Andrew Dunn & Jake Schlaerth

December 7th, 2020

CS 340: Introduction to Databases

Final Project: Movie Database

Link:

http://web.engr.oregonstate.edu/~schlaerj/

Executive Summary:

Our initial idea for a database was straightforward and simple: a database to store movies and show their relationships with actors, directors, composers, and genres. The first draft was a good foundation for our group to build upon, with no major changes being made to our core concepts and ideas throughout the project. The sections below detail the feedback, changes, and functionality we added to our project at each step along the way.

First Step: We developed the outline of our movies database, with the skeleton for all entities and relationships.
Feedback: M:M relationships should have their own intersection tables.
Fixes: We added two intersection tables for our next step, Performances (for Actor:Movie relationships) and GenreInstances (for Movie:Genre relationships)

Second Step: We continued laying the foundation for our project, creating an ER Diagram and schema. These would largely remain the same throughout the rest of the project.
There were no major changes based on feedback at this step.

Third Step: We developed the HTML code for our project, with basic layouts and placeholders for future functionality, such as add, delete, and edits buttons.
Feedback: A navigation bar would be helpful. Search bars should be added as future placeholders too.
Fixes: A nav bar and search bar were both added to our HTML pages.

Fourth Step: We developed our DDL and DML queries in SQL, adding the ability to create our database, its tables, and define its SELECT, INSERT, UPDATE, and DELETE queries. There were no major changes based on feedback at this step.

Fifth Step: We set up a server to handle the backend side of our code, and began to get some functionality working with our SELECT and INSERTs.
There were no major changes based on feedback at this step. The changes at this step were making sure that our code was live and working on each our respective pages.

Sixth Step: We developed our Update and Delete functionality for the majority of our pages.
Feedback: There are display issues when selecting your M:M tables.
Fix: The code was updated and M:M tables now display correctly.

Final Step: This step was mainly about cleaning up our code, and touching up a few final features, such as viewing genres or actors for a specific movie. We also updated the scope to accommodate less movies. We realized that the way our website is structured will not allow for an easy user experience if adding 1,000 movies. We feel the top 100 movies each year is a reasonable amount of data for our database (and users) to handle.

Statement of problem:
It is rather difficult for a user to see additional work of a director, composer, and actors in a movie. It can also be difficult to cross reference movies that fall in multiple genres. This relational database will organize these types of data in a way that will allow the user to easily see additional work by each director, composer, and actor, as well as see movies that share at least one genre. This database will include the top 100 movies released each year, based on total box office.

Movies

Attributes:

- movie_id: int, auto_increment, unique, not NULL, primary key
- title: varchar not NULL
- release_year: int
- director_id: int, a director ID
- composer_id: int, a composer ID
- actor_id: varchar, an actor ID(s)

Relationships:

- A 1:M relationship between Movie and Composers implemented with composer_id as a foreign key inside of movie
- A 1:M relationship between Movies and Directors implemented with director_id as a foreign key inside of movie
- A M:M relationship between Movies and Actors implemented with a relationship table called Performances using movie_id as a foreign key
- A M:M relationship between Movies and Genres implemented with a relationship table called Genre_Instances using movie_id as a foreign key

<u>Directors</u>

Attributes:

- director_id: int, auto_increment, unique, not NULL, primary key
- first_name: varchar, the director’s first name not NULL
- last_name: varchar, the director’s last name not NULL

Relationships:

- A 1:M relationship between Movies and Directors implemented with director_id as a foreign key inside of Movie

<u>Composers</u>

Attributes:

- composer_id: int, auto_increment, unique, not NULL, primary key
- first_name: varchar, the composer’s first name
- last_name: varchar, the composer’s last name

Relationships

- A 1:M relationship between Movies and Composers implemented with composerID as a foreign key inside of Movie

<u>Actors</u>

Attributes:

- actor_id: int, auto_increment, unique, not NULL, primary key
- first_name: varchar, the actor’s first name not NULL
- last_name: varchar, the actor’s last name not NULL

