112-2 商管程期末專案 - 第九組書面報告

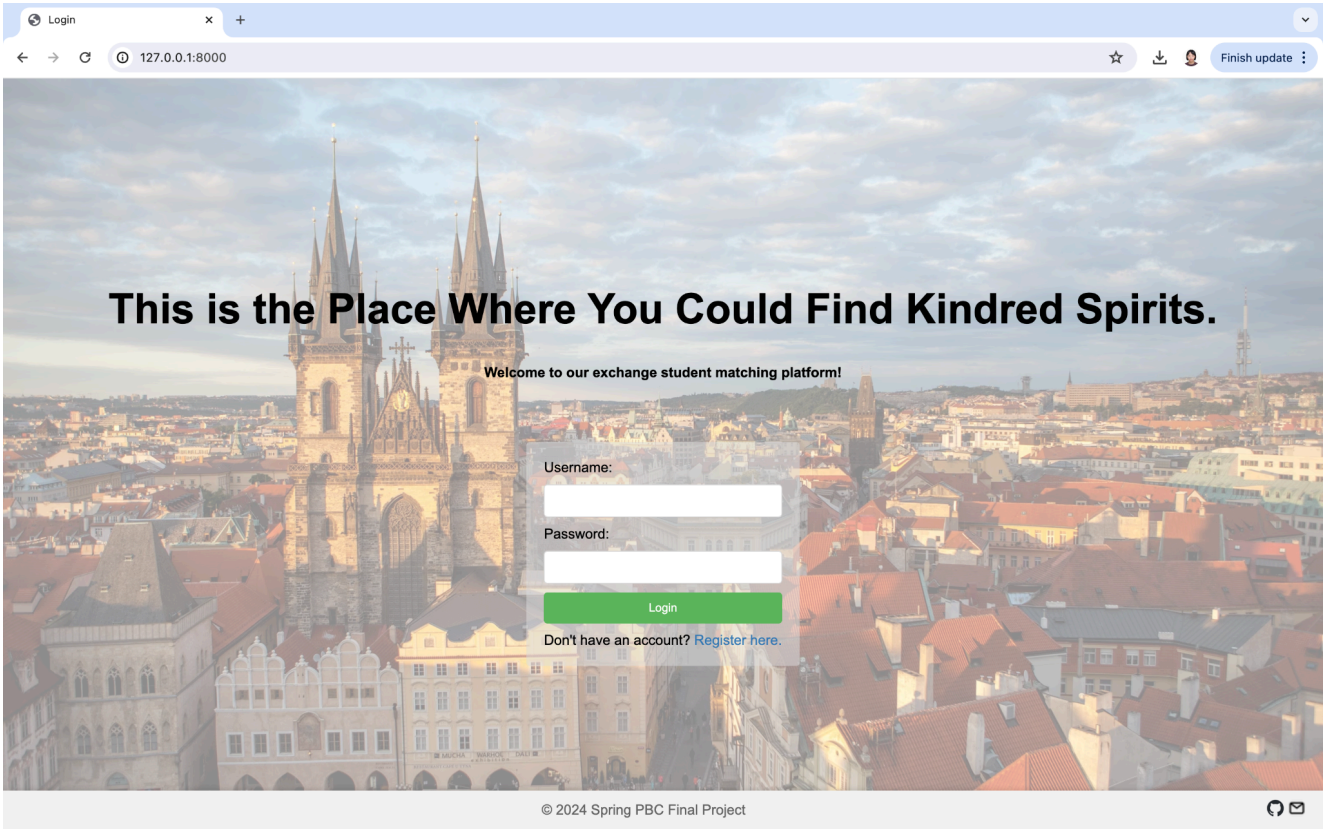
成員 : B09602017 白宗民、B09401140 劉裕緯、B08612024 徐子涵、ntnu_40884231i 鄭若妤

Github 連結: https://github.com/Bai1026/PBC_Final_Project

主題簡介 － 交換生媒合網頁

“This is the place where you could find kindred spirits.”

