# HAND Movies

Koko Imafuku, Park Kyunam, Seo Gaeun

# Table of Contents

# 1. Introduction

# Introduction

## Users?

Anyone trying to find a good movie to watch on Hulu, Amazon Prime, Netflix, and Disney+

People with limited Internet use

## Main Goals?

Provide a convenient application that allows users to search films on the 4 OTT platforms with multiple filters




# Main functions

- Filter function: Select filters like **rating**, **year of release,** or **OTT platform** of the movie

- Keyword function: Type in the **title of the movie, director's name, actor's name,** or the **genre of the movie**

- Delete function: Click the delete button on the details page to delete a film on the database

# Overall flow of usage of application: Search


List of recommended movies

Select filtering
- Rating
- Year of Release
- OTT

Type in the Title, Genre, Director’s name, Actor’s name

Detail page, click on “Details"

# Overall flow of usage of application: recommendation


List of recommended movies

Select filtering
- Rating
- Year of Release
- OTT

Filtered recommendation

Detail page, click on “Details"

# Languages, packages, and frameworks

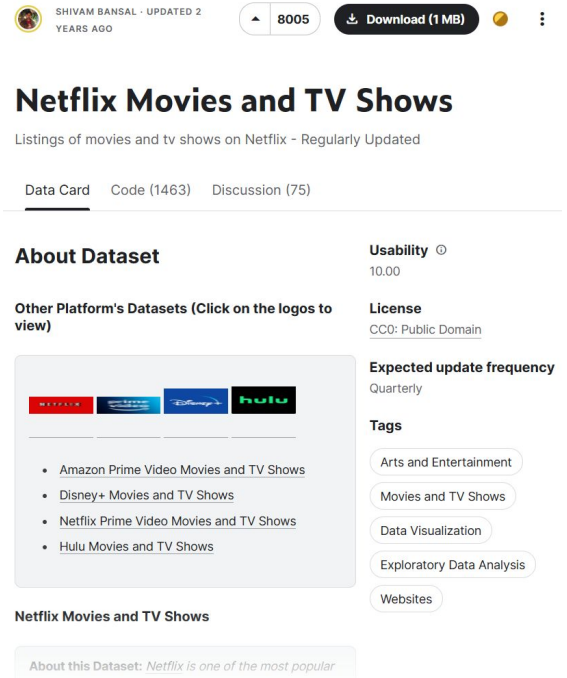
- Python 3
- Tkinter Library for GUI
- Pandas for data cleaning


# 2. DB Modeling and Design

# Raw Dataset

- Collected data from **Kaggle**.
- File format : csv

| Data | row | column |
|---|---|---|
| Netflix | 8807 | 12 |
| Disney+ | 1450 | 12 |
| Amazon Prime | 9668 | 12 |
| Hulu | 3073 | 12 |



# Raw Dataset

| show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| s1 | Movie | Dick Johnson Is Dead | Kirsten Johnson | NaN | United States | September 25, 2021 | 2020 | PG-13 | 90 min | Documentaries | As her father nears the end of his life, filmm... |
| s2 | TV Show | Blood & Water | NaN | Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban... | South Africa | September 24, 2021 | 2021 | TV-MA | 2 Seasons | International TV Shows, TV Dramas, TV Mysteries | After crossing paths at a party, a Cape Town t... |
| s3 | TV Show | Ganglands | Julien Leclercq | Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi... | NaN | September 24, 2021 | 2021 | TV-MA | 1 Season | Crime TV Shows, International TV Shows, TV Act... | To protect his family from a powerful drug lor... |
| s4 | TV Show | Jailbirds New Orleans | NaN | NaN | NaN | September 24, 2021 | 2021 | TV-MA | 1 Season | Docuseries, Reality TV | Feuds, flirtations and toilet talk go down amo... |
| s5 | TV Show | Kota Factory | NaN | Mayur More, Jitendra Kumar, Ranjan Raj, Alam K... | India | September 24, 2021 | 2021 | TV-MA | 2 Seasons | International TV Shows, Romantic TV Shows, TV ... | In a city of coaching centers known to train I... |

# Raw Dataset

| Column name | example |
|---|---|
| show_id | s1 |
| type | Movie |
| title | Dick Johnson is Dead |
| director | Kirsten Johnson |
| date_added | September 25, 2021 |
| release_year | 2020 |
| rating | PG-13 |