Relationships

- A M:M relationship between Movies and Actors implemented with a relationship table Performances, using actor_id as a foreign key inside of Performances

<u>Genres</u>

Attributes:

- genreID: int, auto_increment, unique, not NULL, primary key
- name: varchar, not NULL name of the genre

Relationships:

- A M:M relationship between Movies and Genres implemented with a relationship table Genre_Instances with genre_id as a foreign key inside of Genre_Instances

Genre_Instances (Genres:Movies)

Attributes:

- movie_id: int, Primary Key Foreign Key
- genre_id: int, Primary Key Foreign key

Relationships:

- An associative entity to represent the M:M relationship between Movies and Genres implemented with foreign keys from Genres (genre_id) and Movie (movie_id)

Performances (Actors:Movies)

Attributes:

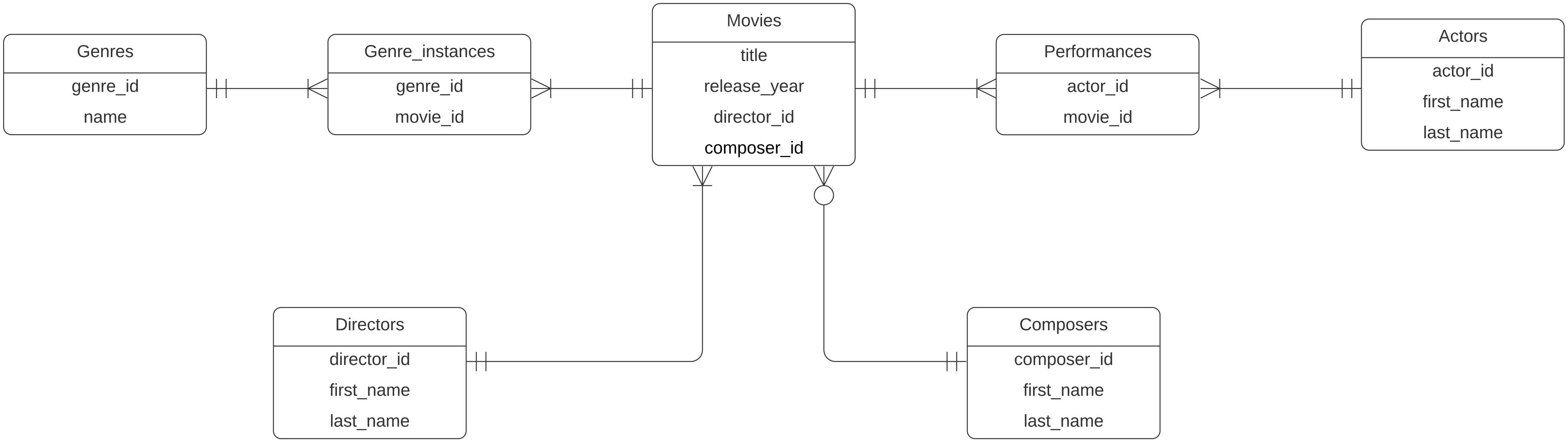
- movie_id: int, Primary Key Foreign Key
- actor_id: int, Primary Key Foreign key

