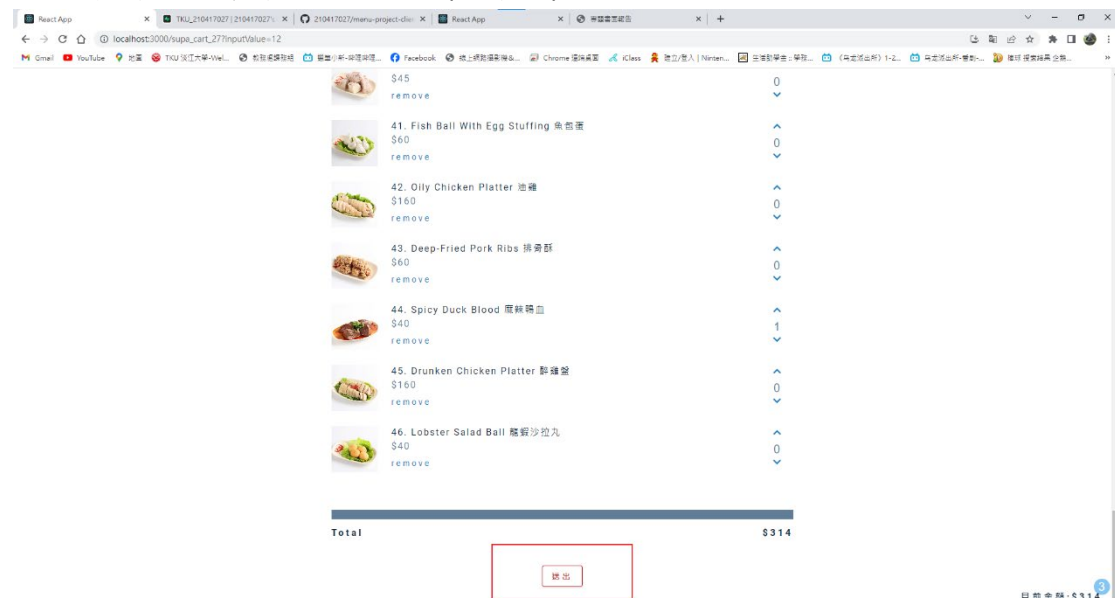
(四)點餐頁面:包含點餐份數、價格顯示等等



1. 當按下數字上方的按鍵，代表增加 1 份，反之下方按鍵為減少一份且當份數增減時，價錢也會跟著浮動(如下圖)



2. 當按下頁面最下方的送出鈕(如下圖)



會跳轉到結帳頁面，會顯示你點了什麼品項以及數量，並且會顯示價錢(如下圖)

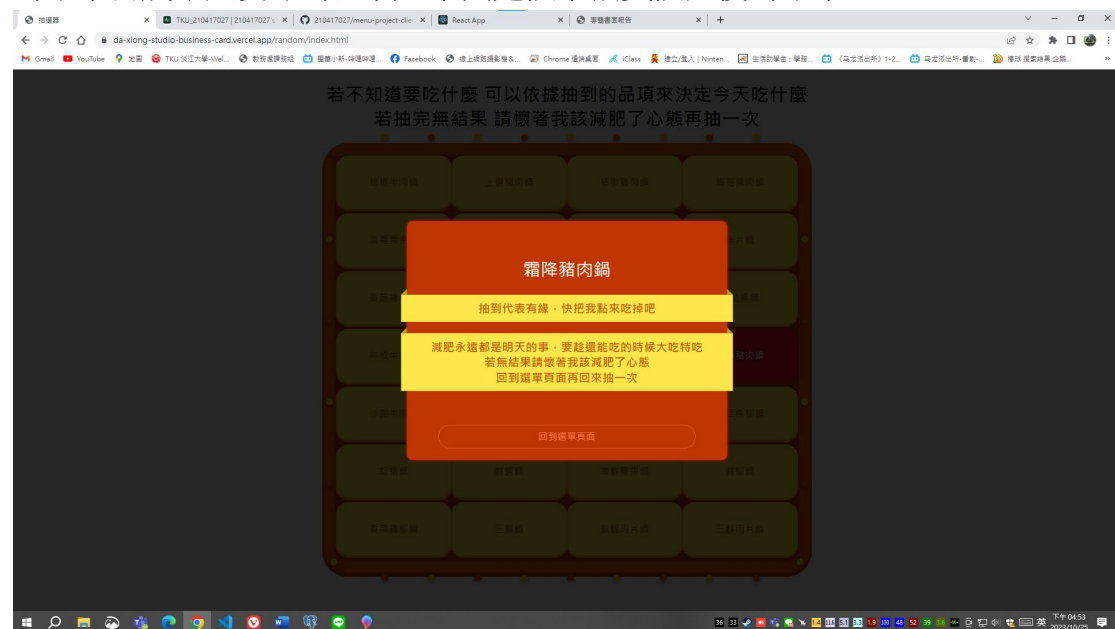


按下結帳去的按鈕就會跳出悠遊付付款的頁面(要用手機操作)

(四)隨機抽選頁面(如下圖)



按下立即抽獎的按鈕會開始隨機抽，抽到的結果可以讓 user 參考，可以把抽到的結果當成今天要點的品項，下圖是按下抽獎按鈕後的結果



有可能會沒結果(如下圖)



按下回到選單頁面按鈕就會回到



七、 結論

我們設計網頁的過程發現資料庫的資料表種類和欄位設置是一件要動腦筋且花不少功夫的部分，要決定有多少資料表和其要對應多少欄位除了取決於網頁表面的功能外，還要考慮到使用者的行為和其對應的解決方法。