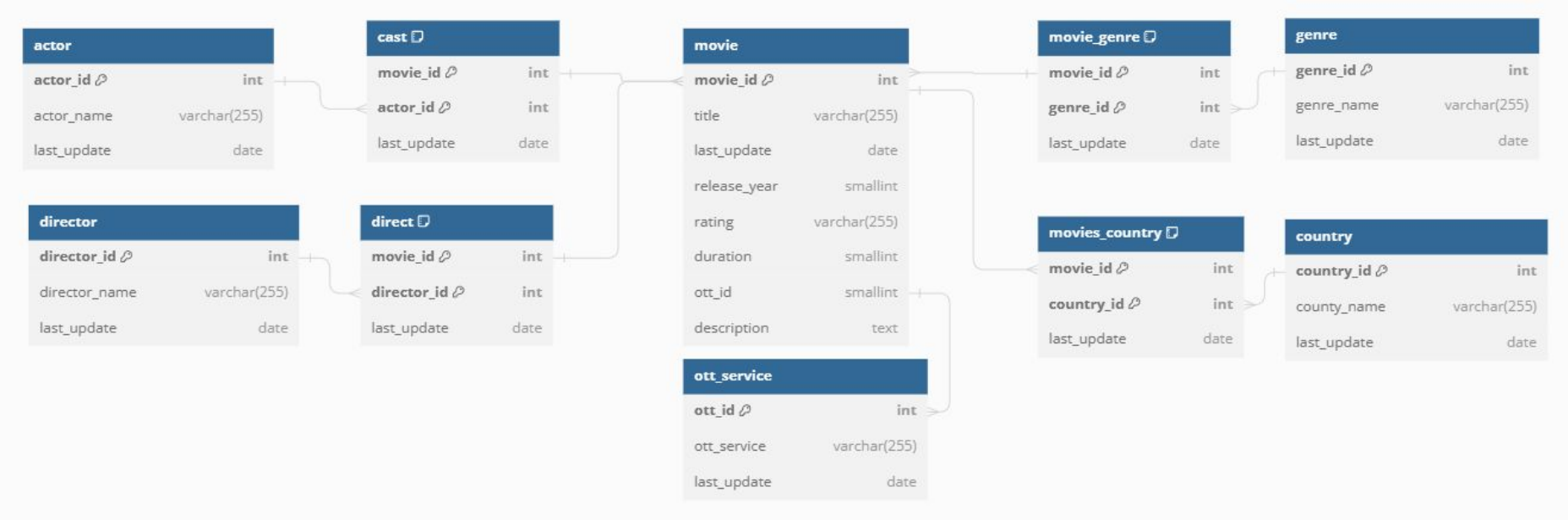
# Raw Dataset

| Column name | example |
| --- | --- |
| cast | Sami Bouajila, Tracy Gotoas, … |
| country | United States, Ghana, United Kingdom, … |
| listed_in | Dramas, Independent Movies, … |
| duration | 125 min |
| description | On a photo shoot in Ghana, an American,,, |

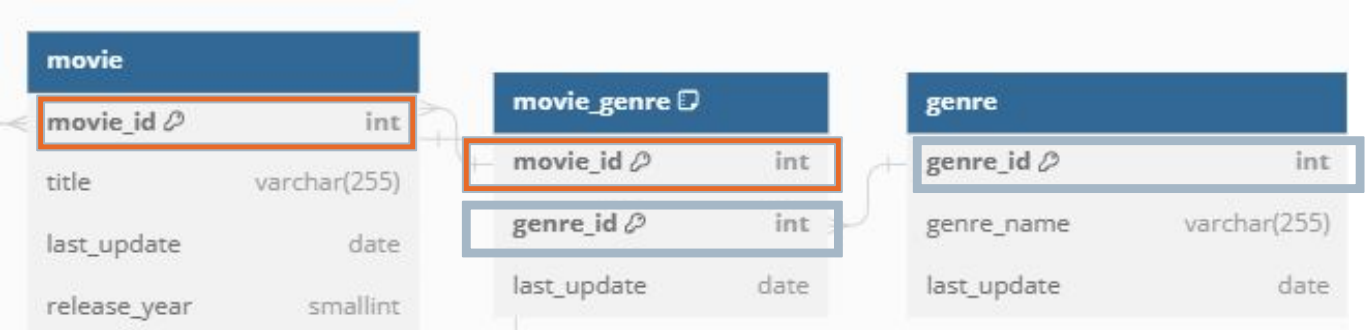
# Data Preprocessing

- **Concat** 4 datasets (netflix, disney, amazon, hulu)
- Extract each string data from **list** data
- Fill missing data_added data ‘1970-01-01’
- Fill missing release_year data ‘0000’
- Fill missing duration data ‘000’
- Fill missing other values to ‘no info’
- Change date format to use it in Mysql datetime format
  - [ex] September 25, 2021 -> 20210925
- Extract **int** data from string data
  - [ex] 90 min -> 90
- Unify **rating** information to [R, PG-13, PG, G, NC-17]
  - TV-MA, 16+ => R

# ER diagram


actor
actor_id int
actor_name varchar(255)
last_update date
cast
movie_id int
actor_id int
last_update date
movie
movie_id int
title varchar(255)
last_update date
release_year smallint
rating varchar(255)
duration smallint
ott_id smallint
description text
movie_genre
movie_id int
genre_id int
last_update date
genre
genre_id int
genre_name varchar(255)
last_update date
director
director_id int
director_name varchar(255)
last_update date
direct
movie_id int
director_id int
last_update date
movies_country
movie_id int
country_id int
last_update date
country
country_id int
county_name varchar(255)
last_update date
ott_service
ott_id int
ott_service varchar(255)
last_update date

# DB Design




| movie_id int | genre_id int | last_update date |
|---|---|---|
| 1 | 25 | 2021-09-25 |
| 2 | 15 | 2021-09-24 |
| 3 | 26 | 2021-09-24 |
| 3 | 36 | 2021-09-24 |
| 3 | 37 | 2021-09-24 |
| 4 | 17 | 2021-09-24 |

| genre_id int | genre_name varchar(255) | last_update date |
|---|---|---|
| 1 | & culture | 2021-11-26 |
| 2 | action | 2021-11-26 |
| 3 | action & adventure | 2021-11-26 |
| 4 | adult animation | 2021-11-26 |
| 5 | adventure | 2021-11-26 |
| 6 | animals & nature | 2021-11-26 |

# Table Shape

- **movies**: 8 columns * 16480 rows
- **ott_service**: 3 columns * 4 rows
- **cast**: 3 columns * 89652 rows
- **actor**: 3 columns * 48545 rows
- **direct**: 3 columns * 17997 rows
- **director**: 3 columns * 10672 rows
- **genre**: 3 columns * 73 rows
- **movies_genre**: 3 columns * 34071 rows
- **country**: 3 columns * 124 rows
- **movies_country**: 3 columns * 18825 rows