Matching Platform 是一個專為出國交換學生所設計的媒合平台。類似於交友軟體的操作模式，用戶可以先註冊並且在平台上瀏覽其他留學生的個人檔案，尋找合適的交流對象。平台會根據隱藏次數和演算法所計算的推薦分數來進行檔案排列，並提供篩選功能，幫助用戶找到最佳配對。而成功配對後，用戶可獲得朋友的聯繫資訊，促進交流和友誼，使得交換生們可以更輕易的找到旅伴或是一個依靠，我們立志成為他們尋找避風港的接口！



靈感來源

“Is there a more efficient way to connect with other exchange students?”

身邊有許多同學選擇使用 Facebook 上的社團尋找國外留學的伙伴，但社團貼文雜亂、容易遺漏且不容易篩選，總是要把全部看過才可能找到一個適合自己的旅伴或是朋友，我們希望能有更有效率的平台。因此，我們計畫創建一個新的 交換生 Matching 平台，提供更有效的配對模式，幫助留學生找到志同道合的交流夥伴。並且有真實的計畫deploy給大家使用，使這個專案不再只是一個課內專案。

程式設計說明

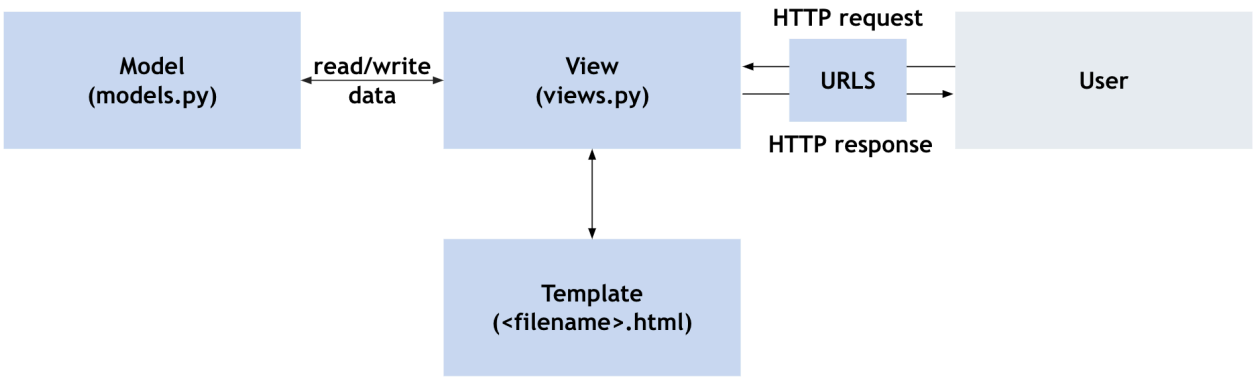
系統架構簡介

一、Django 概述

交換生媒合網頁由 Django 所建構。Django 是一個高級 Python 網頁框架，旨在快速開發和乾淨、實用的設計。Django 遵循 MTV（Model-Template-View）架構，其中：

- Model 負責數據層，定義數據庫結構和業務邏輯。
- Template 負責表示層，定義如何與HTML合作。
- View 負責處理業務邏輯和與前端互動的頁面顯示。

Apps內部運作模式	
Models	數據庫管理（添加，修改，刪除）
Templates	使用者會接觸到的頁面（HTML, CSS）
View	Views 接收 HTTP request，並返回 HTTP 響應給 Templates
URLs	將 HTTP request 定向到相對應的 Views



