

1. Project Title: **Cinema Finds**

2. Project Summary:

Cinema Finds is an innovative online platform designed to redefine the way individuals discover and enjoy movies. This project aims to create a comprehensive catalog of downloadable movies, complete with user-friendly features for easy navigation and exploration. Users can tailor their movie searches using various filters such as genres, cast, length, and release year, enhancing their ability to uncover hidden gems that align with their unique preferences.

Unlike conventional platforms, Cinema Finds goes beyond mainstream offerings, providing a space where users can unearth lesser-known cinematic treasures. Through user ratings of movies not found on other platforms, people can find their new favorite movie that would otherwise have gone buried. The website will not only serve as a repository for movie enthusiasts but also as a community hub where users can freely watch, share, and express their reactions, fostering a dynamic and engaging cinematic experience for all.

3. Problems to be solved:

1. Hidden gems not being available/recommended in popular streaming services such as Netflix that try to push their own products or trending movies. This works for the vast majority but many movies get neglected as a result
2. Users who have already exhausted the recommended movies from subscription services and want to find new experiences.
3. Subscription services are costly and multiple subscriptions are required to watch most movies
4. Online streaming services often delete resource and hide them from public
5. Content is often scattered across multiple platforms and even more so for content that is not widely popularized. We wish to be the platform that centralizes this untapped content.

We want to provide an alternative platform where users can discover movies that otherwise would go unrecognized in mainstream platforms where all the blockbuster movies are placed on the front page.

4. creative component and technically challenging function:

- A. to include TV series with many seasons
- B. to include related shows that are similar based on directors, tags, actors, user history.
- C. to include a recommendation system
- D. to include (possibly interactive) thumbnail pictures.
- E. to have an advanced User Management System (user profile customization, viewing history, bookmarks,).

5. basic functions and usefulness:

1. A search engine that allows users to search with name , tags, actors, directors. With a sorting feature.
2. A detailed page to include basic information of the shows. And torrent links to download.
3. User interactive component to let users rate the show, rate a torrent link to be fast/slow/dead/wrong, write comments, (with approval or privilege): publish new torrent link, edit info page, add a new show .

The distinct features of this web application set it apart from similar platforms. While RARBG served as a free torrent sharing website, it has recently closed. Our platform seeks to fill the void left by RARBG by utilizing their data archive, providing a renewed and improved experience for users. The incorporation of a user-driven, interactive component distinguishes our platform, offering a dynamic and engaging environment for movie enthusiasts. The focus on a rebirth with enhanced features and user privileges aims to create a unique space for discovering and enjoying diverse content. To mitigate potential legal concerns, our website will serve solely as an academic research demo. We will refrain from offering public pirating services, taking precautionary measures such as possibly hiding all links.

6. data sources:

Torrent links come from <https://github.com/2004content/rarbg.git> (this repo is taken down, but we do have a clone of it.), an archive of the former rarbg site. Format is txt file, 3468028 lines, each line contains one torrent link with names embedded. To avoid any copyright issues, we will not open this website to the public and we may hide the link when published.

Metadata comes from

<https://developer.imdb.com/non-commercial-datasets/> Each dataset is contained in a gzipped, tab-separated-values (TSV) formatted file in the UTF-8 character set. The first line in each file contains headers that describe what is in each column. A sample file has 13240147 lines and 6 columns.

## 7. A low-fidelity UI mockup:

Menu goes here

Recommendations goes here

Genre 1

Movie poster here

Genre 2

Menu goes here

# Movie Title (Year)

--Movie subtitle movie subtitle

Rating: \*\*\*

Movie Summary: Once upon a time, ...  
...  
...

Movie Cast >

Actor 1 Name  
Role in the movie

Actress 2 Name  
Role in the movie

Actor 3 Name  
Role in the movie

Movie Review >

Reviewer name  
Rating: ★★★

This is review 1  
detail of the movie  
... .. The movie was  
very good... ..

Download Link  
4K(2160P)  
magnet:?BAADBAADF00DF00D0000

1080P  
magnet:?BAADBABEBAADF00D0000

Fast	Slow	Dead
11	0	0

Fast	Slow	Dead
1	15	20

## 8. Project work distribution:

Frontend Designer: Kaifeng Wang (kaifeng3)

- Frontend visual appearance
- User Interface functionality
- Functional design for each respective website page
- In general, have framework design that will end up being functional once applied to API development.
- Lead overall direction of what kind of information that this app seeks to pull from the database and display to the user

Database Development: Gio Zavalza (gzava3)

- Database schema foundation
- Database tables, data types
- Analyzing the database being pulled from and constructing relational data
- Establish optimized SQL queries
- Establish and communicate the functional feasibility of data storage and access

API Development: Brian Ngeunjuntr (bngau2)

- Setup frontend-backend communication
- REST API establishment
- Make sure backend data tables are receiving accurate data and are not having the data corrupted
- Make sure frontend pulls accurate data
- Setup relational data queries

Team Organizer and Integration Testing: Bryan Jia (yizhenj3)

- Keep track of tasks to be completed and monitor pace of project development
- Be the main organizer of GitHub repo
- Test code integration and be the main merger of gitHub repo conflicts
  - If possible, optimize code or communicate optimization opportunities.
- Communicate related information that may apply between the different development tasks.
- Check overarching project details with TAs in advance