

Final Project Report

Group 6

112550031 陳炤宇、11255039 李宗諺、112550040 游子慶

112550044 吳宇藤、112550191 許書瑋

- Main idea

庫存管理是銷售業會面臨的重要課題，無論企業規模大小，都常常遇到缺貨、庫存過剩或是各類成本驟增的問題。因此，我們希望藉由這次的期末專題，深入練習如何有效進行庫存分析。我們將建立相關的資料庫、設計便於查詢的方式，以達成更有效的庫存管理，並提供上游廠商討論區對產品進行評論，進而模擬企業的營運效率的優化過程。

- Data

- i. Source

此專案使用的資料集來自 Kaggle，資料是來自一家酒類上游公司，內容包括酒品的庫存分布和下游廠商的訂單紀錄。庫存紀錄涵蓋了酒品的容量、品牌、價格及存放的倉庫位置；而在訂單紀錄部分，則記載了下游廠商的編號、名稱、訂單日期、提貨日期、付款日期和訂單金額等資訊。

Dataset 連結: [Inventory Analysis Case Study](#)

- ii. Our Data

因為我們希望打造一個可以搜尋酒品並顯示商業系統相關資訊的網站，因此選取了我們認為較具實用性的欄位，透過酒品名稱進行查詢，並與其他表格連結，最終匯總出「剩餘庫存、進貨資訊、銷售資訊」等完整的欄位。

- iii. Tables

2017PurchasePricesDec: 記錄酒品銷售的情況。

欄位有酒品代號(Brand)、酒品名稱(Description)、賣出價格(Price)、容量大小(Size)、買入價格(PurchasePrice)、購買公司編號(VenderNumber)、購買公司名稱(VendorName)

InvoicePurchases12312016: 紀錄購買的紀錄。

欄位有購買公司編號(VenderNumber)、購買公司名稱(VendorName)、發票開立日期(InvoiceDate)、訂單編號(PONumber)、訂單成立日期(PODate)、付款日期(PayDate)、訂購的量(Quantity)、訂購總額(Dollars)

PurchasesFINAL12312016:紀錄購買過程的詳細內容。

欄位有 資料編號(ID)、酒類代號(Brand)、InventoryID(Store+城市+Brand)、購買公司編號(VenderNumber)、購買公司名稱(VendorName)、訂單編號(PONumber)、發票開立日期(InvoiceDate)、付款日期(PayDate)、訂購的量(Quantity)、訂購總額(Dollars)

EndInvFINAL12312016: 紀錄最終(2016/12/31)的庫存情形。

欄位有 InventoryID(Store+城市+Brand)、倉庫編號(Store)、倉庫所在城市(City)、酒類代號(Brand)、酒品名稱(Description)、容量大小(Size)、及時剩餘庫存(onHand)、賣出價格(Price)

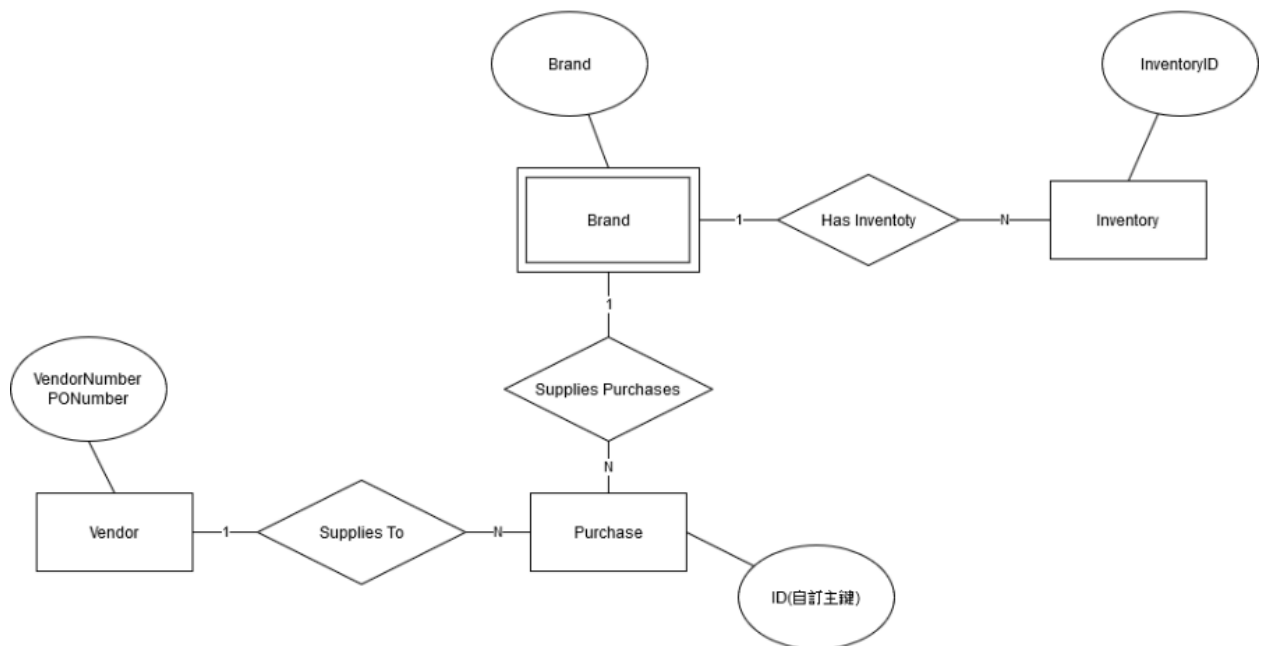
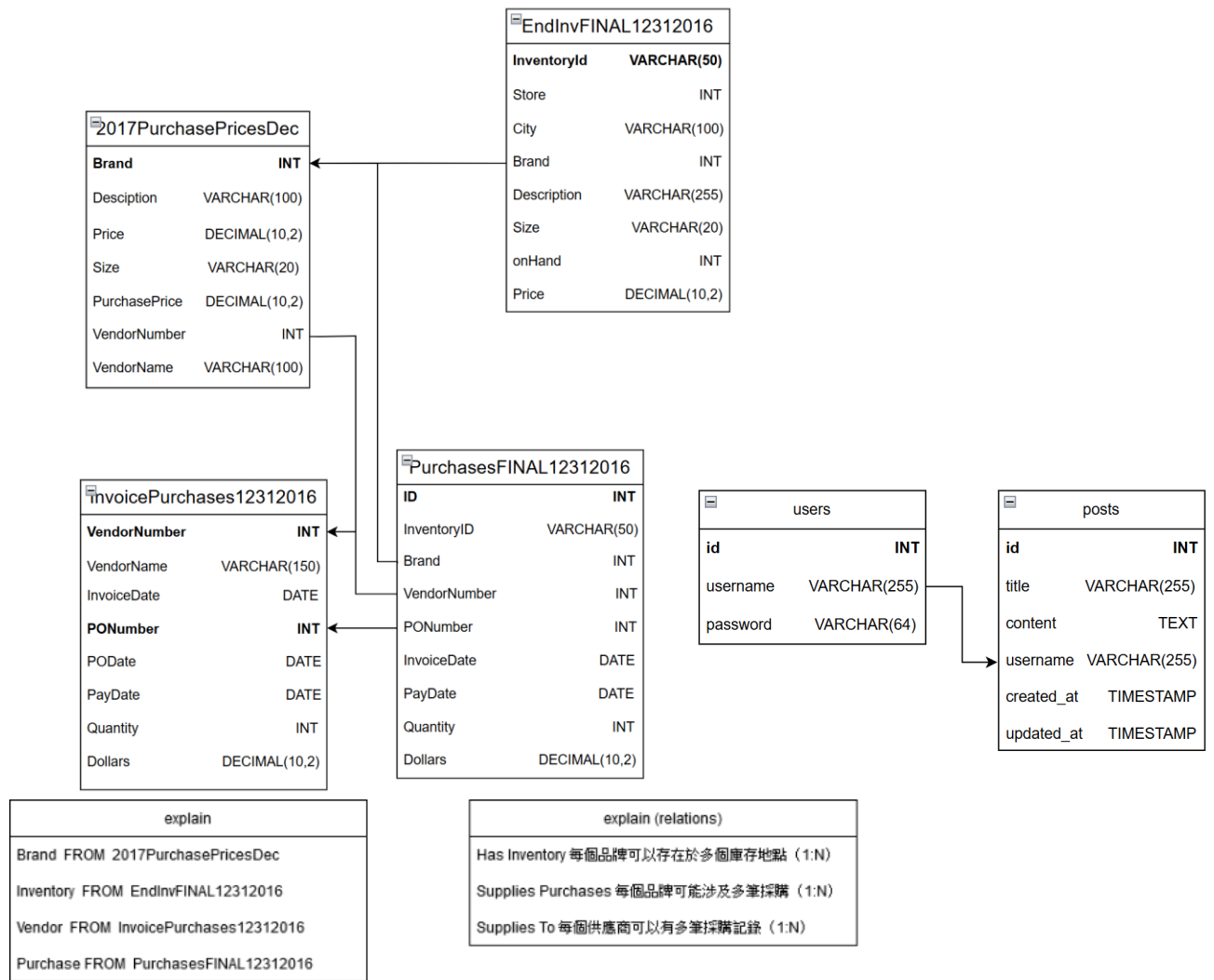
users:紀錄已註冊的網站使用者。