# movies Table

- **Shape**: 8 columns * 16480 rows
- **Columns**: movie_id, title, last_update, release_year, rating, duration, ott_id, description
- **movie_id**: primary key, distinct number to be distinguished (1 ~ 16481)
- **title**: title of movies
- **last_update**: last update date of movies data (1970.01.01, 2008.01.01 ~ 2021.11.26)
- **release** year: release year of movie (0, 1920 ~ 2021)
- **rating**: Motion Picture Association film rating system [no info, R, PG-13, PG, G, NC-17]
- **duration**: running time of the movie (0, 1 ~ 601)
- **ott_id**: id of ott_service (1 ~ 4, 1: Amazon Prime, 2: Disney, 3: Hulu, 4: Netflix)
- **description**: detailed description of movie

# ott_service Table

- **Shape**: 3 columns * 4 rows
- **Columns**: ott_id, ott_service, last_update
- **ott_id**: primary key, distinct number to be distinguished (1 ~ 4)
- **ott_service**: name of ott service
- **last_update**: last update date of ott_service data

# actor/cast Table

- **Shape:** 3 columns * 48545 rows
- **Columns**: actor_id, actor_name, last_update
- **actor_id**: primary key, distinct number to be distinguished (1 ~ 48545)
- **actor_name**: name of actor
- **last_update**: last update date of actor data

- **Shape**: 3 columns * 89652 rows
- **Columns**: movie_id, actor_id, last_update
- **movie_id**: primary key, foreign key refers to movies.movie_id
- **actor_id**: primary key, foreign key refers to actor.actor_id
- **last_update**: last update date of cast data

# director/direct Table

- **Shape:** 3 columns * 10672 rows
- **Columns**: director_id, director_name, last_update
- **director_id**: primary key, distinct number to be distinguished (1 ~ 10672)
- **director_name**: name of director
- **last_update**: last update date of director data

- **Shape**: 3 columns * 17997 rows
- **Columns**: movie_id, director_id, last_update
- **movie_id**: primary key, foreign key refers to movies.movie_id
- **director_id**: primary key, foreign key refers to director.director_id
- **last_update**: last update date of direct data

# genre/movies_genre Table

- **Shape:** 3 columns * 73 rows
- **Columns**: genre_id, genre_name, last_update
- **genre_id**: primary key, distinct number to be distinguished (1 ~ 73)
- **genre_name**: name of genre
- **last_update**: last update date of genre data

- **Shape:** 3 columns * 34071 rows
- **Columns**: movie_id, genre_id, last_update
- **movie_id**: primary key, foreign key refers to movies.movie_id
- **genre_id**: primary key, foreign key refers to genre.genre_id
- **last_update**: last update date of movies_genre data

# country/movies_country Table

- **Shape:** 3 columns * 124 rows
- **Columns**: country_id, country_name, last_update
- **country_id**: primary key, distinct number to be distinguished (1 ~ 124)
- **country_name**: name of country
- **last_update**: last update date of country data

- **Shape:** 3 columns * 18825 rows
- **Columns**: movie_id, country_id, last_update
- **movie_id**: primary key, foreign key refers to movies.movie_id
- **country_id**: primary key, foreign key refers to country.country_id
- **last_update**: last update date of movies_country data

# Main SQL Queries: search

```
SELECT DISTINCT m.movie_id, m.title,
GROUP_CONCAT(DISTINCT director.director_name SEPARATOR ', ') AS directors,
GROUP_CONCAT(DISTINCT actor.actor_name SEPARATOR ', ') AS actors,
GROUP_CONCAT(DISTINCT genre.genre_name SEPARATOR ', ') AS genres,
m.release_year, m.rating, m.duration, ott_service
FROM movies m
LEFT JOIN direct ON m.movie_id = direct.movie_id
LEFT JOIN director ON direct.director_id = director.director_id
LEFT JOIN cast ON m.movie_id = cast.movie_id
LEFT JOIN actor ON cast.actor_id = actor.actor_id
LEFT JOIN movies_genre ON m.movie_id = movies_genre.movie_id
LEFT JOIN genre ON movies_genre.genre_id = genre.genre_id
LEFT JOIN ott_service ON m.ott_id = ott_service.ott_id
WHERE {condition}
GROUP BY m.movie_id
LIMIT {limit}"
```

# Main SQL Queries: recommendation

```
SELECT DISTINCT m.movie_id, m.title,
GROUP_CONCAT(DISTINCT director.director_name SEPARATOR ', ') AS directors,
GROUP_CONCAT(DISTINCT actor.actor_name SEPARATOR ', ') AS actors,
GROUP_CONCAT(DISTINCT genre.genre_name SEPARATOR ', ') AS genres,
m.release_year, m.rating, m.duration, ott_service
FROM movies m
LEFT JOIN direct ON m.movie_id = direct.movie_id
LEFT JOIN director ON direct.director_id = director.director_id
LEFT JOIN cast ON m.movie_id = cast.movie_id
LEFT JOIN actor ON cast.actor_id = actor.actor_id
LEFT JOIN movies_genre ON m.movie_id = movies_genre.movie_id
LEFT JOIN genre ON movies_genre.genre_id = genre.genre_id
LEFT JOIN ott_service ON m.ott_id = ott_service.ott_id
WHERE {where_clause}
GROUP BY m.movie_id # for GROUP_CONCAT
ORDER BY RAND()
LIMIT {limit}
```

