FUNCTIONAL REQUIREMENTS

User Registration and Login:

Users should be able to log in securely using their credentials (username/email and password). The database needs to store user login credentials securely. This includes storing hashed passwords and validating user credentials during login.

-There must be a USERS table or entity to implement this.

USERS has attributes userID, username, fullName, Password, email, dateOfBirth, age, gender, address, contactDetails, salt, country, languagePreference, isAdmin, apiKey.

Fetching the movies or series:

Users should be able to retrieve movie information from the database. : The database should store movie details such as title, genre, release year, director, cast, etc. It should support efficient querying to retrieve movies based on different criteria.

-Include two entities called MOVIE and SERIES.

MOVIE has attributes movieID (primary key), title, genre, releaseYear, director, productionCompany, cast, plot, rating, duration, language, country, reviews, userRatings.

1. **Title**: The name of the movie.
2. **Genre**: The category or type of the movie (e.g., action, comedy, drama).
3. **Release Year**: The year when the movie was released.
4. **Director**: The person who directed the movie.
5. **Cast**: The main actors/actresses who star in the movie.
6. **Plot**: A brief summary or description of the movie's storyline.
7. **Rating**: The movie's rating, such as MPAA rating (e.g., G, PG, PG-13, R) or IMDb rating.
8. **Duration**: The length of the movie in minutes.
9. **Language**: The language(s) in which the movie is available.
10. **Country**: The country where the movie was produced.
11. **Production Company**: The company responsible for producing the movie.
12. **Reviews**: Critical reviews or audience feedback about the movie.
13. **User Ratings**: Average ratings provided by users who have watched the movie.

SERIES has attributes seriesID (primary key), title, genre, releaseYear, creator, productionCompany, cast, plot, rating, seasons, episodes, duration, language, country, reviews, userRatings.

Create entity **ACTOR**

-An actor **ACTS** in a movie/series.

ACTOR has attributes actorID, name, dateOfBirth

Create entity **CAST**

A MOVIE/SERIES has a CAST

It has an identifying relationship derived from MOVIE/SERIES

CAST has attributes movieID, actorID, role, billingOrder

Create entity **PRODUCTION_COMPANIES**

A production company produces a movie/ series

PRODUCTION_COMPANIES has attributes companyID, name, headquarters, foundingDate, CEO, description.

Favourites page:

Users should be able to view and manage their favourite movies. The database should store user-favorite associations, linking users to the movies they've marked as favourites. This will involve a many-to-many relationship table. Create a new entity called FAVOURITES.

FAVOURITES has attributes userID(foreign key) and movieID/seriesID (foreign key),

When a user wants to add a movie/series to their favorites, a new entry is inserted into the "FAVOURITES" table with the user's ID and the movie's/series' ID.

When a user wants to remove a movie from their favorites, the corresponding entry is deleted from the "FAVOURITES" table based on the user's ID and the movie's/series' ID.

Users can view a list of their favorite movies by querying the "FAVOURITES" table for all entries associated with their user ID and joining with the "movies/series" table to retrieve movie details.

The system can check if a particular movie/series is marked as a favorite for a specific user by querying the "FAVOURITES" table with the user's ID and the movie's ID.

Recommendations:

The database should store user viewing history and preferences. It should implement algorithms to generate recommendations based on this data.