欄位有 身分編號(id)、使用者名稱(username)、密碼(password)

posts:紀錄評論區的內容。

欄位有 評論編號(id)、標題(title)、內文(content)、使用者名稱(username)、發布日期(created_at)、更新日期(updated_at)

iv. Schema/ER model



- Database

- i. Data Language

我們使用 mysql 當作後端，並用 python 的 flask 網路框架來完成前後端對接。

- ii. Data Maintance and Connect Database to the Application

- 新增資料

在新增資料方面，包括新增使用者與新增貼文，兩者皆使用相同的流程。應用程式透過連接資料庫，執行 MySQL 的 INSERT 查詢來完成資料新增操作，如下圖所示。

```
conn = get_db_connection()
cur = conn.cursor()
try:
    cur.execute("INSERT INTO posts (title, content,username) VALUES (%s, %s, %s)", (title, content,usname))
    conn.commit()
    flash('Post added successfully!', 'success')
except Exception as e:
    flash(f'Error: {str(e)}', 'danger')
finally:
    cur.close()
    conn.close()
```

- 更新資料

更新資料包含更新貼文，如下圖，當應用程式收到 POST 請求，也就是當使用者要提交修改後的貼文內容，就執行 sql 來更新指定 id 的貼文標題與內容；否則，當應用程式收到 GET 請求(此函式前面限制只能使用 POST 和 GET)，也就是網頁要載入欲編輯的貼文內容，就從資料庫查詢指定 id 的貼文資料並顯示至編輯頁面。

```
if request.method == 'POST':
    title = request.form['title']
    content = request.form['content']
    cur.execute("UPDATE posts SET title = %s, content = %s WHERE id = %s", (title, content, id))
    conn.commit()
    cur.close()
    conn.close()
    flash('Post updated successfully!', 'success')
    return redirect("/discussion")
else:
    cur.execute("SELECT * FROM posts WHERE id = %s", (id,))
    post = cur.fetchone()
    cur.close()
    conn.close()
    return render_template('edit_post.html', post=post)
```

- 刪除資料

刪除包含刪除貼文，如下圖，應用程式刪除指定 id 的貼文並更新資料庫，執行後顯示成功或錯誤訊息。

```

try:
    cur.execute("DELETE FROM posts WHERE id = %s", (id,))
    conn.commit()
    flash('Post deleted successfully!', 'success')
except Exception as e:
    flash(f'Error: {str(e)}', 'danger')
finally:
    cur.close()
    conn.close()

```

搜尋資料

搜尋功能為搜尋酒名跳出相關查詢解果，sql 和前後端方面這兩步驟比較長所以分兩段落講，先講 sql 方面，我們透過跑 sql 腳本來完成搜尋功能，由於 sql 腳本較長，因此這裡只用文字說明，實際 code 可以參考 search_script.sql 檔案。

此 SQL 腳本用於從四個表中提取酒品相關的完整資訊，並篩選符合特定酒品名稱的結果（如 "Belvedere Vodka"）。首先透過 `@inputDescription` 設定搜尋的酒名，然後使用 `JOIN` 和 `LEFT JOIN` 分別將 `EndInvFINAL12312016`、`2017PurchasePricesDec`、`PurchasesFINAL12312016` 以及 `InvoicePurchases12312016` 進行關聯。查詢的欄位包括剩餘庫存、城市、售價、進貨資訊以及發票支付日期。使用 `WHERE` 條件過濾關鍵字和非空值資料，確保結果完整且精準。結果可直接用於生成匯總表或進一步分析。