Django 提供了一系列內建的功能，如 ORM、管理界面、表單處理、中間件、測試框架等，幫助開發者快速構建高效的網頁應用。

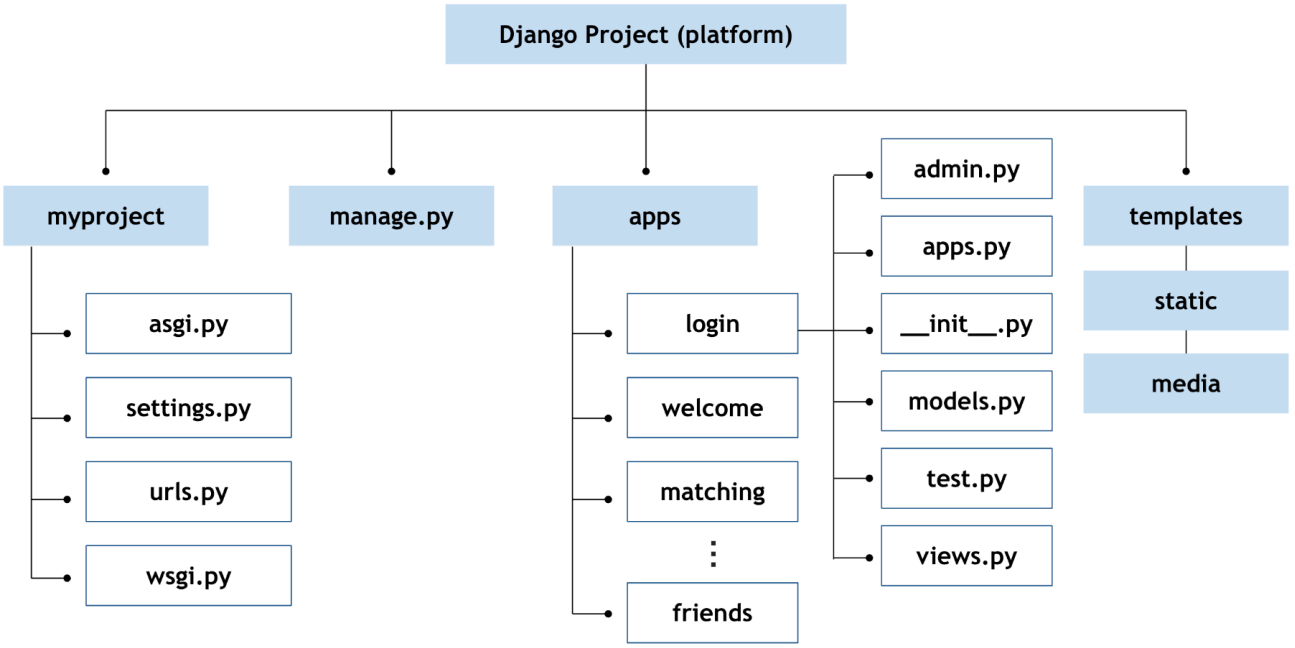
二、項目結構

交換生媒合網頁項目包含以下主要應用（apps）：

- **Login**：處理用戶認證和授權。
- **Welcome**：顯示歡迎頁面和相關資訊。
- **Matching**：處理交換生的媒合邏輯。
- **Friends**：管理交換生之間的好友關係。