其它如網頁美化、頁面顯示以及按鈕種類、位置和功能，都要看一下其他開發者的頁面，並且要有自己獨一無二的風格，如何在這當中取得平衡亦是一項挑戰。

此外，運用 **Github** 佈署到 **vercel** 對我們而言真的省了不少時間，解決了買伺服器、網域等等的問題，完全不需要額外付其他的費用，對我們而言真的蠻方便的。

八、 未來展望

目前我們的點餐機網站只要 **user** 的裝置有瀏覽器(**browser**)，都可以使用我們的網站，或許未來有機會可以開發 **app** 引入會員制度，讓功能用起來更完善，讓網站的功能更加的多元。

功能的擴充也是未來可以加強的部分，雖然目前網站的狀態要啟用已經是 **ok** 的狀態，但如果能加上一些特別的功能，例如說能加上和 **user** 互動的小遊戲之類的，相信這樣能讓網站內容更豐富。

當然我們還有許多可以改進的地方，撇除掉介面的部分，好比說之後可以在我們的網頁上加上文字機器人，當 **user** 對網站使用有疑問時可以直接做回應，以及行動支付方面可以增加大眾比較常用的那些，例如 **line pay**、街口支付之類的，想必這樣 **user** 用起來會比較方便，更能打造出一個方便的網站，這是我們之後可以朝這方面改進的地方。

九、參考網站

1. <https://www.evernote.com/shard/s16/client/snv?isnewsnv=true¬eGuid=724b9fca-f13e-89b5-ca00-a4d29302adfd¬eKey=7e0c0c133e0bf901fb9ef8ce0783d1b7&sn=https%3A%2F%2Fwww.evernote.com%2Fshard%2Fs16%2Fsh%2F724b9fca-f13e-89b5-ca00-a4d29302adfd%2F7e0c0c133e0bf901fb9ef8ce0783d1b7&title=%255B1101%255D%2B%25E5%25AE%2589%25E8%25A3%259D%2BVS%2BCode%2B%2526%2BGit%2B%2526%2BNode.js>
2. <https://www.evernote.com/shard/s16/client/snv?isnewsnv=true¬eGuid=f569c654-c8e8-bd35-6238-9b3657453f21¬eKey=27ad94999ac822e43239b99a04a3b7bc&sn=https%3A%2F%2Fwww.evernote.com%2Fshard%2Fs16%2Fsh%2Ff569c654-c8e8-bd35-6238-9b3657453f21%2F27ad94999ac822e43239b99a04a3b7bc&title=%255B110%255D%2BStack%2BOverflow%2B-%2BSurvey%2Bof%2BTrends%2B2021>
3. <https://www.udemy.com/course/design-and-develop-a-killer-website-with-html5-and-css3/>
4. <https://www.evernote.com/shard/s16/client/snv?isnewsnv=true¬eGuid=b1ad4950-6e4e-f429-bfdd-a886301c1696¬eKey=aedfa24f7f6a99ef50525b008eba14f9&sn=https%3A%2F%2Fwww.evernote.com%2Fshard%2Fs16%2Fsh%2Fb1ad4950-6e4e-f429-bfdd-a886301c1696%2Faedfa24f7f6a99ef50525b008eba14f9&title=%255B1101-1A%255D%2B%2BGithub%2BRepo%2BSetup%2Bas%2BPrivate%2BUsing%2BPersonal%2BAccess%2BToken>
5. <https://stackoverflow.com/questions/1441010/the-shortest-possible-output-from-git-log-containing-author-and-date>
6. <https://www.udemy.com/course/react-tutorial-and-projects-course/>
7. <https://zh.wikipedia.org/wiki/HTML>
8. <https://zh.wikipedia.org/zh-tw/JavaScript>
9. <https://zh.wikipedia.org/wiki/React>

10. <https://zh.wikipedia.org/zh-tw/PostgreSQL>
11. <https://zh.wikipedia.org/wiki/Git>
12. <https://zh.wikipedia.org/wiki/GitHub>
13. <https://medium.com/nosegates/5a92aafe23e4>
14. <https://www.w3schools.com/js/default.asp>
15. <https://www.w3schools.com/html/default.asp>
16. <https://www.w3schools.com/jquery/default.asp>
17. <https://www.w3schools.com/react/default.asp>
18. <https://www.w3schools.com/postgresql/index.php>
19. <https://www.w3schools.com/git/default.asp>
20. <https://chat.openai.com/>
21. <https://zh-hant.legacy.reactjs.org/tutorial/tutorial.html>
22. [https://developer.mozilla.org/zh-TW/docs/Learn/Tools and testing/Client-side JavaScript frameworks/React getting started](https://developer.mozilla.org/zh-TW/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started)
23. <https://medium.com/something-about-javascript/react-react-%E4%BB%8B%E7%B4%B9%E5%8F%8A%E7%90%86%E8%A7%A3-e7f90565350b>
24. <https://aws.amazon.com/cn/rds/postgresql/what-is-postgresql/>
25. <https://azure.microsoft.com/zh-tw/resources/cloud-computing-dictionary/what-is-postgresql/>

26. <https://medium.com/%E9%BA%A5%E5%85%8B%E7%9A%84%E5%8D%8A%E8%B7%AF%E5%87%BA%E5%AE%B6%E7%AD%86%E8%A8%98/%E7%AD%86%E8%A8%98-%E9%80%8F%E9%81%8E-jwt-%E5%AF%A6%E4%BD%9C%E9%A9%97%E8%AD%89%E6%A9%9F%E5%88%B6-2e64d72594f8>

十、github 和 vercel

Github URL:

<https://github.com/210417027/menu-project-client>

Vercel URL:

<https://menu-project-client.vercel.app/>