在前後端方面，我們透過輸入 `@inputDescription` 設定搜尋的酒名，並跑 sql 腳本，最後拿 sql 的結果來顯示，如下圖。

```

sql_script=sql_script.replace("SET @inputDescription = 'Belvedere Vodka';", f"SET @inputDescription='{description}'")

conn = get_db_connection()
cursor = conn.cursor(dictionary=True)
results = []

try:
    for statement in sql_script.split(';') :
        if statement.strip():
            cursor.execute(statement)
            results +=cursor.fetchall()
except Exception as e:
    flash(f'Error: {str(e)}', 'danger')
finally:
    cursor.close()
    conn.close()

```

iii. Exception Handling

此專案的每個函數都具備完整的檢查機制，能在更新失敗、連線失敗等異常情況下，於前端顯示對應的錯誤訊息。同時，所有操作均確保資料庫連線會被正確關閉，避免資源浪費，並且在出現錯誤時停止後續資料庫操作，確保數據完整性；在輸入方面也有防止 sql injection，透過使用下圖語法，讓使用者輸入被視為資料存放，而不是直接放進 sql 語法裡。

```
try:
    for statement in sql_script.split(';'):
        if statement.strip(): # ensure statement is non-empty.
            if "SET @inputDescription" in statement:
                # statement=> SET @inputDescription = %s;
                # description=> a search term from user
                cursor.execute(statement, (description,)) # parameterized input
            else:
                cursor.execute(statement)

    results = cursor.fetchall()
```

- **Application**

- a. **Main page**

此為我們網站進入後的第一個畫面，如圖 Fig. a1 所示。我們將我們的搜尋系統建立在首頁以方便，上方為導覽列，方便使用者前往討論區、關於我們以及登入系統。為避免使用者以訪客身分在此網站上進行查詢，所以我們會先判斷使用者的登入狀態，如果使用者未登入（即畫面右上角的狀態顯示 Log in），如圖 Fig. a2，我們會跳出請登入的訊息，並跳轉至登入頁面。已經登入的使用者，就會利用使用者輸入的資料進行搜尋，如果有搜尋到酒名，就會顯示酒類資訊，如圖 Fig. a3，如果沒有的話則跳出無結果，如圖 Fig. a4

CRUD: R:讀取使用者輸入的酒名

查詢應用：查詢使用者的登入狀態、利用使用者輸入的酒名進行查詢

錯誤處理：如果未登入就跳轉登入頁面

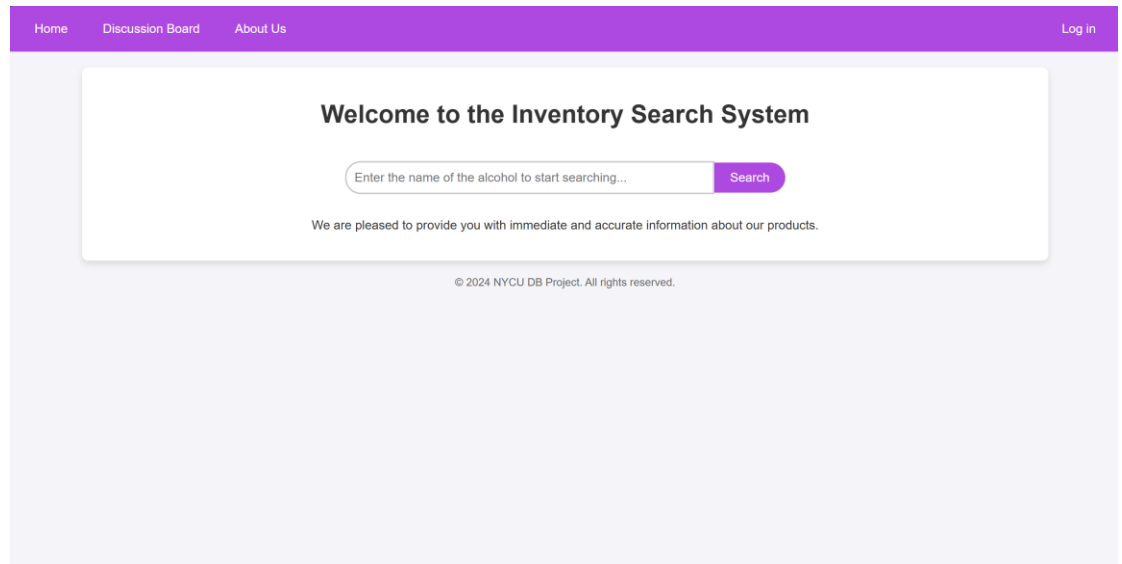


Fig. a1

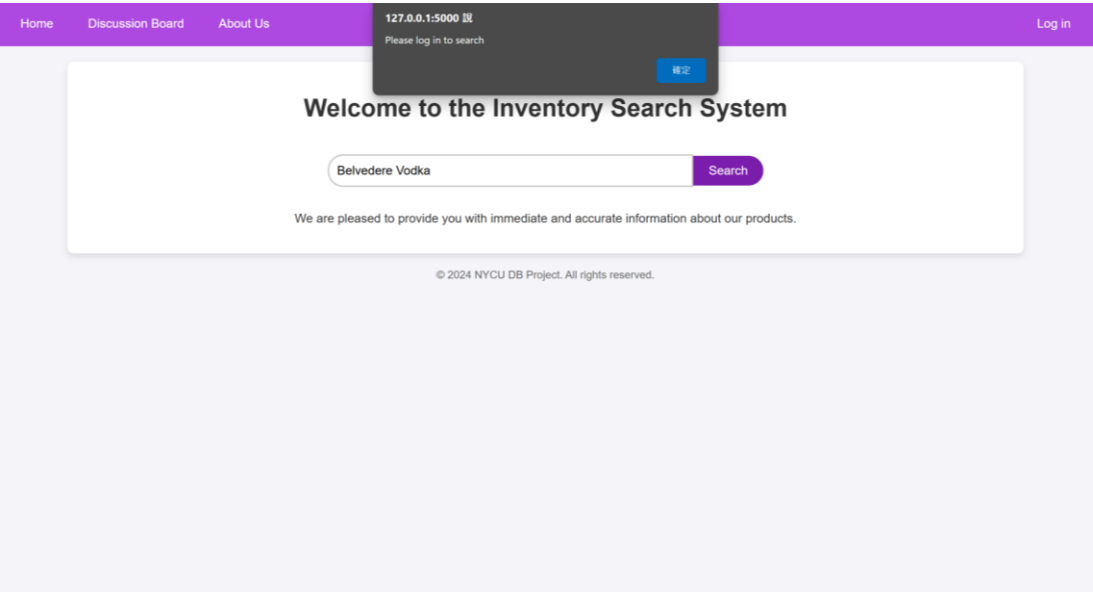


Fig. a2

Home

Discussion Board

About Us

Log out

Search Results

Results for "Belvedere Vodka":