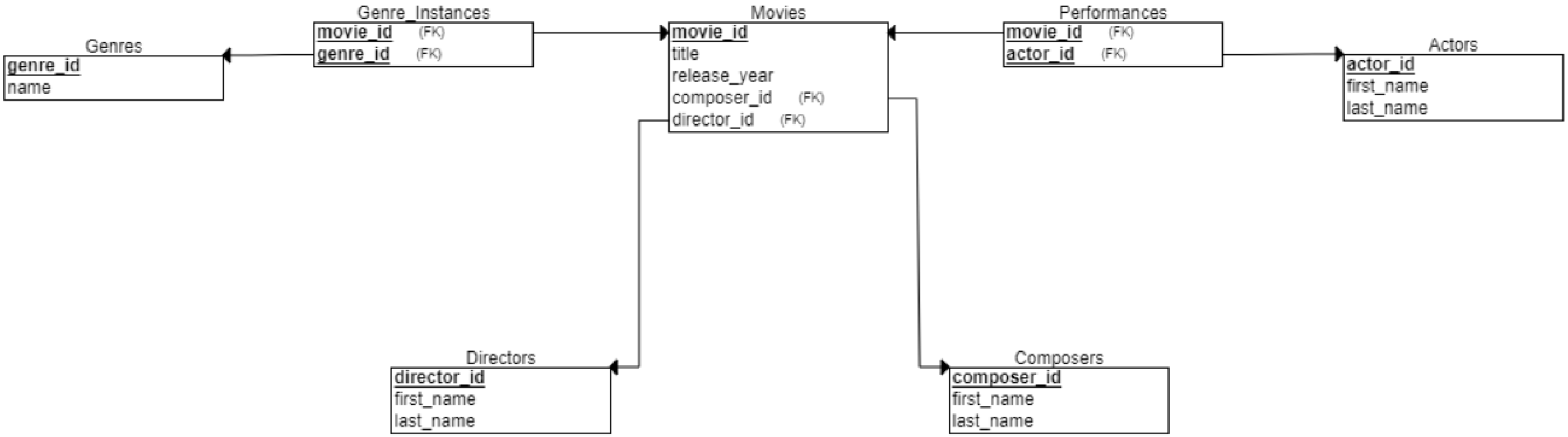
Relationships:

- An associative entity to represent the M:M relationship between Movies and Actors implemented with foreign keys from Actor (actor_id) and Movie (movie_id)

Database ER diagram (crow's foot)

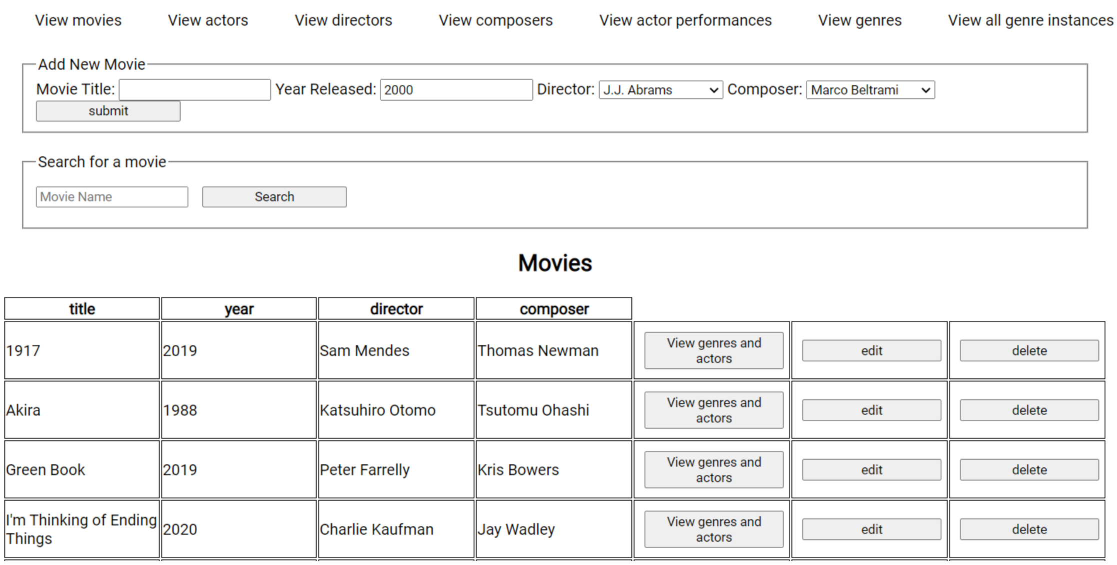
Andrew Dunn | December 6, 2020




Genres
genre_id
name
Genre_Instances
movie_id (FK)
genre_id (FK)
Movies
movie_id
title
release_year
composer_id (FK)
director_id (FK)
Performances
movie_id (FK)
actor_id (FK)
Actors
actor_id
first_name
last_name
Directors
director_id
first_name
last_name
Composers
composer_id
first_name
last_name