# Main SQL Queries: detail

```
SELECT DISTINCT m.movie_id, m.title,
GROUP_CONCAT(DISTINCT director.director_name SEPARATOR ', ') AS directors,
GROUP_CONCAT(DISTINCT actor.actor_name SEPARATOR ', ') AS actors,
GROUP_CONCAT(DISTINCT genre.genre_name SEPARATOR ', ') AS genres,
m.release_year, m.rating, m.duration, ott_service, m.last_update, m.description
FROM movies m
LEFT JOIN direct ON m.movie_id = direct.movie_id
LEFT JOIN director ON direct.director_id = director.director_id
LEFT JOIN cast ON m.movie_id = cast.movie_id
LEFT JOIN actor ON cast.actor_id = actor.actor_id
LEFT JOIN movies_genre ON m.movie_id = movies_genre.movie_id
LEFT JOIN genre ON movies_genre.genre_id = genre.genre_id
LEFT JOIN ott_service ON m.ott_id = ott_service.ott_id"
WHERE m.movie_id = {movie_id}
GROUP BY m.movie_id'
```

# Main SQL Queries: delete

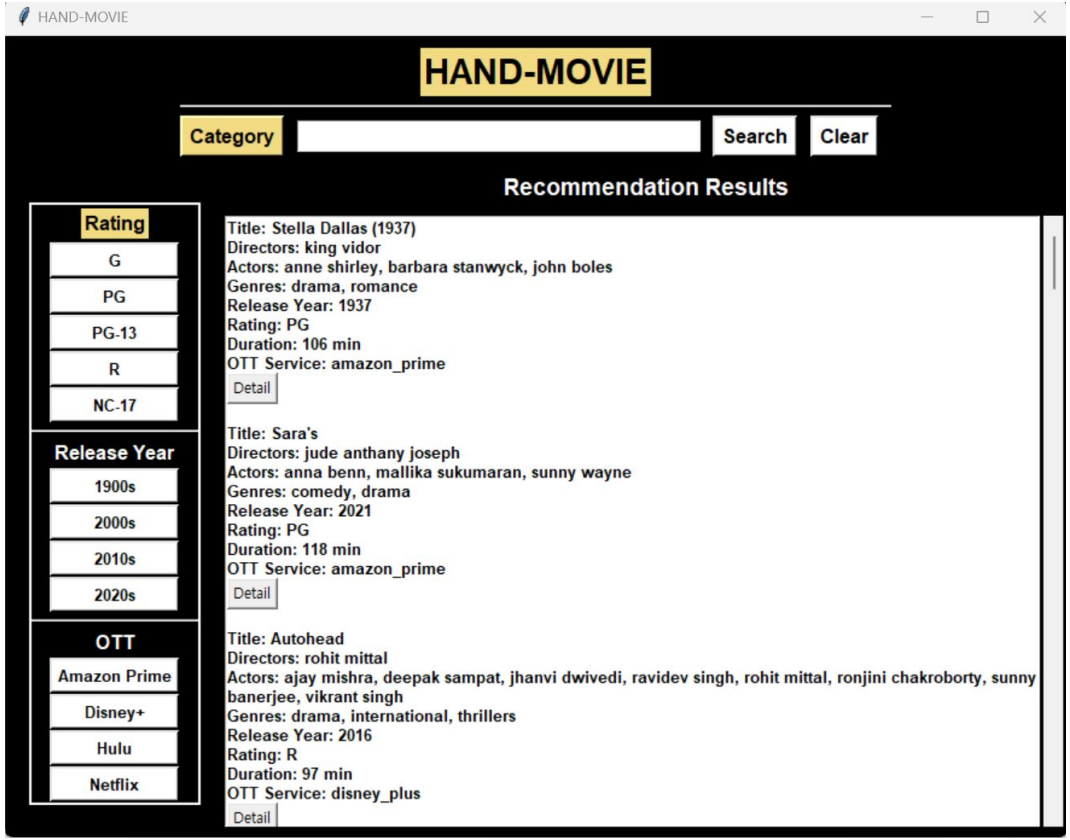
DELETE FROM {'movies', 'direct', 'cast', 'movies_genre', 'movies_country'} WHERE movie_id = {movie_id}

# 3. GUI Demo

# GUI

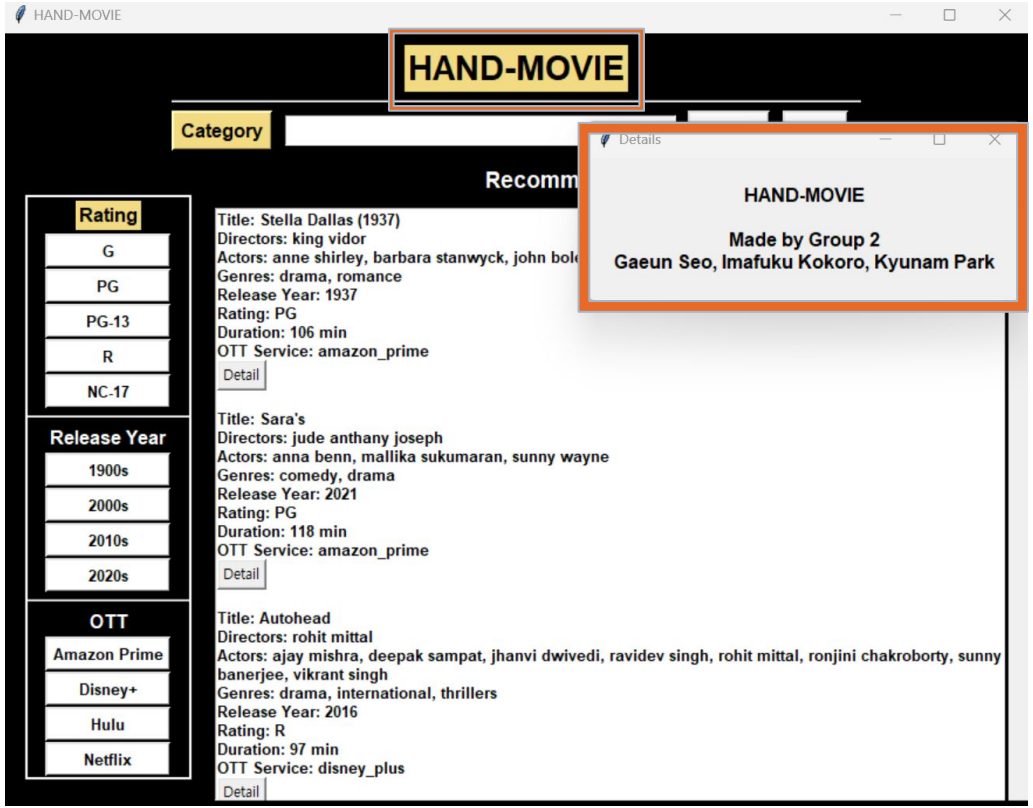
Recommendation
When you turn on the program, there will be recommendation.