Brand	On Hand	City	Sell Price	Vendor Name	Purchase Quantity	Purchase Dollars	Invoice Pay Date
3651	61	HARDERSFIELD	3.99	MOET HENNESSY USA INC "3652,"Ciroc Pineapple Vodka	1	2.93	2016-03-10
3682	51	HARDERSFIELD	22.99	MOET HENNESSY USA INC "3683,"Stolichnaya Ohranj	18	299.88	2016-02-06
3682	51	HARDERSFIELD	22.99	MOET HENNESSY USA INC "3683,"Stolichnaya Ohranj	5	83.30	2016-05-29
3682	51	HARDERSFIELD	22.99	MOET HENNESSY USA INC "3683,"Stolichnaya Ohranj	6	99.96	2016-09-07
3682	51	HARDERSFIELD	22.99	MOET HENNESSY USA INC "3683,"Stolichnaya Ohranj	5	83.30	2016-09-15
3682	51	HARDERSFIELD	22.99	MOET HENNESSY USA INC "3683,"Stolichnaya Ohranj	6	99.96	2016-09-16
3682	51	HARDERSFIELD	22.99	MOET HENNESSY USA INC "3683,"Stolichnaya Ohranj	6	99.96	2016-09-24
3682	51	HARDERSFIELD	22.99	MOET HENNESSY USA INC "3683,"Stolichnaya Ohranj	3	49.98	2016-11-06
3651	29	HORNSEY	3.99	MOET HENNESSY USA INC "3652,"Ciroc	60	175.80	2016-05-12

Fig. a3

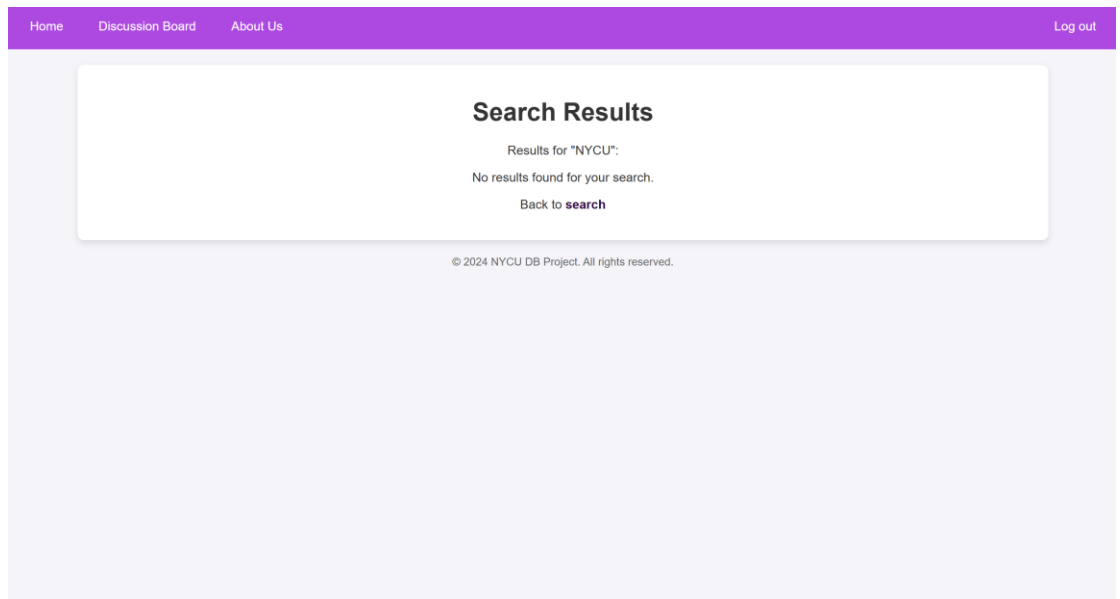


Fig. a4

b. Login page

由使用者點擊導覽列的登入或是由系統自動跳轉登入後，可進入至我們的登入系統，如圖 Fig. b1。此時帳號密碼為尚未輸入的狀態，藉由使用者輸入正確帳密後，即可進行登入，並跳轉回剛剛登入前的介面(比如如果是由討論區點擊登入且登入成功的話，系統會跳轉至剛剛討論區的畫面)，此時右上角的登入狀態從原先的 Log in 轉變成 Log out，如圖 Fig. b2。但如果輸入的帳密有誤的話，會跳出登入失敗的訊息，如圖 Fig. b3。