系統架構詳細說明

一、目錄結構

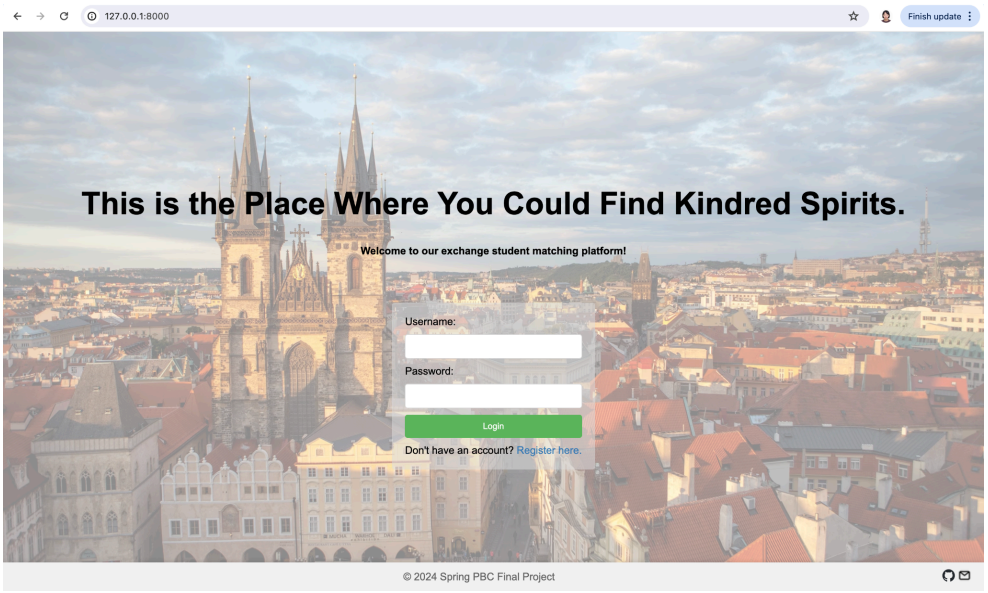


Architechture	
└── **platform** ├── **myproject**: **main folder contains settings and urls** ├── **avatars**: store the avatars for each user ├── **error_handlers**: folder for error handling ├── **friends**: folder for all the friends data ├── **login**: folder for login page ├── **matching**: folder for welcome page ├── **templates**: folder for all the HTML files ├── **welcome**: folder for welcome page ├── **db.sqlite3**: store the back-end data with sqlite ├── **manage.py**: the code to start the website ├── **migrate.sh**: shell script for migrate └── **README.md**	└── **each app**(page) folder ├── __init__.py ├── admin.py ├── apps.py ├── models.py ├── tests.py ├── urls.py ├── views.py └── ...

二、主要文件

- **myproject/settings.py**：配置文件，定義數據庫設置、已安裝的應用、模板路徑等。
- **myproject/urls.py**：主 URL 配置文件，將各應用的 URL 包含進來。
- **manage.py**：Django 的命令行工具，用於執行各種管理命令，如運行伺服器、遷移數據庫等。

三、頁面說明

<table>
<tr><th colspan="3">Login</th></tr>
<tr><td>畫面</td><td colspan="2"> 
</td></tr>
<tr><td>功能簡介</td><td colspan="2">負責用戶的註冊、登錄。forms.py 收集和驗證用戶輸入，並使用views.py 來處理相關的業務邏輯，確保用戶可以順利地註冊、登錄並更新其個人資料。models.py裡面有classes 負責存儲用戶信息和相關的關係數據，如推薦分數、媒合邀請。</td></tr>
<tr><td rowspan="2">Pseudocode</td><td>forms.py</td><td>class UserRegistrationForm:
Fields: username, password, password2, …, avatar2, avatar3
class Meta
def clean_password: # check password == password2
def clean: # additional validation (start_date < end_date)

class UserProfileForm:
Fields: username, password, password2, …, avatar2, avatar3
class Meta</td></tr>
<tr><td>models.py</td><td>class UserProfile:
Fields: username, password, password2, …, avatar2, avatar3
Custom property: friends

class RecommendationScore:
user_from, user_to, score
class Meta

def create_or_update_user_profile:
# When a new user created, a new UserProfile object is created
# When created=False, the existing userprofile object is saved

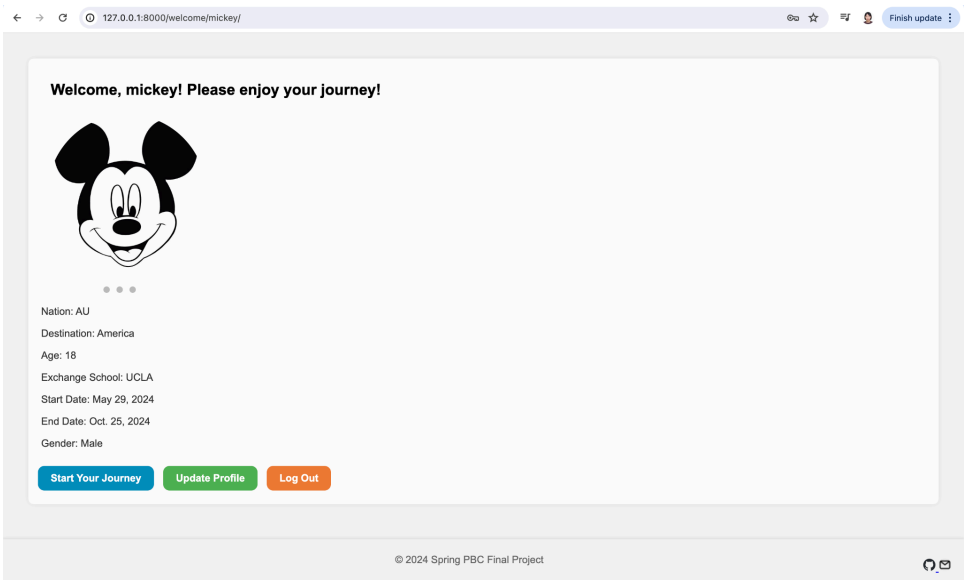
class HiddenProfile:
user (hiding user), hidden_user, hide_count
class Meta