You can get another recommendation by pressing Clear button.



# GUI

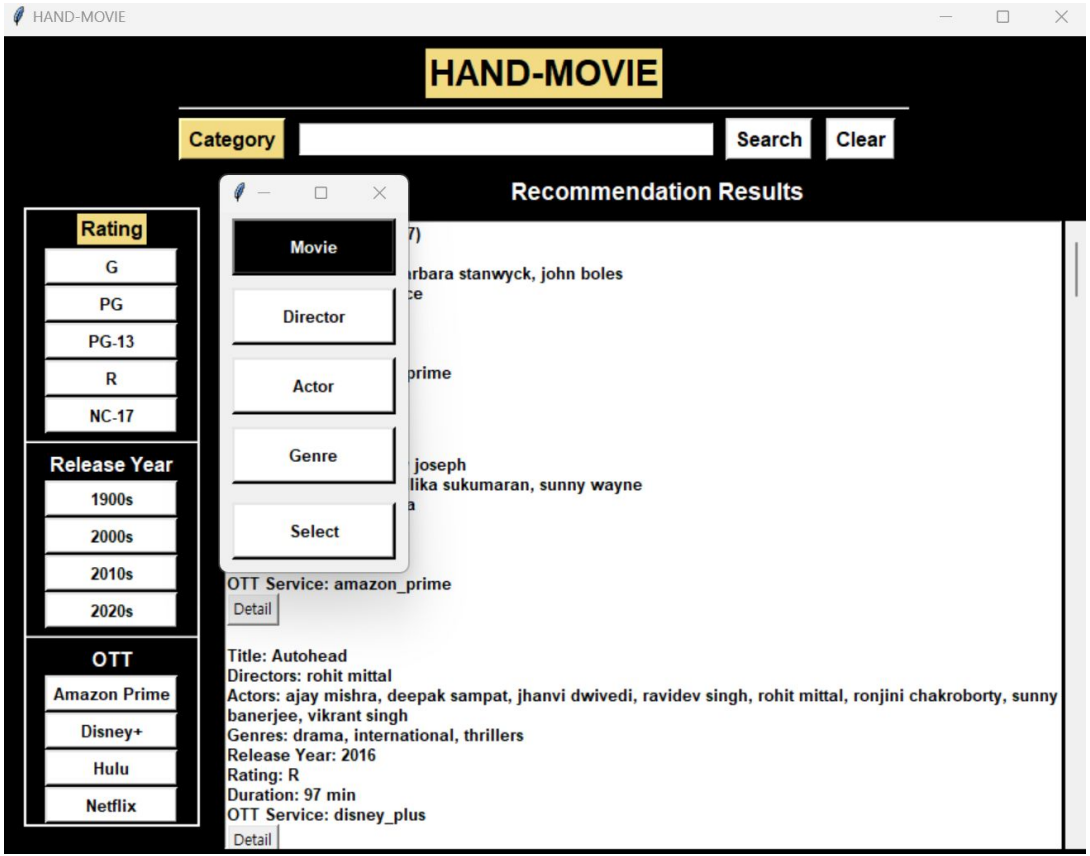
Click title HAND-MOVIE
- You can see ‘made by’ info