CRUD: R: 讀取使用者帳密 U: 更新使用者的登入狀態

查詢應用: 查詢使用者帳密是否在資料庫中

錯誤處理: 當使用者輸入錯誤的帳密，會跳出登入失敗的訊息

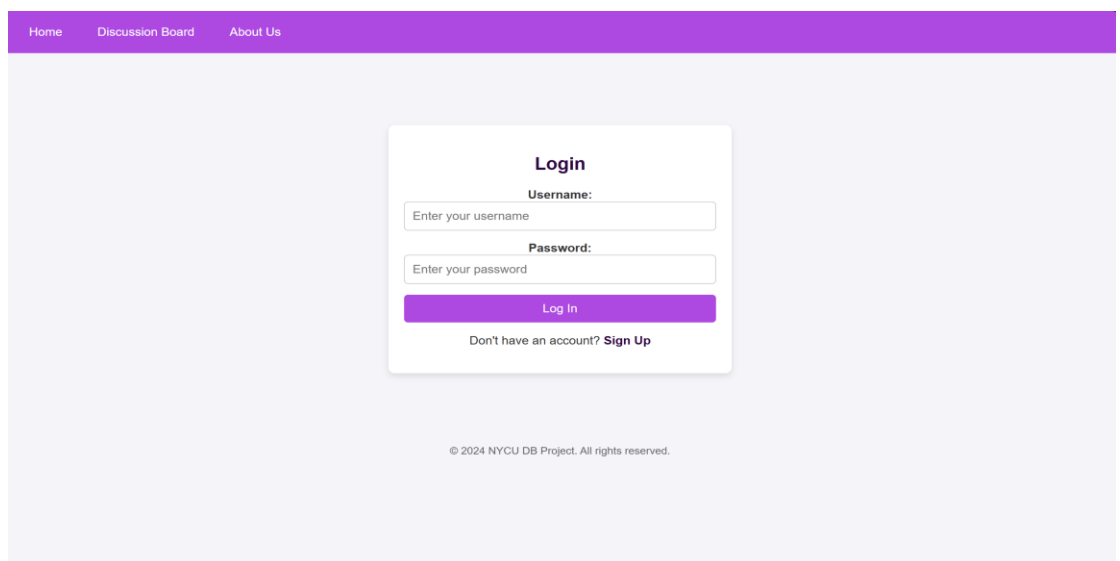


Fig. b1

The screenshot shows a web application with a purple header bar containing links for 'Home', 'Discussion Board', 'About Us', and 'Log out'. The main content area features a white card titled 'Create a new post'. This card has a 'Title:' label above a text input field, a 'Content:' label above a larger text area, and a purple 'Post' button. Below the card is a section titled 'Discussion Board' with the text 'Feel free to share your thoughts!'. At the bottom of the page, a copyright notice reads '© 2024 NYCU DB Project. All rights reserved.'

Fig. b2

The screenshot shows the same web application with the purple header. The main content area features a white card titled 'Login'. It contains a 'Username:' label above an input field with placeholder text 'Enter your username', and a 'Password:' label above an input field with placeholder text 'Enter your password'. Below these is a purple 'Log In' button. Under the button, there is a link: 'Don't have an account? Sign Up'. At the bottom of the card, a red error message is displayed: 'Invalid username or password'. The footer contains the same copyright notice: '© 2024 NYCU DB Project. All rights reserved.'

Fig. b3

c. Sign up page

如果使用者沒有帳號的話，可以從 Log in page 點擊下方的 Sign up，並跳傳至 Sign up page，如圖 Fig. c1 所示。在此我們可以創立我們的帳號密碼，如果創立成功，會自動跳轉至 Log in page，並跳出創立成功的訊息，使用者只須輸入剛剛設立的帳密即可登入，如圖 Fig. c2 所示。但如果帳號已經創立了，且使用者嘗試使用同個名字在創建的話，系統會跳出創建失敗的訊息，如圖 Fig. c3 所示。

CRUD: C 創建使用者帳密於資料庫中 R 讀取使用者輸入的帳密

查詢應用: 查詢資料庫的所有帳號

錯誤處理: 如果查詢到已經有重複的帳號，跳出創建失敗的訊息

[Home](#) [Discussion Board](#) [About Us](#)

Sign Up

Username:

Password:

[Sign Up](#)

Already have an account? [Back to Login](#)

© 2024 NYCU DB Project. All rights reserved.

Fig. c1

[Home](#) [Discussion Board](#) [About Us](#)

Login

Username:

Password:

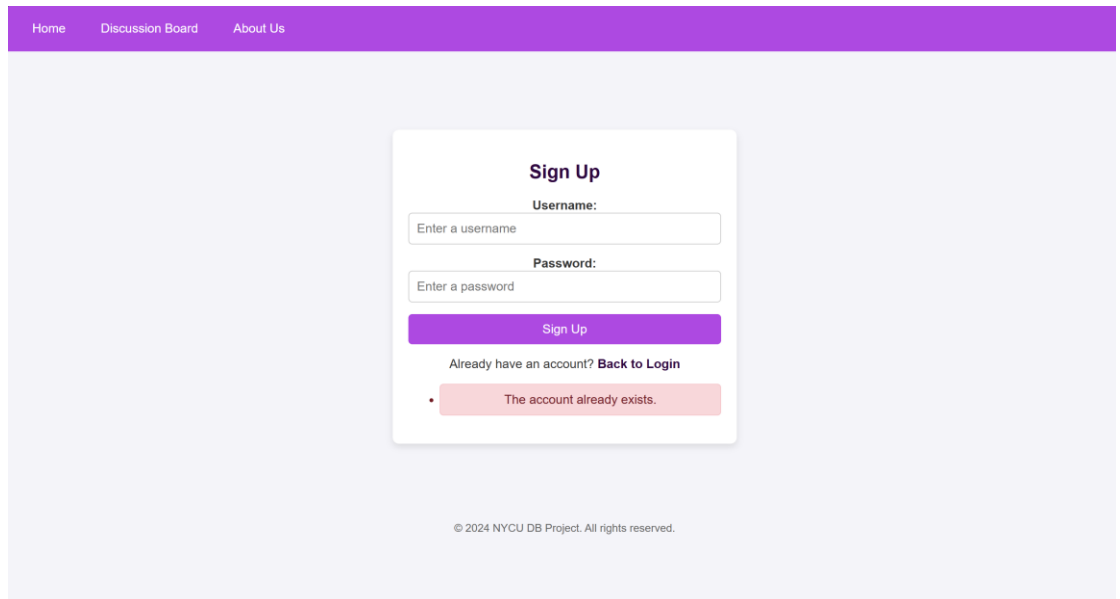
[Log In](#)

Don't have an account? [Sign Up](#)

- Account created successfully! Please log in.

© 2024 NYCU DB Project. All rights reserved.

Fig. c2



The image shows a web page with a purple header containing links for Home, Discussion Board, and About Us. The main content area is light purple and features a white 'Sign Up' form. The form has two input fields: 'Username:' with the placeholder 'Enter a username' and 'Password:' with the placeholder 'Enter a password'. Below these is a purple 'Sign Up' button. Under the button, it says 'Already have an account? [Back to Login](#)'. At the bottom of the form, there is a red error message: 'The account already exists.'.