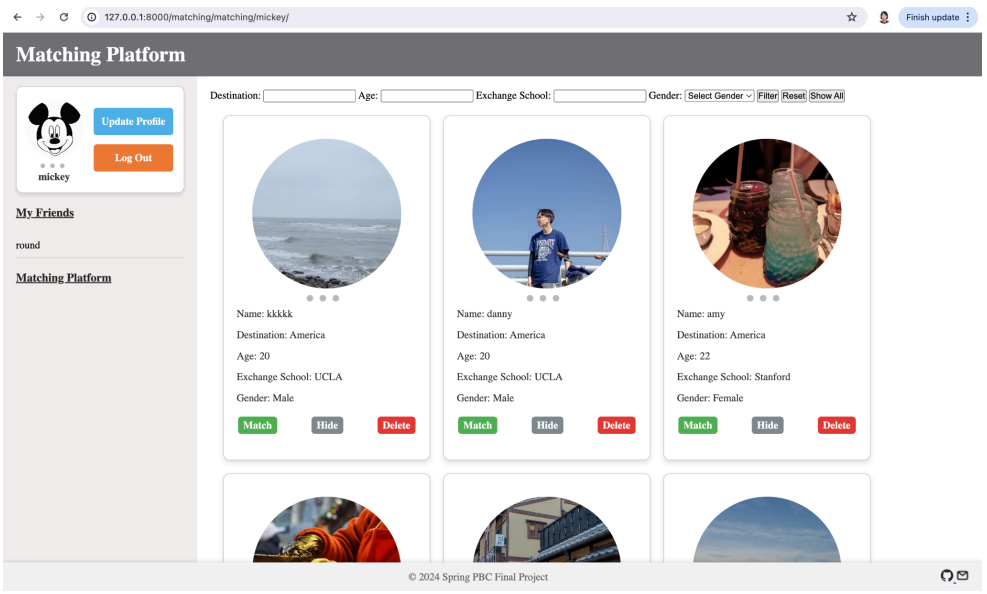
class DeleteProfile:
user (deleting user), deleted_user
class Meta

class MatchInvitation:
sender, receiver, mutual: Boolean field indicating invitation</td></tr>
</table>

		class Friend: user1 (friends1), user2 (friends2), create_at class Meta: unique_together

	views.py	**def user_login(request):**
		if request.method == 'POST':
		# Get username and password, authenticate
		if user is not None: # Redirect welcome page
		else:
		# “Password isn’t correct”, “Account does not exist”
		# Render login page
		else: # Render login page
		
		def register(request):
		# form = UserRegistrationForm
		if form is valid:
		# Save new user, new profile form
		# After register, login the account to welcome page
		
		class UserProfileUpdate:
		model = UserProfile
		form_class = UserProfileForm
		template = ‘login/register.html’
		def get_success_url: redirect after successful update
		def get_object: retrieve the specific UserProfile object to be edited

