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,).

    F. to create a friend system. This friend system will be included in the search query for movie recommendations. The movie recommendation system will look at a movie's ratings, user's preferences to genres, and how popular the movie is for friends and family of the user.

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.

    3. User interactive component to let users rate the show, write comments, (with approval or privilege): edit info page or add a new show.

    4. Social Network to let friends recommend movies to each other

    The distinct features of this web application set it apart from similar platforms. While IMDB may have a massive catalog of movies, it lacks a social network. Much like Spotify with music and friends, we seek to provide a renewed and improved experience for users by letting others recommend movies. 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.

    Users will login to the website with a password and be able to select between the 'movie', 'profile' or 'social' section.

In the movie section, users may 1. Search a specific movie by title in a search engine, or 2. Select their favorite movie genres or favorite people in the film industry, to get a set of movie recommendations. 3. View implicit recommendations at the bottom of the page that are generated by previous watching habits and similar attributes.

In the profile section, a user will be able to 1. Check their watching history, rating & review history. 2. Revisit the movie or the producers they liked or followed. 3. Change personal information and privacy settings 4. Log out.

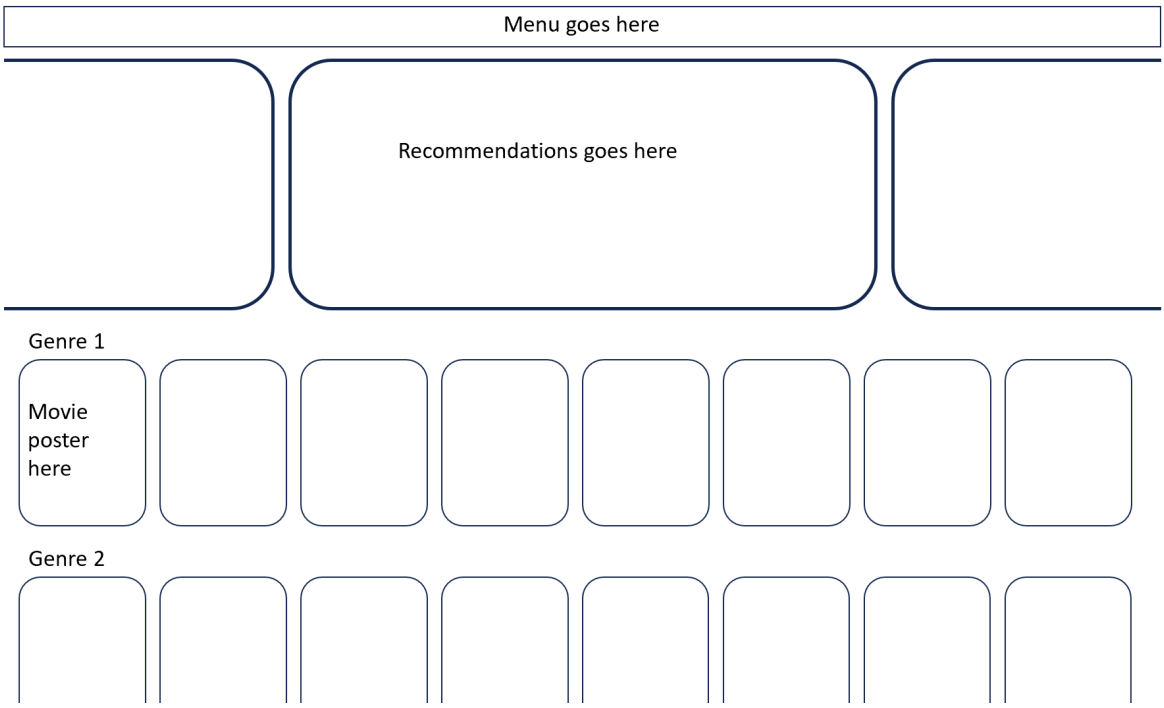
In the social section, users will be able to 1. View Friend Connect. 2. See friends' movie preferences 3. View friends' activity feed: recent watch, ratings and reviews.

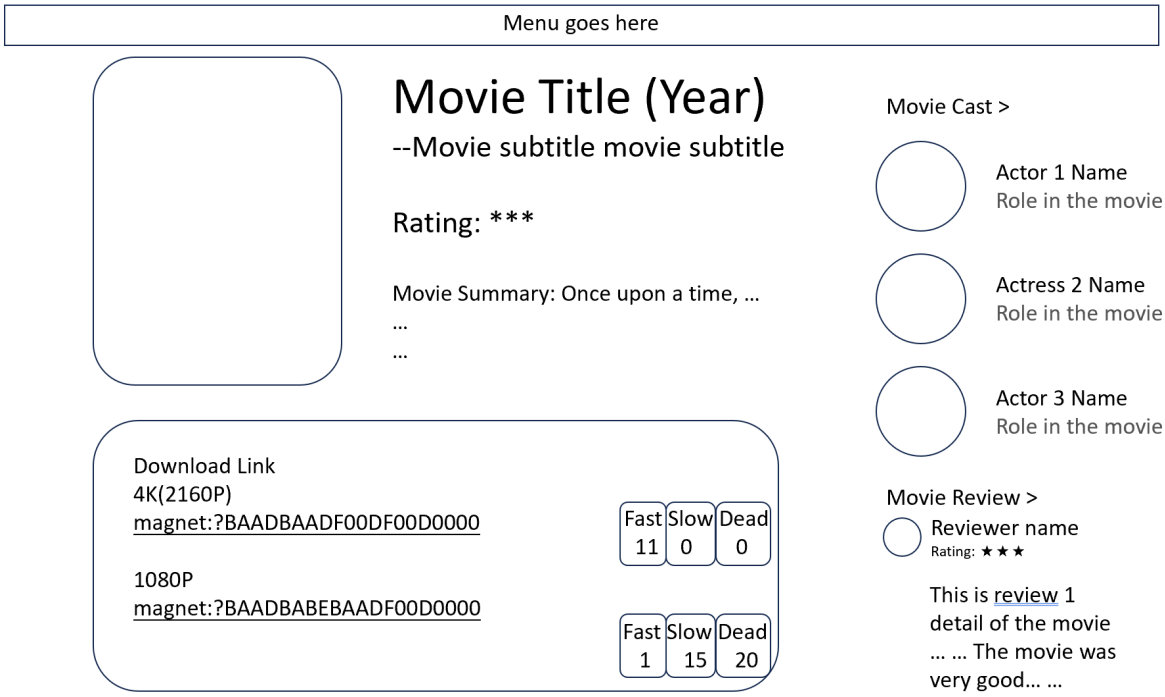
6. data sources:
   Data comes from https://github.com/2004content/rarbg.git (this repo is taken down, but we do have a clone of it.), an archive of the former rarbg site. Format is txt file, 3468028 lines, each line contains movie info with names embedded.

   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.

   Userdata:
   Userdata will be generated by random user data generating libraries.

7. A low-fidelity UI mockup:



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 (bngeu2)

- 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