# GUI

Click Category

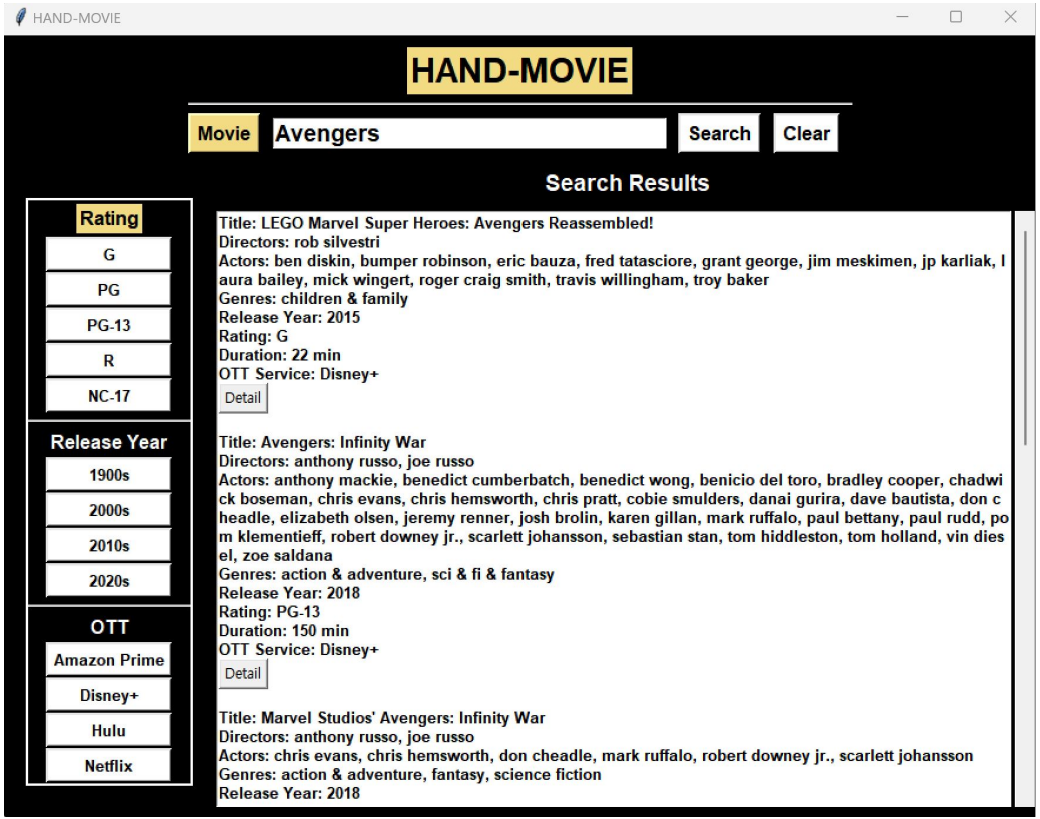
- Can choose the way to search
  - Movie
  - Director
  - Actor
  - Genre