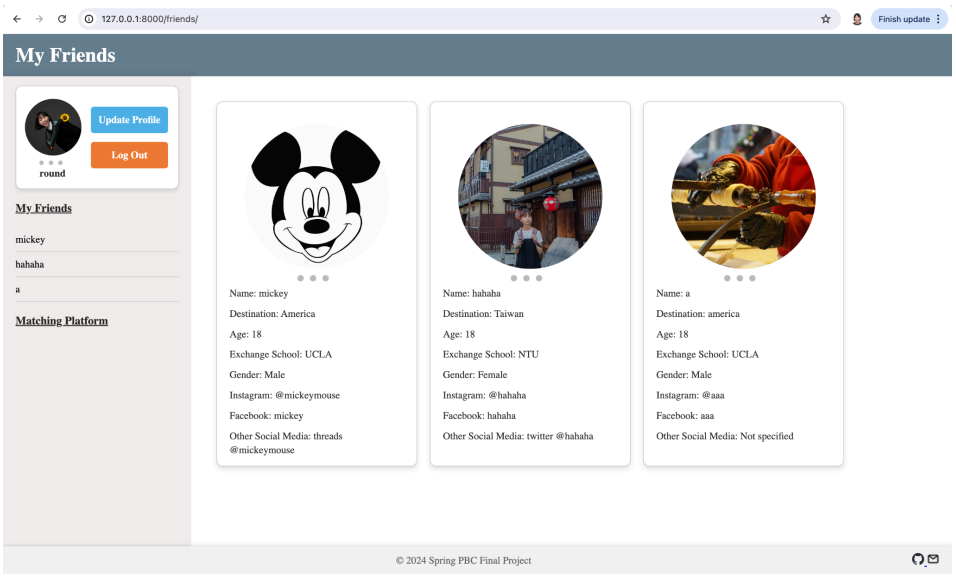
Welcome		
畫面		
功能簡介	負責顯示用戶的個人信息和資料，可以更新以及登出，且確保只有已登錄的用戶才能訪問自己的歡迎頁面。如果其他用戶嘗試訪問該頁面，系統將返回403錯誤。	
Pseudocode	**views.py**	**def welcome(request, username):** # retrieve the user object by username if request.user != user: render a 403 error page # Retrieve or create the user profile # Create a context dictionary with the username and profile # Render the welcome page with the context data

<table>
<tr><th colspan="3">Matching</th></tr>
<tr><td>畫面</td><td colspan="2"> 
</td></tr>
<tr><td>功能簡介</td><td colspan="2">交換生媒合網頁的核心功能之一，旨在幫助用戶找到適合的交換生匹配對象。
主要功能包含：
1. 隱藏個人資料（hide_profile）
2. 刪除個人資料（delete_profile）
3. 過濾匹配對象（filter_function 和 show_all_profiles）
4. 計算推薦分數（recommend_scores）：使用 pandas 和 sklearn 函式庫
5. 發送匹配請求（sent_match_request）
6. 顯示匹配頁面（user_matching）</td></tr>
<tr><td rowspan="3">Pseudocode</td><td>hide.py</td><td>def hide_profile(request, username):
Get the user profile to hide
Check if the profile has already been hidden by this user
if the profile has already been hidden: # increment the hide count
else:
# hidden for the first time, display a success message
# Redirect the user back to the matching platform</td></tr>
<tr><td>delete.py</td><td>def delete_profile(request, username):
Get the user profile to delete
Check if the user has already been deleted
if the user was already deleted: # Display “already deleted”
else: # successfully deleted, display a success message
Redirect the user back to the matching platform</td></tr>
<tr><td>filter.py</td><td>def filter_function(request):
if request.method == 'POST':
# Get filter parameters
# Clear filter parameters if reset button pressed
# Query filtered profiles excluding current user
# Render matching page
if it's a GET request, return matching page directly

def show_all_profiles(request):
# Get current user's profile
# Exclude profiles of current user and excluded users (friends)</td></tr>
</table>

		# Render matching page with profiles and current user's profile

	views.py	**def recommend_scores(current_user_profile):**
		# Get profiles from the database, preprocess the DataFrame
		# Handle missing values and ensure data types
		# Convert categorical variables into dummy/indicator variables
		# Calculate age distances between current user and other users
		# Calculate age distances and features for KNN
		# Calculate recommendation scores based on distance and age
		return scores
		
		def sent_match_request(request, username):
		# Retrieve User Objects
		if invitation exist:
		# Mark as mutual=True to indicate a two-way match
		# Create a new match request in the opposite direction
		# Create a Friend relationship between the two users
		else:
		# Create a new MatchInvitation object
		# Set mutual flag to False to indicate a one-way match request
		# Redirect the user back to the matching platform
		
		def user_matching(request, username):
		# Get the current user and ensure authentication
		# Get the user's friends
		# Fetch profiles not hidden by current user
		# Exclude deleted profiles and profiles with sent match requests
		# Calculate recommendation scores
		# Order profiles by recommendation scores and hide counts
		# Render matching page