Movies.html

View Movies, Add New Movie, Search for a Movie, Delete or Edit Movie, or View Actors/Genres for a Specific Movie



Add New Movie

Movie Title: Year Released: 2000 Director: J.J. Abrams Composer: Marco Beltrami

Search for a movie

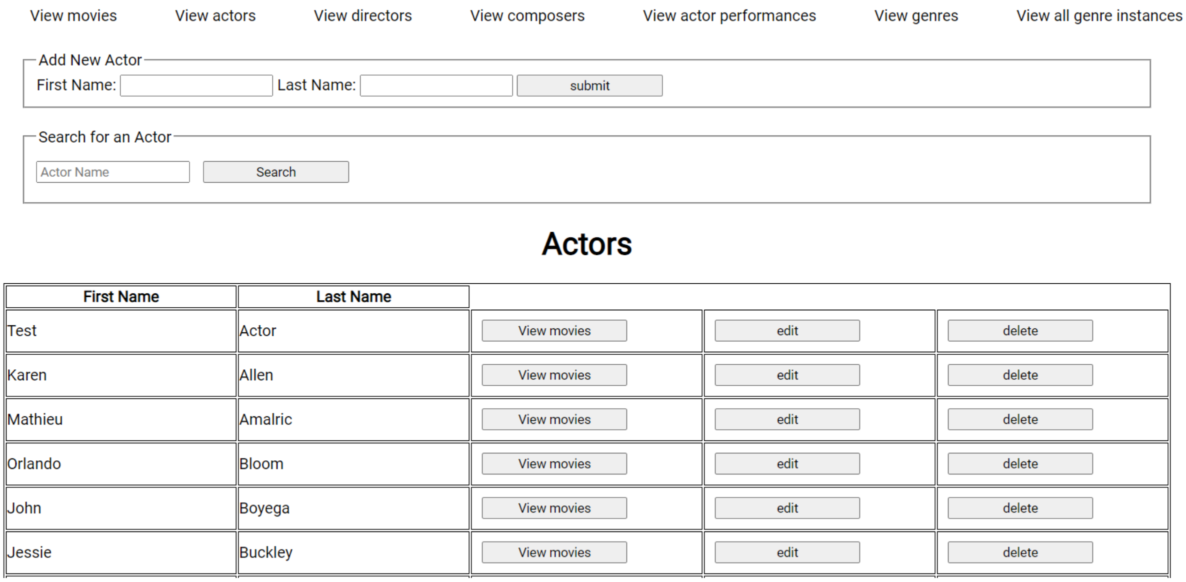
Movie Name

Movies

title	year	director	composer			
1917	2019	Sam Mendes	Thomas Newman	View genres and actors	edit	delete
Akira	1988	Katsuhiro Otomo	Tsutomu Ohashi	View genres and actors	edit	delete
Green Book	2019	Peter Farrelly	Kris Bowers	View genres and actors	edit	delete
I'm Thinking of Ending Things	2020	Charlie Kaufman	Jay Wadley	View genres and actors	edit	delete

Actors.html

View Actors, Add New Actor, Search for an Actor, Delete or Edit an Actor, or View Movies for a Specific Actor