# GUI

Click Search

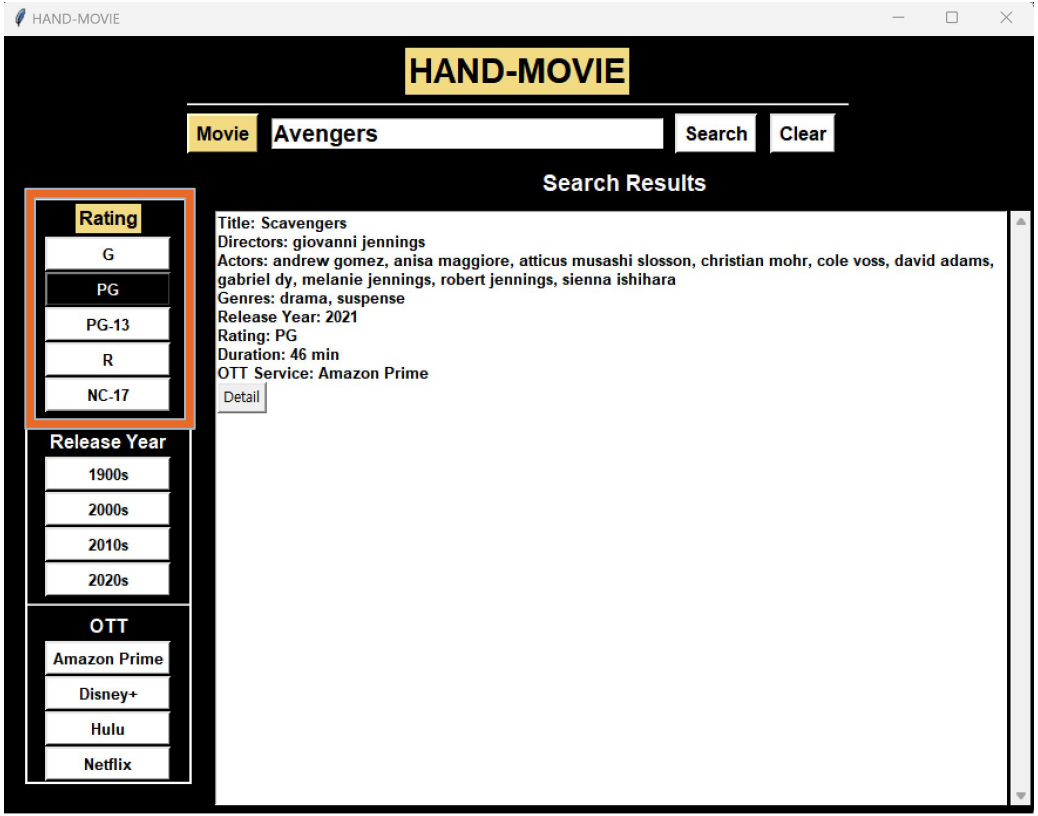
- Can get the information of the movie



# GUI

Filtering

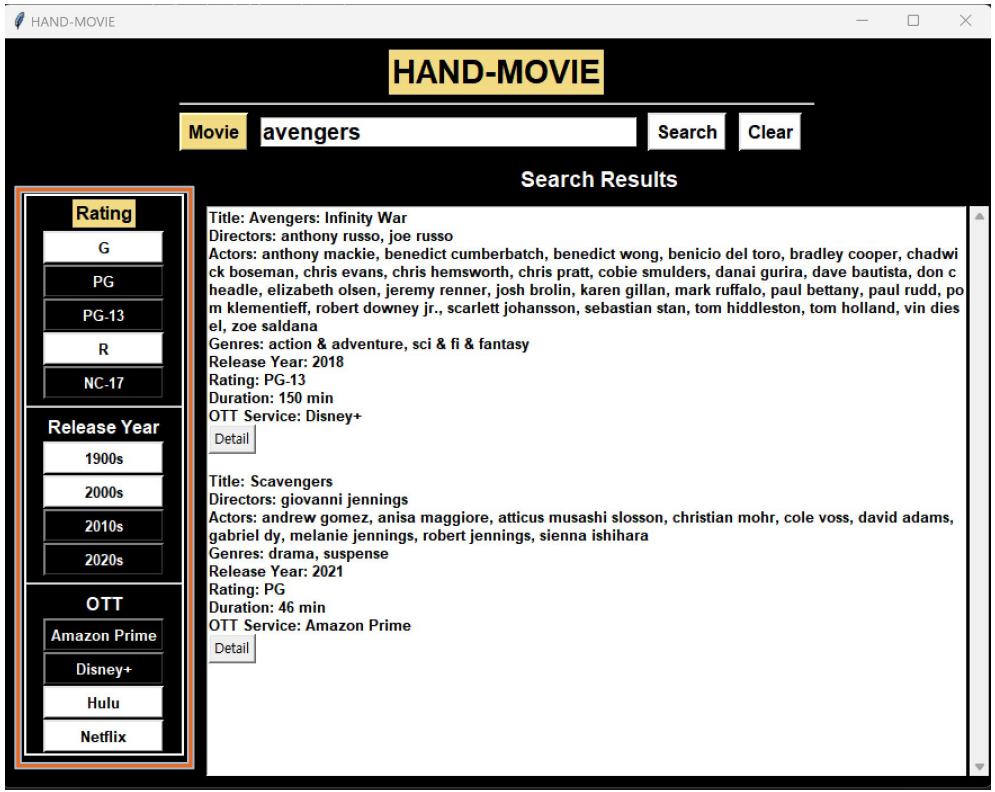
- Click ‘PG’ and click ‘search’ again, you can get the filtered movie list.



# GUI

Filtering

- You can also filter through multiple element.



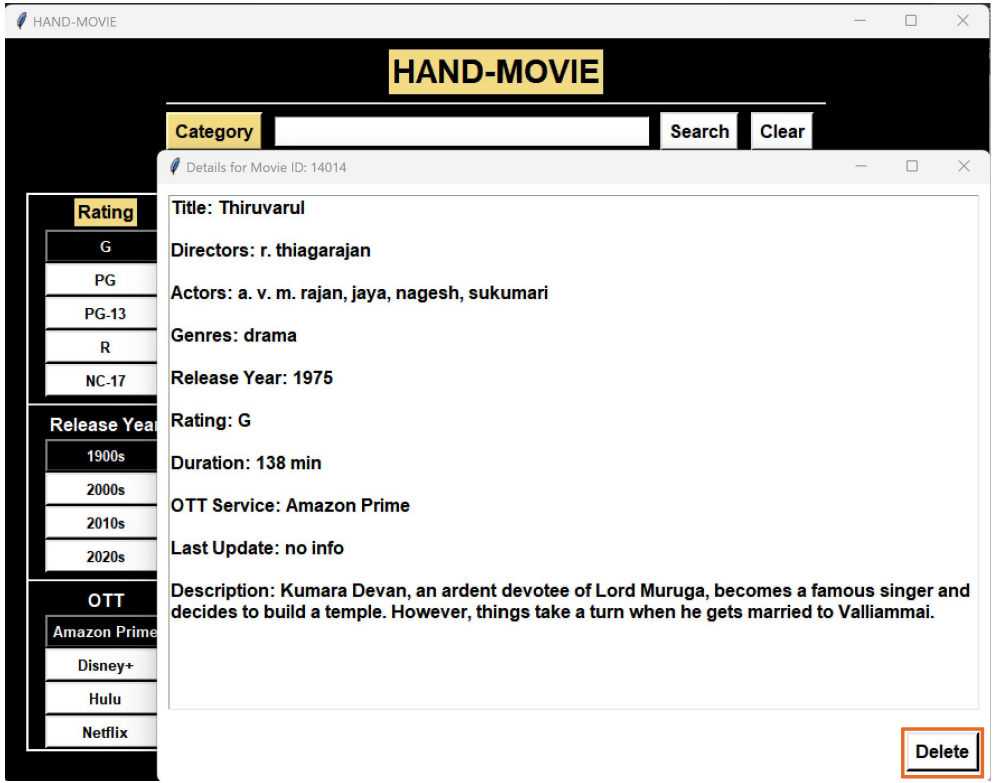
# GUI

Detail

- You can enter detail movie page by pressing detail button placed under the searching or recommendation result.

