

專案名稱: Anime Helper

成員:

謝侑哲(112550069)、張程翔(112550205)、廖漢軒(112550087)、
游翔宇(112550103)、徐柏安(112550158)

1. Introduction

製作一個現代化、高互動性的動畫資料整合網站，具有分類、檢索、評價、登入、收藏清單等功能，讓使用者能夠更方便找到感興趣的動畫。

每部動畫都有個主頁以及不同的tag，tag可以分類讓用戶更快得找出自己喜歡的動畫，主頁中顯示的內容是簡介這部動畫的內容。

評分系統可以讓用戶給出自己的評價，令其他人有參考依據，在做選擇的時候更加容易。並且搭配收藏清單的功能，可以令會員分類自己的喜好使得用戶對網站的黏著度上升。

留言系統讓用戶可以互相交流，並提出自己的見解與他人互相交流。網站會做出趨勢排名，讓用戶在選擇的時候，有更多數據可以參考。

2. Data

2.1 Introduction to data

我們所選的資料庫裡面，包含了2023以前(含)的動漫資訊以及網路上的評價。像是：上映時間、使用者評分、使用者性別等等資料。

2.2 Where is the data from

本資料庫的資料來源為歐美知名動漫資訊網站，MyAnimeList，由資料庫作者使用爬蟲技術爬取。

2.3 What do the column and tables mean

我們選用的資料集內，總共有六個table，以下將以括號說明。

Tables	Columns	Primary Key
anime-dataset-2023 (動漫詳細資料)	anime_id, Name (羅馬拼音名稱), English Name, Other name (日文名稱), Score, Genres, Synopsis (簡介), Type (TV/Movie...), Episodes, Aired (上映日期), Premiered (上映年份和季節), Status (現正上映/已完結...), Producers, Licensors, Studios, Source (原作小說/漫畫...), Duration, Rating(年齡限制), Rank, Popularity, Favorites, Scored By(評分人數), Members (加到清單中的人數), Image URL	anime_id

anime-filtered (動漫的分類、特徵)	anime_id, Name (羅馬拼音名稱), Score, Genres, English Name, Japanese Name, synopsis (簡介), Type (TV/Moive...), Episodes, Aired (上映日期), Premiered (上映年份和季節), Producers, Licensors, Studios, Source (原作為小說/漫畫...), Duration, Rating(年齡限制), Ranked, Popularity, Favorites, Scored By(評分人數), Members (加到清單中的人數), Favorites (設為喜歡的人數), Watching (正在觀看的人數), Completed (已全部看完的人數), On-Hold (加到待觀看的人數), Dropped (棄坑的人數)	anime_id
final_animatedataset (使用者的評價)	username, anime_id, my_score (使用者給的評分), user_id, gender, title, type (TV/Movie...), Source (原作為小說/漫畫...), score (平均分數), scored_by (評分人數), rank, popularity, genre	anime_id
user-filtered (使用者的評價)	user_id, anime_id, rating (使用者評分)	user_id, anime_id
user-details-2023 (使用者詳細資料)	Mal ID(使用者的Unique ID), Username, Gender, Birthday, Location, Joined (加入日期), Mean Score(使用者的平均評分), Days Watched (總觀看天數), Watching (正在看的動漫數量), Completed (完整看完的動漫數量), On_Hold (設為待觀看的動漫數量), Dropped (已棄坑的動漫數量), Plan to Watch (預計要看的動漫數量), Total Entries (清單中的動漫總數), Rewatched (重看過的動漫數量), Episodes Watched (看過了幾集)	Mal ID
user-score-2023 (使用者的評價)	user_id, Username, anime_id, Anime Title, rating (使用者評分)	user_id, anime_id

除了這些原本就有的tables，我們預計還會新增別的tables以符合我們想達成的功能:

Favorites-List (收藏清單)	user_id, Username, anime_id, Anime Title, favorited date(收藏的時間，用來排序清單)	user_id, anime_id
Our-User (使用者登入資訊)	user_id, Username, password	user_id
Comment (留言)	anime_id, user_id, comment_id, content	comment_id

2.4 When will data be updated

像是註冊帳號或是修改個人資訊時，紀錄使用者登入資訊的table就會被更新，使用者新增動漫到收藏清單時也會更新table的內容。使用者對動漫評價、評分時也會更改table內容(動漫的平均分數、使用者評論總數等等)。