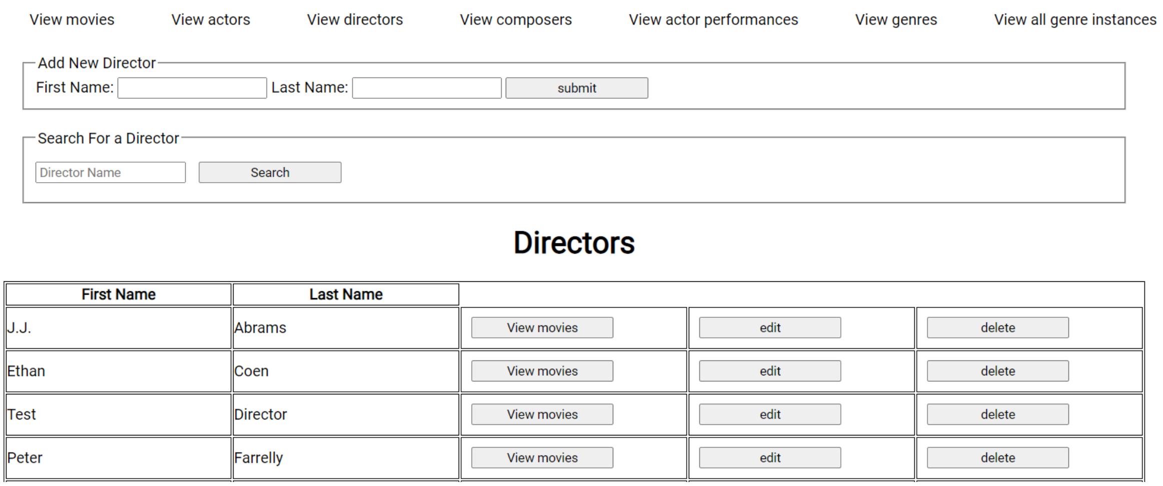


Actors

First Name	Last Name			
Test	Actor	View movies	edit	delete
Karen	Allen	View movies	edit	delete
Mathieu	Amalric	View movies	edit	delete
Orlando	Bloom	View movies	edit	delete
John	Boyega	View movies	edit	delete
Jessie	Buckley	View movies	edit	delete

Directors.html

View Directors, Add New Director, Search for a Director, Delete or Edit a Director, or View Movies for a Specific Director



Add New Director

First Name: Last Name: submit

Search For a Director

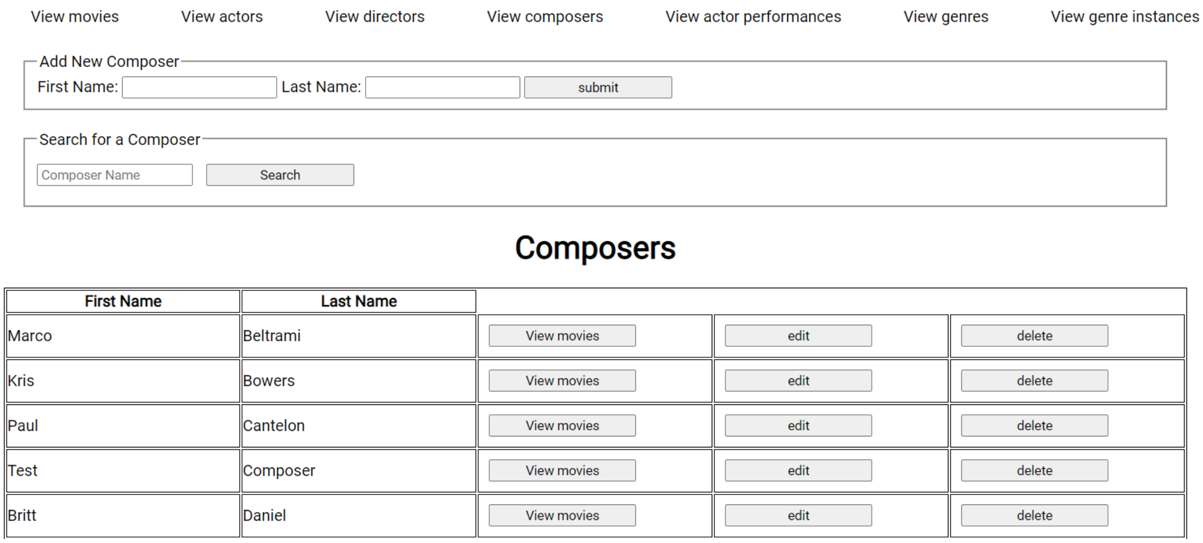
Director Name Search

Directors

First Name	Last Name			
J.J.	Abrams	View movies	edit	delete
Ethan	Coen	View movies	edit	delete
Test	Director	View movies	edit	delete
Peter	Farrelly	View movies	edit	delete

Composers.html

View Composers, Add New Composer, Search for a Composer, Delete or Edit a Composer, or View Movies for a Specific Composer