Fig. c3

d. Discussion board page & edit post page

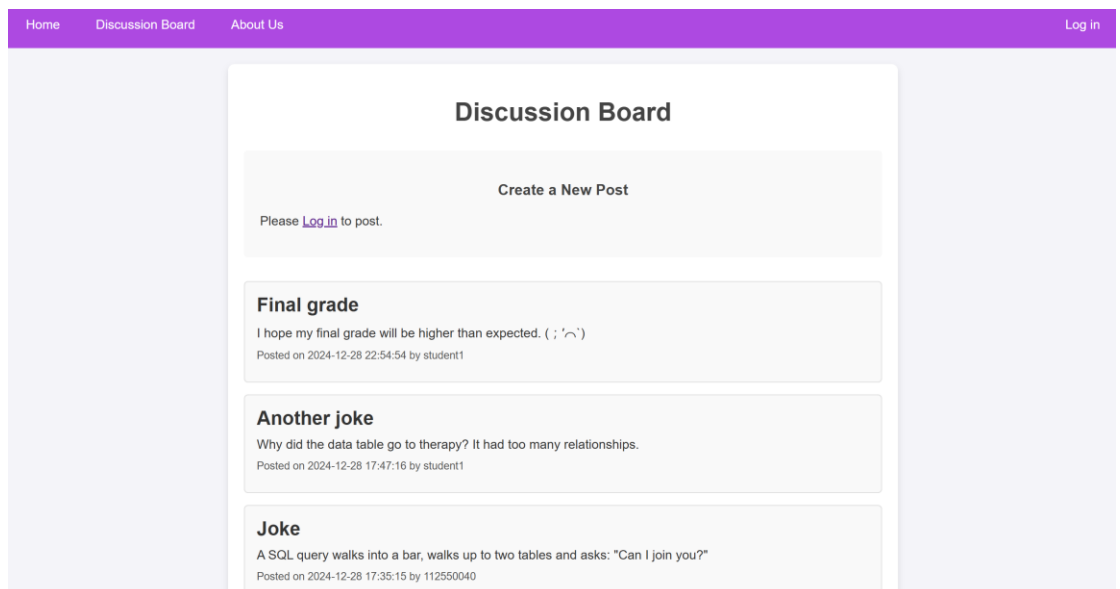
i. 初始畫面(未登入)

使用者點選導覽列上的 Discussion board 即可前往討論區。其中未登入的使用者無法對於討論區做任何的動作，僅可察看討論區上的貼文，如圖 Fig. d1。

CRUD: R:讀取 post 的資料

查詢應用: 查詢所有 post 裡的貼文，並顯示在網頁上

錯誤處理: 對於未登入的使用者，限制其操作並僅允許察看貼文



The image shows a web page with a purple header containing links for Home, Discussion Board, and About Us, and a 'Log in' link on the right. The main content area is light purple and features a white 'Discussion Board' section. At the top of this section is a button that says 'Create a New Post'. Below this is a message: 'Please [Log in](#) to post.' Below this message are three post cards. The first card is titled 'Final grade' and contains the text 'I hope my final grade will be higher than expected. (; '¬`)' and 'Posted on 2024-12-28 22:54:54 by student1'. The second card is titled 'Another joke' and contains the text 'Why did the data table go to therapy? It had too many relationships.' and 'Posted on 2024-12-28 17:47:16 by student1'. The third card is titled 'Joke' and contains the text 'A SQL query walks into a bar, walks up to two tables and asks: "Can I join you?"' and 'Posted on 2024-12-28 17:35:15 by 112550040'.

Fig. d1

ii. 初始頁面(登入後)

當使用者登入後，討論區中的創建貼文就開放使用，如圖 Fig. d2。
使用者可以在討論區中輸入該貼文的標題、內容，輸入完成後按下送出即在資料庫創建好了貼文內容，並顯示在網頁上面，如圖 Fig. d3 與 Fig. d4。

CRUD : C:使用者可以自行創造貼文 R: 當使用者創造完貼文後可以顯示在網頁上

查詢應用：查詢資料庫現有的貼文

錯誤處理：使用者不得送出空標題與空內容

The screenshot shows a web application interface for a Discussion Board. At the top, there is a purple navigation bar with links for 'Home', 'Discussion Board', and 'About Us', and a 'Log out' button on the right. The main content area is titled 'Discussion Board'. Below the title, there is a section for 'Create a New Post' with a 'Title:' label and an input field, a 'Content:' label and a larger text area, and a purple 'Post' button. Below this, there are two existing posts. The first post is titled 'Final grade' and contains the text 'I hope my final grade will be higher than expected. (; 'ㄟ')'. It is dated 'Posted on 2024-12-28 22:54:54 by student1' and has 'Edit' and 'Delete' links. The second post is titled 'Another joke'.

Fig. d2

This screenshot is similar to Fig. d2, showing the same Discussion Board interface. However, the 'Create a New Post' form now contains sample data: the 'Title' field has 'Help!!!' and the 'Content' field has 'How can I get lots of vodka?'. The 'Post' button remains purple. The existing posts, 'Final grade' and 'Another joke', are identical to those in Fig. d2.