2.5 Source of data

<https://www.kaggle.com/datasets/dbdmobile/myanimelist-dataset?select=user-filtered.csv>

3. Application Design

3.1 Main idea

因為MyAnimeList的網站年代久遠，很多的設計都比較過時，並且網站的資訊很雜亂。因此我們的網站的核心想法就是取得MyAnimeList的資料，並重新建構一個更現代化的網站，用來呈現動畫資訊、各類品項、動畫排名等。

除了直接取用資料以外，我們也希望打造一個互動性強的網站，因此我們會設計網站的登入系統，並且讓每個使用者能擁有自己的收藏清單等個人功能。另外也將設計留言、評分等功能，讓網站使用者能互動形成獨立社群。

在互動的過程中，使用者留下的資料也會被我們紀錄下來，用來做後續的趨勢分析、推薦系統等等。

3.2 Target audience and application scenario

這個網站的主要受眾是喜愛動畫的觀眾，在我們看動畫的過程中，常常會陷入要看哪一部動畫的選擇，這時使用者就可以到這個網站，查看我們的推薦系統，又或是選擇想要觀看的品類，從中挑選評分高、描述有興趣的作品，也可以看到其他人的留言評鑑等等，來決定是否要觀看這部動畫。

另外，收藏功能也可以很好的幫助使用者紀錄找到的資訊，讓這個網站能更方便使用。

3.3 Functionality

我們網站設計的功能有很多，這邊主要依照我們開發的優先順序排列，不一定會全部開發完，但會依據優先序一個個去盡量開發：

(1) 分類 & 排名：

在網頁中，可以透過動畫的分類，去找出每個品類中有哪些動畫，並依據評分、熱門程度等等來做排序，或是直接列出所有動畫的評分排名等。讓使用者可以知道大家都推薦什麼好動畫。

(2) 查詢 & Filter

在網頁中，可以透過關鍵字搜尋等功能找到某個動畫，進而點進去動畫的主頁。另外，我們也會在查詢中，讓使用者可以設定Filter，篩選出想要什麼樣的動畫，包括了評分在某個評級以上，又或者是集數在某個數字以下。讓使用者可以根據個人需求找到對應的動畫。

(3) 登入系統:

上述提到的收藏、留言等功能，基本上都是綁定登入系統，因此我們會優先做登入系統，可能會使用google登入的API，或者是使用者可以註冊網站帳號密碼的方式實作。

(4) 收藏清單:

在網頁中，使用者在登入之後，可以在創建附屬在個人帳號底下的收藏清單，使用者可以創建多個不同title的收藏清單，並且這個收藏清單裡面可以新增、刪除動畫，讓使用者可以做個人化的動畫紀錄。

(5) 動畫主頁:

在網頁中，使用者搜尋到的動畫，或者是收藏清單裡面儲存的動畫，都將可以連結到一個動畫的主頁，這個主頁會有動畫的描述資訊、分類tag、播出時間、評分等等。讓使用者可以快速了解動畫資訊。

(6) 評分系統:

我們的資料庫中，原有的資料大多來自歐美的使用者，因此我們會在動畫的主頁底下增加評分系統，讓該網站的使用者可以對網站做評分，這樣得出來的分數會更接近網站使用者而非資料庫的固有資料。

(7) 留言系統:

在動畫主頁底下，我們會建立留言系統，讓網站的使用這可以在底下留言，寫出對於動畫的評論或是心得等等，促進網站社群的討論風氣，也給其他使用者分數以外的參考點。

(8) 首頁推薦:

在網頁的首頁，我們會隨機推薦各個品類中，評分高的動畫，讓使用者不一定要有特別的目的地性去找動畫，可以根據我們的隨機推薦來看看有沒有有趣的動畫。

(9) 趨勢分析:

在網頁中，我們可以設立一個介面，根據資料庫的資料去做一些趨勢分析，例如隨著時間的過去，某個品類的動畫是更流行還是更衰頹，又或者是整體動畫數量的增長幅度等等。

(10) LLM助手:

在網頁中，我們可以引入Open AI API等等，透過RAG的方式，讓LLM可以根據網站的資料，去跟使用者互動或是推薦一些動畫等等。

3.4 CRUD analysis

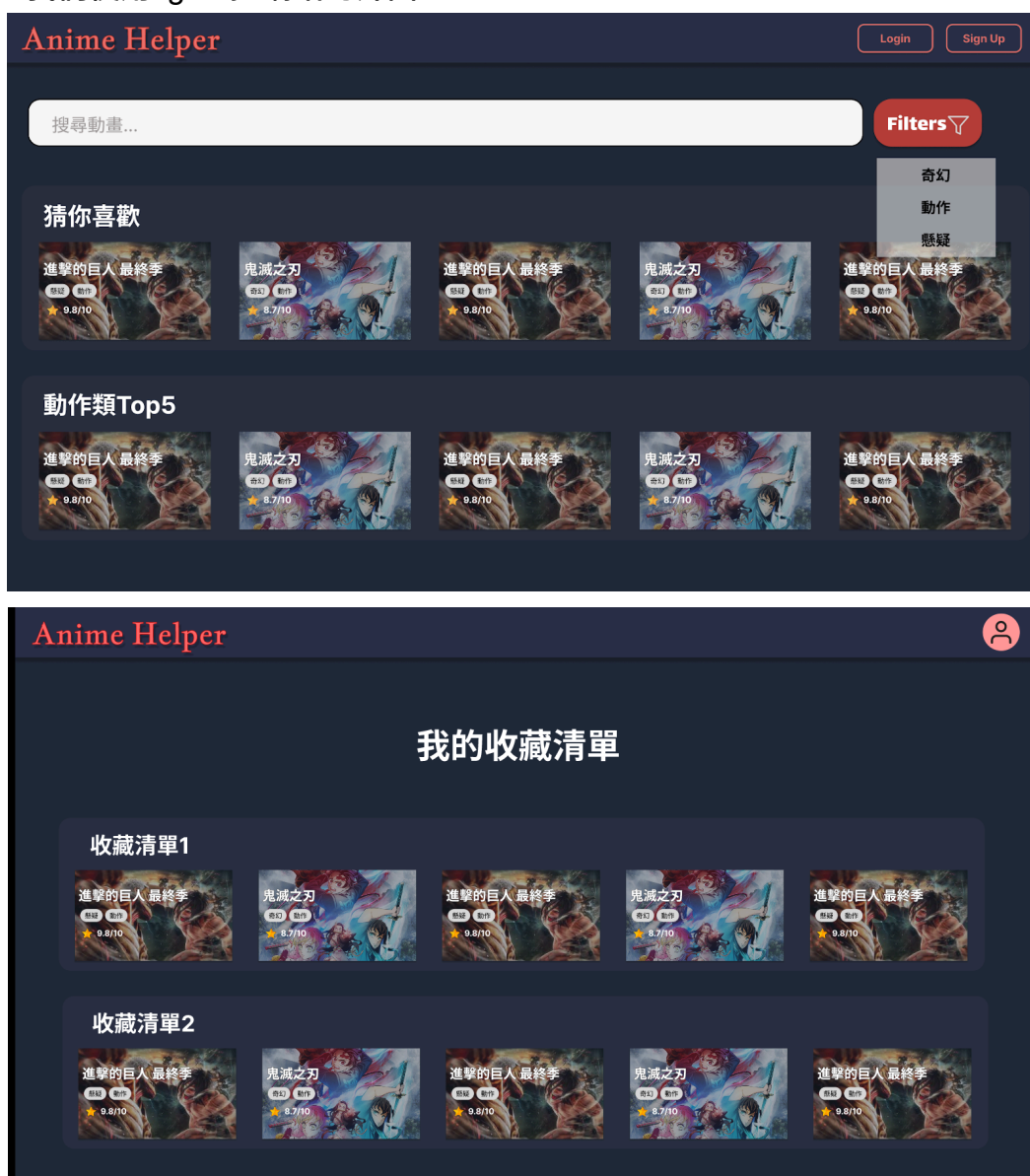
在3.3的部分講到了網站會有的所有功能跟使用者可以有的互動。這個section主要分析那些功能是否符合CRUD:

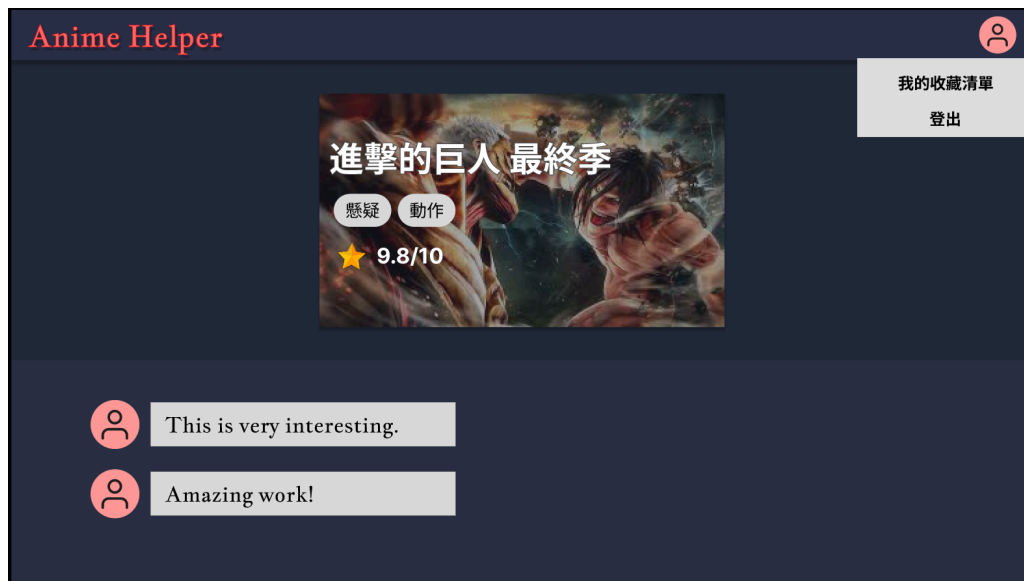
- (1) 登入系統: 可以透過註冊新增使用者資料，並透過登入看到新增的資料並修改，符合CRUD的CRU。
- (2) 收藏清單: 可以增刪清單、增刪動畫等收藏清單的資料，並且能很好的視覺化收藏清單資料，符合CRUD的CRUD。
- (3) 評分系統: 可以新增評分資料，符合CRUD的CRU。
- (4) 留言系統: 可以新增留言資料，符合CRUD的CRU。

其他未提到的皆符合CRUD的R。

3.5 Interface

我們使用figma拉網站的介面





3.6 Exception handling

對於例外處理的部分，我們主要會在後端做request的資料檢查，當資料不符合所要求的型別、格式、範圍時，會回傳錯誤訊息，由前端做錯誤訊息的呈現

4. Work Plan

4.1 Time schedule:

學習:

前端: (1) 基本前端三要件 (2) React.js

後端: (1) 基本Node.js (2) Express.js

前端:

- (1) 首頁
- (2) 分類介面
- (3) 查詢功能
- (4) 登入頁面
- (5) 收藏清單頁面
- (6) 動畫主頁(資訊呈現)
- (7) 動畫主頁(評分系統)
- (8) 動畫主頁(留言系統)
- (9) 首頁推薦區塊
- (10) 趨勢分析頁面
- (11) LLM助手區塊

後端:

- (1) 整體後端框架
- (2) 種類API (回傳種類、種類對應動畫)

- (3) 查詢API (回傳關鍵字對應動畫)
- (4) 登入API (建立帳號、回傳帳號資訊、帳號修改)
- (5) 收藏清單API (建立清單、回傳清單資訊、清單修改)
- (6) 動畫詳細資訊API
- (7) 評分API (接收評分)
- (8) 留言API (接收、回傳留言資訊)
- (9) 推薦API (推薦演算法+回傳資訊)
- (10) 趨勢資料API
- (11) RAG + LLM
- (12) LLM助手 API

4.2 Discussion

我們使用Notion作為主要的討論紀錄及專案管理頁面

<https://marsh-venus-f8c.notion.site/12b9a1ba8a1180388573f969972b4237>

4.3 Repo

<https://github.com/NYCU-DBMS/DBMS-Final-Project>

4.4 Roles

前端: 廖漢軒、游翔宇、徐柏安

後端: 謝侑哲、張程翔