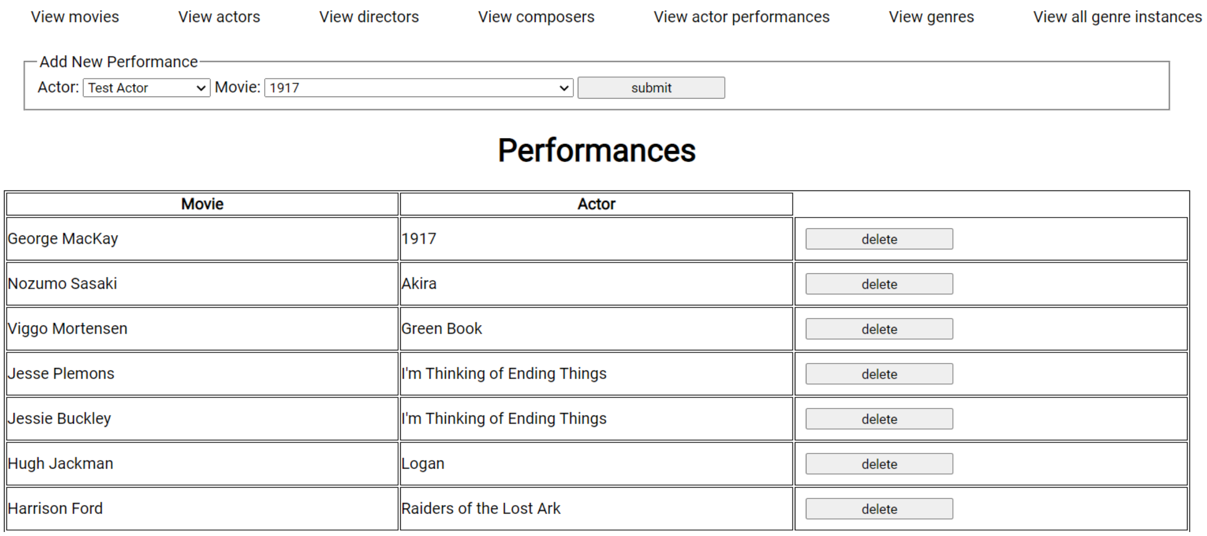


Composers

First Name	Last Name			
Marco	Beltrami	View movies	edit	delete
Kris	Bowers	View movies	edit	delete
Paul	Cantelon	View movies	edit	delete
Test	Composer	View movies	edit	delete
Britt	Daniel	View movies	edit	delete