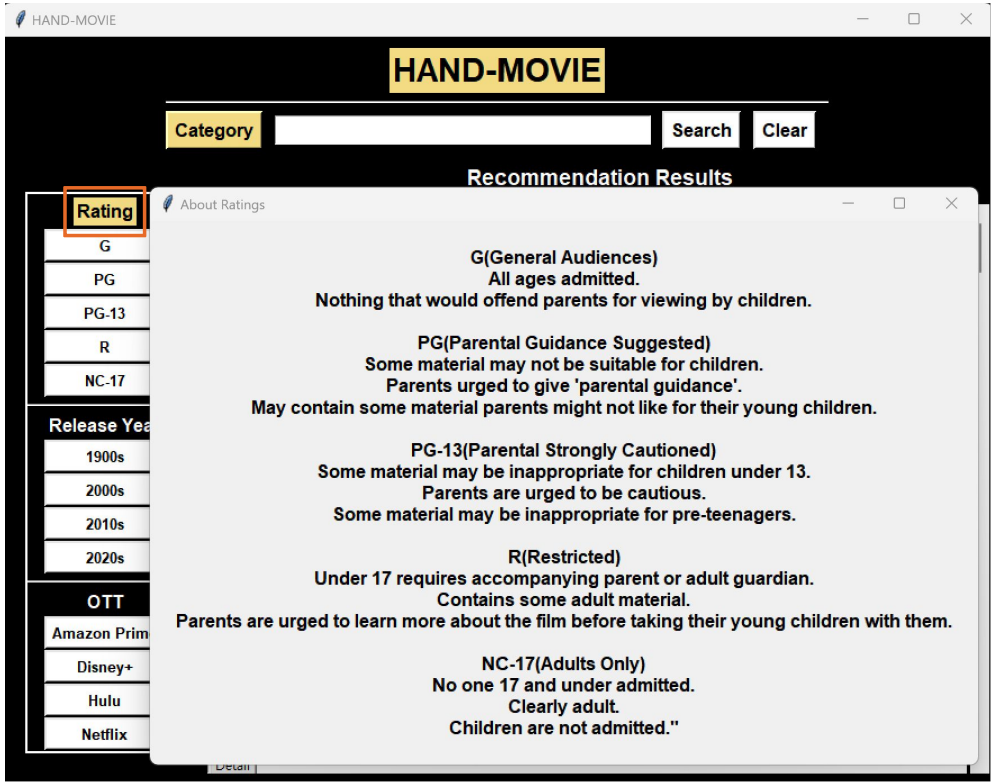
Delete

- You can delete movie by pressing delete in detail page.



# GUI

Rating info
- Press rating title



# 4. Lessons Learned

# Lessons Learned

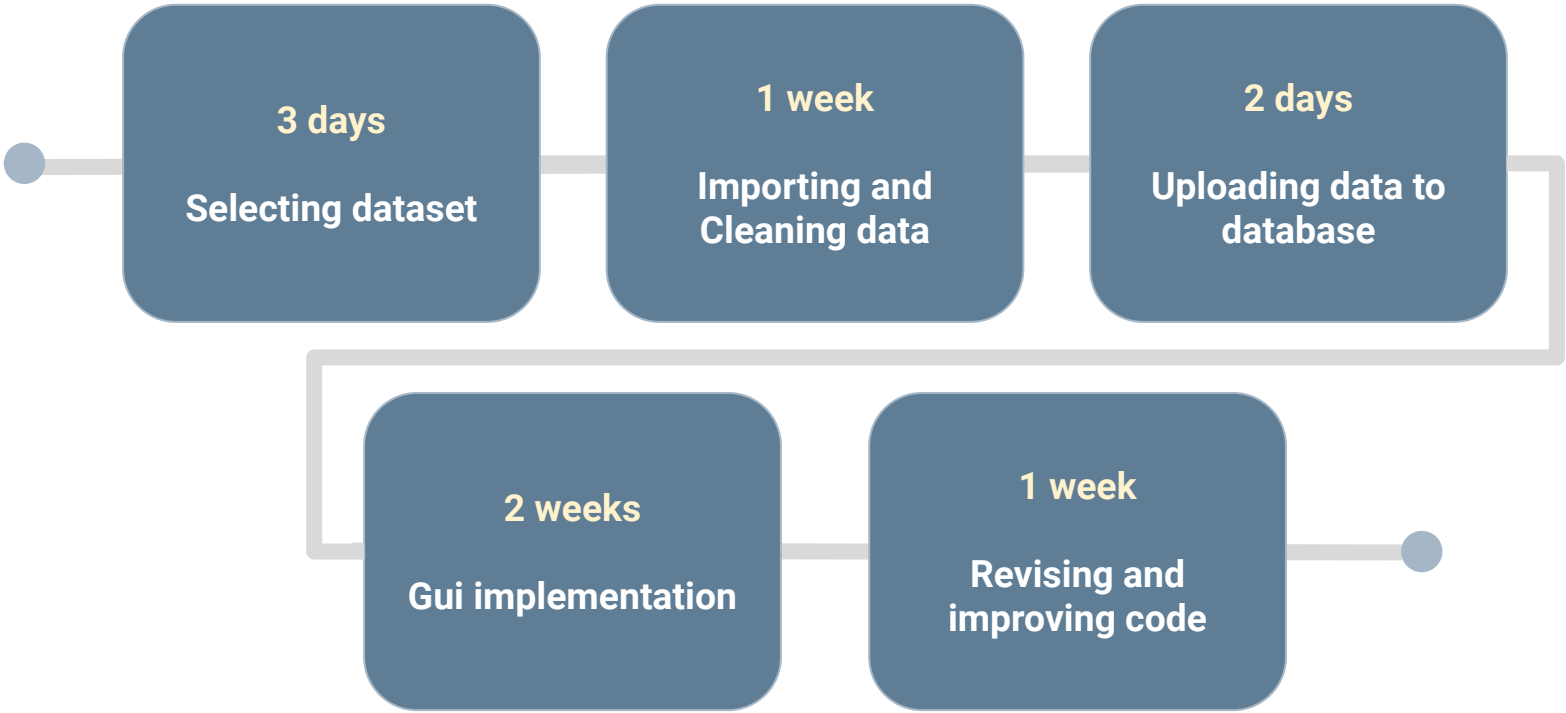
## Problems

- Inefficient Communication and Collaboration method
- Unclear graphics that does not promote usability

## Solutions

- Github for version control and development
- Color coded GUI, Button click effects, and clear word indication for functions like “Search” , “Clear”

# Timeline


3 days
Selecting dataset
1 week
Importing and Cleaning data
2 days
Uploading data to database
2 weeks
Gui implementation
1 week
Revising and improving code

# Questions?