Friends		
畫面		
功能簡介	顯示當前用戶的好友列表，顯示好友的社群軟體資訊。	
Pseudocode	**views.py**	**def friends_list(request):** # Get current user profile, get friend list # Render friend page with current user's profile, and friends

分工方式

B09602017 白宗民	整體架構建置；login, update, welcome, matching頁面及基本功能建置, error_handlers; Readme撰寫 → All include python, HTML, and CSS
B09401140 劉裕緯	完成delete, matching funciton & recommend score的演算法
B08612024 徐子涵	撰寫filter, show all的演算法；完成matching, friends的頁面配置；新增多張avatar功能
ntnu_40884231i 鄭若妤	嘗試avatar新增圖片, gender others button, 優化頁面(html,css), videos

心得感想

B09602017 白宗民	必須說我完全沒有想到我們的final project可以做的這如此完整，甚至本來想要單幹整個專案... 結果組員從第一次開會就展現了很好的配合及默契，主題一下就想好且達成共識。因為我本身比較熟悉程式開發，就先把基礎架構都弄好，後面的功能大家都有幫忙我覺得超讚，開會起來很有效率，能感受到組員的用心以及實作能力，每個人都盡到一份心力感覺。而且兩位女生都是初學者的情況我是真心佩服，可以一下子就搞懂Django, git, conda, HTML, CSS等的用法，是認真被嚇到... 裕緯的用心程度也是非常值得讚賞，偶爾會自己來問我問題，愛組員、愛PBC!!
B09401140 劉裕緯	首先這是我第一次參與一個完整的程式相關專案，前期花了很多時間去熟習，網站的架構，各種class該如何實作完成，到後期要解配對的實作時還有排列順序時，很謝謝組長及組員提供很多的建議以及相互配合，使得我的部分能順利完成。此外也很謝謝組員們完成拍攝與剪輯影片、書面報告等的工作，這些都是這個專案非常重要的一環，我也從中獲益斐淺。在這次專案中，不管是程式的實作，或是如何管理專案，這些收穫都是非常寶貴的，也非常開心能將自己學習的coding能力真實的實踐在一件事情上，也期許自己以後要更精進這項能力，完成更多的挑戰，最後也謝謝組長及組員這半個學期的照顧以及幫助！
B08612024 徐子涵	很高興能參與這次的專案製作。整個過程充滿了趣味與挑戰，能和組員一起討論題目，並延伸發想不同網頁功能是一段有趣的過程。不過在製作過程中也遇到了不少障礙，Github, Django, html, css這些東西對我來說都很新很陌生，光是理解Django的架構就花費了不少時間，寫程式的過程中也會反覆出現問題，需要不斷debug並解決。雖然過程充滿曲折，但與大家分工合作，最終完成了一個功能完整的網頁，真的收穫良多，很有成就感，感謝非常carry的組員們！最後感謝商管程這堂課帶我認識程式設計，對我很有啟發性，希望未來能繼續精進，應用所學。
ntnu_40884231i 鄭若妤	很高興能參與這次的專案製作，雖然對於我這個超純文組腦袋和程式完全新手來說非常具有挑戰性，甚至是一開始還沒開始撰寫程式，只是在理解如何共同編輯程式碼、Github、環境等操作就遇到了一些阻礙。在真正撰寫程式時也遇到時間很多但看著電腦怎麼想也想不出來，或是花了很多時間寫但網站卻跑不動的問題。很感謝組員們都很包容我，也沒有任何責罵，而是很有耐心地講解並且把適合我的工作內容分配給我。感謝商管程式設計這門課讓我認識了程式設計，對我來說彷彿開啟了新的世界，對於工作上原本看不懂的程式碼，現在也能稍微理解一點，真的是我大學最具啟發性、感受到進步、收穫也最多的一堂課。希望未來能繼續精進Python或學習其他語言，有一天也能像我回頭看作業一一樣，覺得這個專題的內容很簡單！(因為現在我覺得很難)謝謝老師、謝謝組員。