Performances.html

View Performances, Add New Performances, Delete Performances



Add New Performance

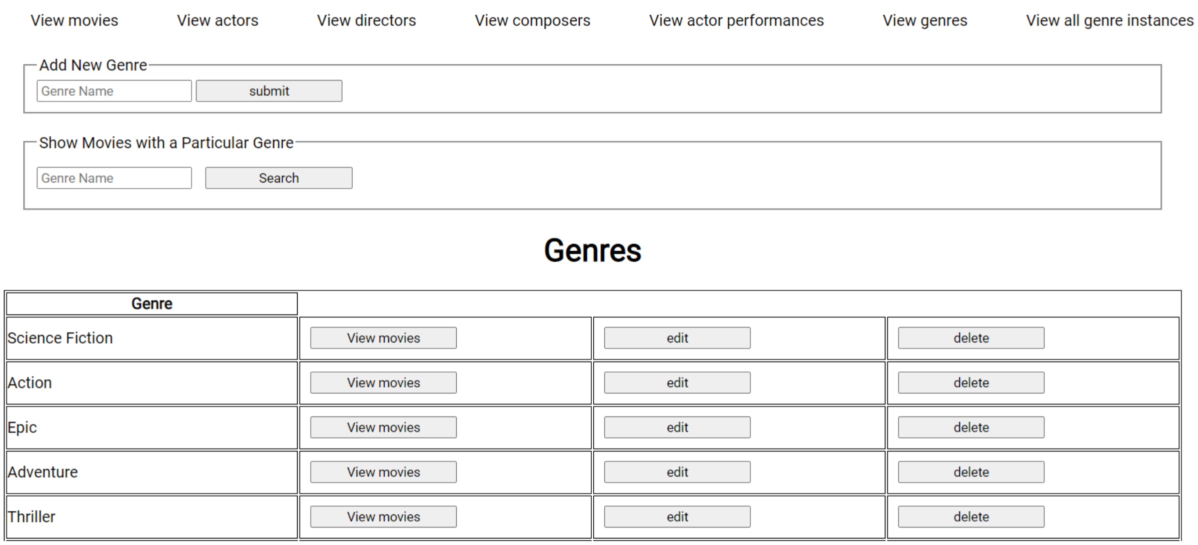
Actor: Test Actor Movie: 1917

Performances

Movie	Actor	
George MacKay	1917	delete
Nozumo Sasaki	Akira	delete
Viggo Mortensen	Green Book	delete
Jesse Plemons	I'm Thinking of Ending Things	delete
Jessie Buckley	I'm Thinking of Ending Things	delete
Hugh Jackman	Logan	delete
Harrison Ford	Raiders of the Lost Ark	delete

Genres.html

View Genres, Add New Genre, Search for a Genre, Delete or Edit Genre, or View Genres for a Specific Movie

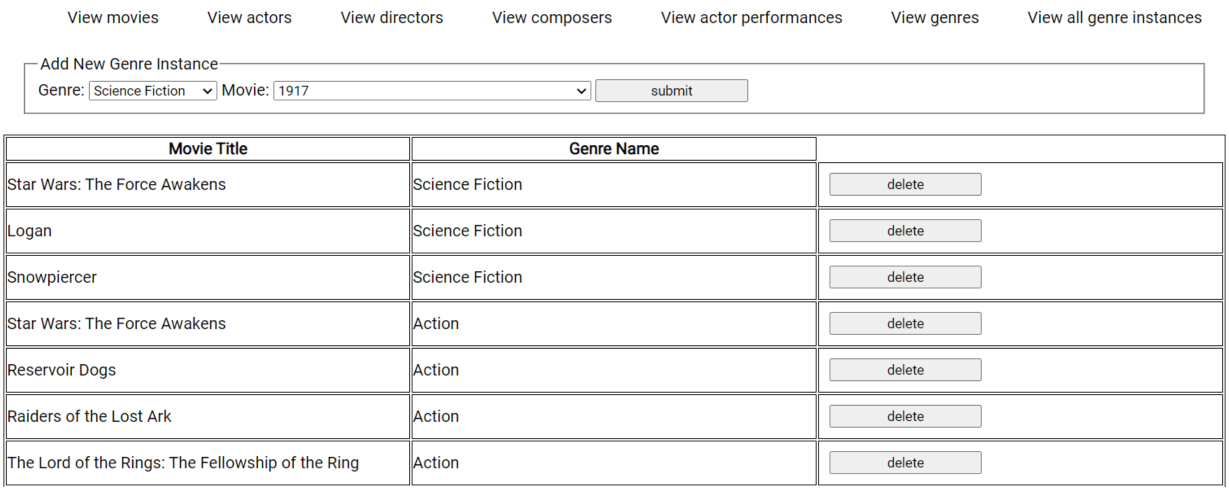


Genres

Genre			
Science Fiction	View movies	edit	delete
Action	View movies	edit	delete
Epic	View movies	edit	delete
Adventure	View movies	edit	delete
Thriller	View movies	edit	delete

GenreInstances.html

View Genre Instances, Add New Genre Instance, Delete Genre Instance



Add New Genre Instance

Genre: Science Fiction Movie: 1917

Movie Title	Genre Name	
Star Wars: The Force Awakens	Science Fiction	delete
Logan	Science Fiction	delete
Snowpiercer	Science Fiction	delete
Star Wars: The Force Awakens	Action	delete
Reservoir Dogs	Action	delete
Raiders of the Lost Ark	Action	delete
The Lord of the Rings: The Fellowship of the Ring	Action	delete