Fig. d3

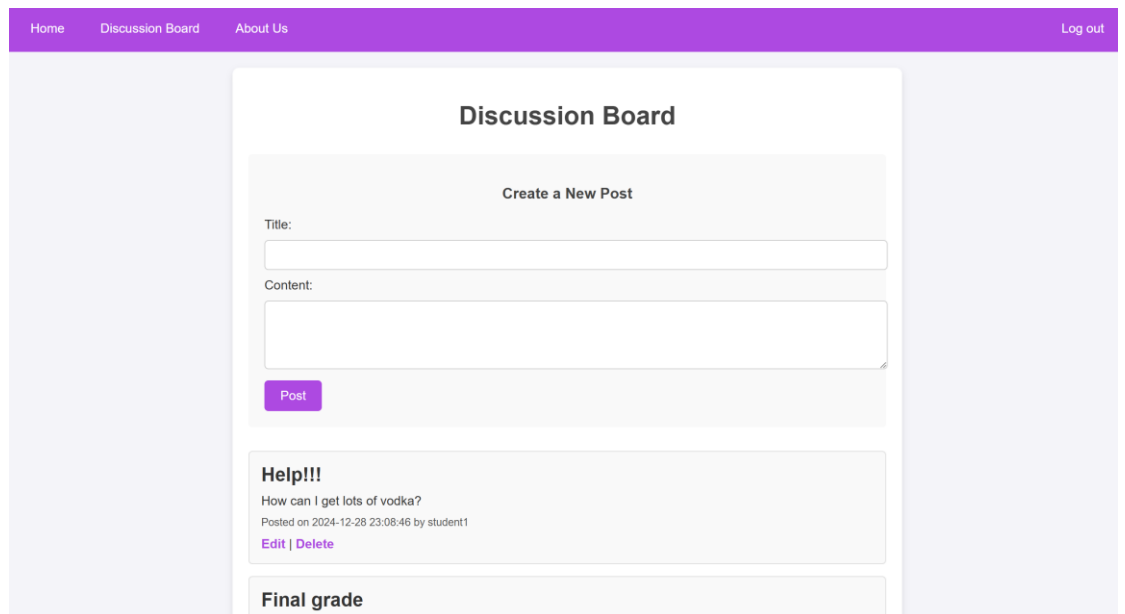


Fig. d4

iii. 編輯頁面

在討論區中，使用者只能針對自己的文章進行編輯，如圖 Fig. d5。如果想要編輯文章的話，僅需點擊該文章中的 Edit，畫面會跳轉至編輯頁面，如圖 Fig. d6。此時使用者如果編輯完標題、內容後，按下 Update，畫面就會跳轉回剛剛的討論區的內容，而貼文也會隨之更新為使用者剛剛更新的內容，如圖 Fig. d7。

CRUD: R:讀取文章現有的內容以及使用者更新的內容 U:將新的內容更新於資料庫中

查詢應用: 查看討論區的文章，系統須從資料庫中查詢

錯誤處理: 更新後的內容不能為空

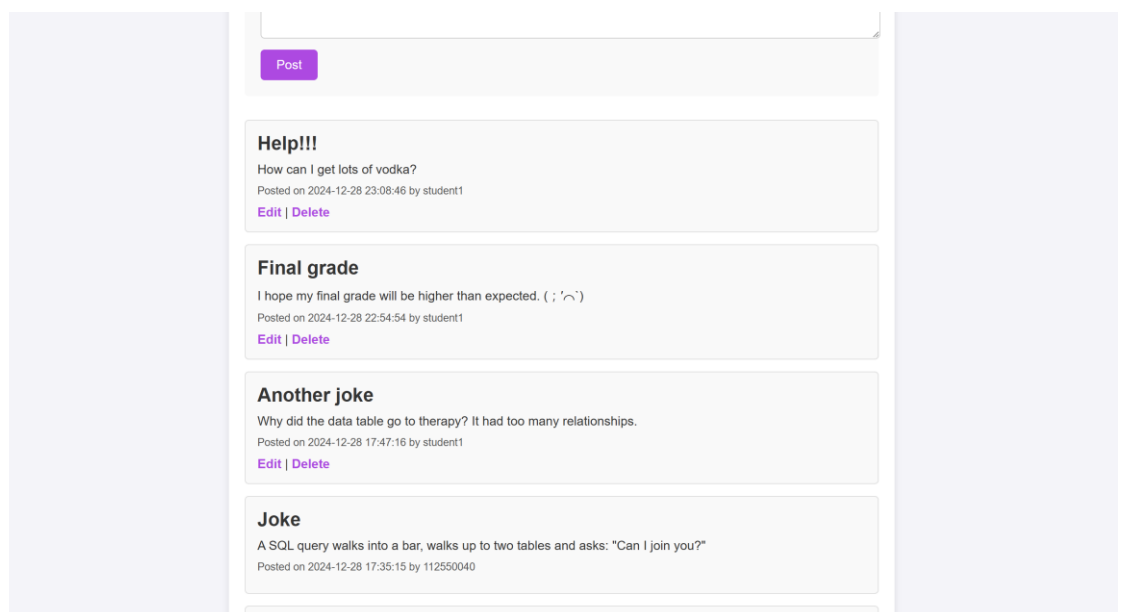


Fig. d5

Fig. d6

Fig. d7

iv. 刪除頁面

與編輯頁面相似，使用者僅可以刪除自己的文章，故使用者可以點擊文章的 delete 按鈕，點擊完後資料庫裡的文章就會被刪除，網站的文章內容也不會顯示該文章，如圖 Fig. d8 (刪除前)與 Fig.d9 (刪除後)。

CRUD: D:使用者點擊 delete 就可以刪除資料庫中的對應文章 R:使用者刪除完成後，系統會更新並顯示剩餘文章

查詢應用:系統需要查詢資料庫中的現有文章

錯誤處理:無

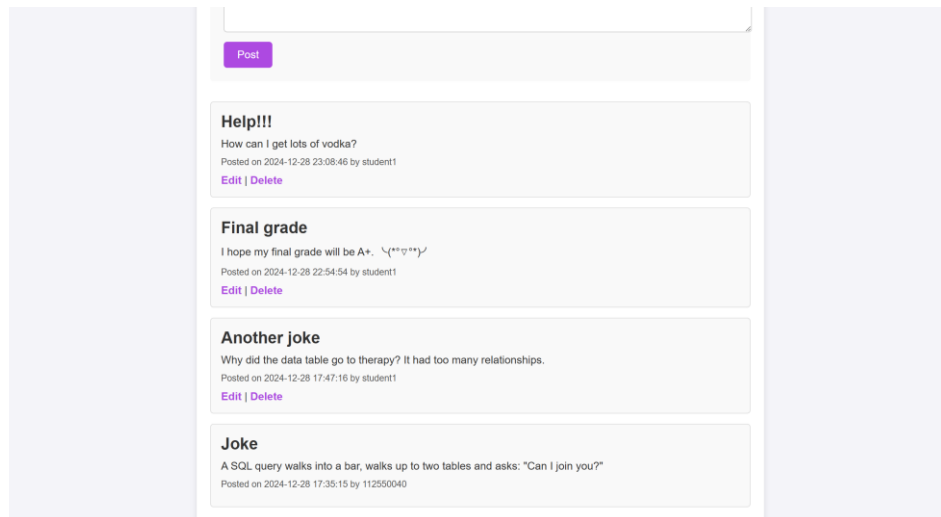


Fig. d8

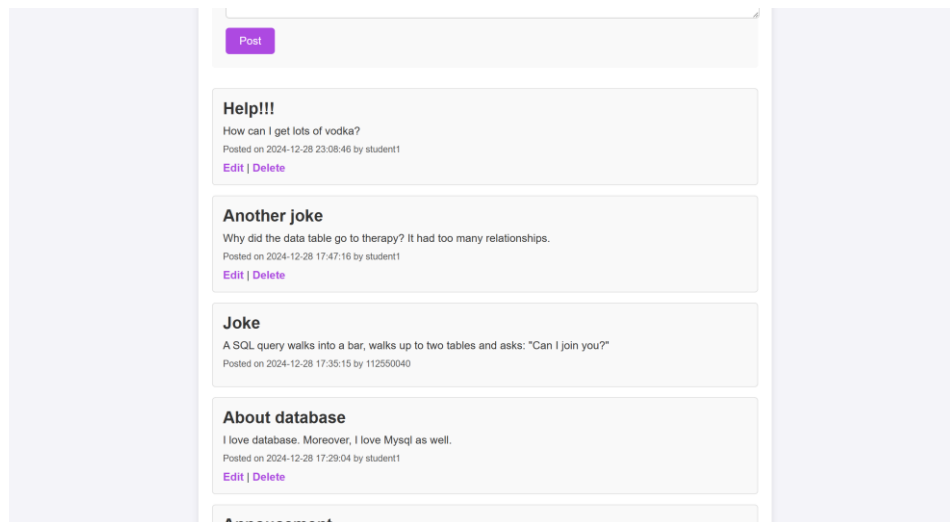


Fig. d9

e. About us page

這邊僅做簡單介紹，沒有特別的功能，如圖 Fig. d10

CRUD:無 查詢應用:無 錯誤處理:無

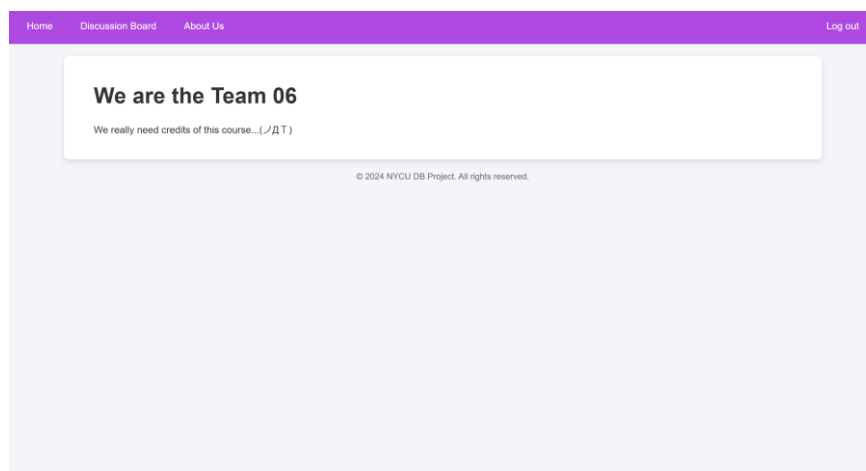


Fig. d10

- Attachment

Link to project code:

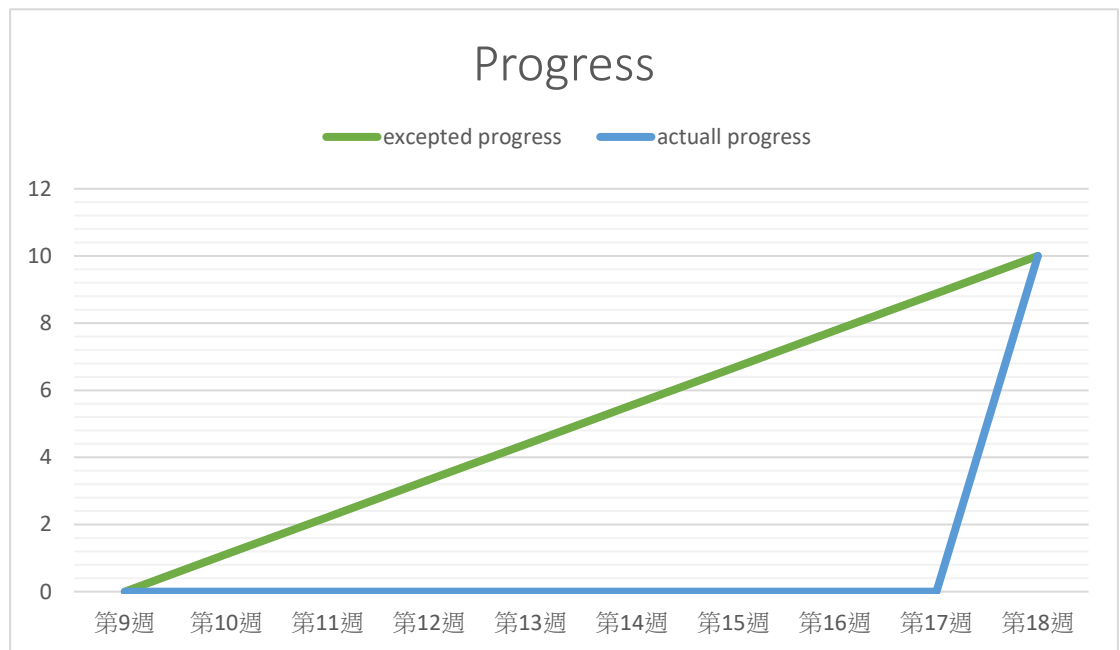
<https://github.com/mikehsuhoodie/NYCU-Intro-to-Database>

Link to video:

<https://youtu.be/-gNprZvhTAc>

- Other problems

- a. Draw the progress of your project



- b. Problems we met

- i. 如何結合資料庫與網頁設計出一個專案?
我們這組使用 Mysql 搭配 Python+flask 做前後端連結，並透過 git 進行共同編程，達成多人實作專案的目的。
 - ii. 部分的 csv 檔的項目無法找到最有效的複合主鍵?
我們經過討論後決定自創一個遞增數列用來辨別項目
 - iii. 原本的資料集並未描述各個欄位，我們並不知道每個欄位所代表意義是甚麼?
我們經過多方查詢、翻閱商業領域的文章，逐漸了解各欄位的意義
 - iv. 部分資料在 csv 檔中的格式出現換行,空白,非可見字符等情況，導致 MySQL CLI 的表格輸出對齊混亂
在 SELECT 時使用 REPLACE(REPLACE(資料名, '\r', ''), '\n', ''), 在查詢時清理字串
 - v.

c. Contribution of each team member

陳炤宇	繪製 schema、網站偵錯、撰寫報告
李宗諺	資料導入與查詢、後端製作、撰寫報告
游子慶	網頁架構、前端、後端製作與對接、撰寫報告
吳宇藤	撰寫報告、功能偵錯、資料分析
許書瑋	管理 github、前後端對接、網頁美化、撰寫報告