Dream to Stream
Stage 4 Midterm Demo

Requirements

1. Insert new records (rows) to the database and reflect the change on the frontend interface:
   - Insert new records into the Users table when a created user is added.
     - When you sign in, this is retrieved from the Users table to reflect the update
   - SQL Statement: `INSERT INTO Users VALUES (${next_id}, '${firstName}', '${lastName}', '${dob}', '${country}')`

2. Search the database using a keyword search. Your application should allow the user to input their search keyword and return the result to the interface:
   - Using some advanced queries, allow for users to Search titles from Shows and Movies providing the inputs and getting ratings of shows/movies with the highest rating that year and sort them by Name
     - Genre
     - Year Range
   - SQL Statement:

     "SELECT name, m.releaseYear, value genre
     FROM Movies as m Natural JOIN MovingRating as mr JOIN (
     SELECT MAX(value) as mxs, releaseYear
     FROM Movies Natural JOIN MovieRating
     GROUP by releaseYear
) as maxs ON maxs.releaseYear=m.releaseYear
     WHERE genre LIKE '%${genre}%' and mr.value=maxs.mxs and m.releaseYear between ${sYear} and ${eYear}
     ORDER BY name'

   - Search 2 allows for users to find movies, users, and values of ratings from Users who live in a country where the specified Platform (Netflix, Hulu, Disney+, Prime Video) is supported. Then display the results in a table
   - SQL Statement:

     "SELECT name, firstName, lastName, value
     FROM Users

```
NATURAL JOIN MovieRating
WHERE country
IN (
        SELECT country
        FROM Platform
        WHERE platform = ‘“
         + platform +
        “‘) ORDER BY movieName;”
```

3. Update records on the database and reflect the change on the frontend interface
   - Allow for users to select the Update tab where they are able to select Update based on their sign in ID. This will allow for them to update their rating on a Show/Movie provided in their input parameters. After the update is done, it will take them to a page that shows a user all their ratings based on their ID which will reflect the change.
   - SQL Statement:

     `UPDATE ${Movie/ShowRating} SET value = ${rating} WHERE id = ${userId} AND name = ‘${media}’`
4. Delete rows from the database
   - Allow for users to select the Delete tab where they are able to select Delete based on their sign in ID. This will allow for them to update their rating on a Show/Movie provided in their input parameters. After the delete is done, it will take them to a page that shows a user all their ratings based on their ID which won’t show the deleted record.
   - SQL Statement:

     `DELETE FROM ${Movie/ShowRating} WHERE id = ${userId} AND name = ‘${media}’`

5. Integrate into your application both of the advanced SQL queries you developed in stage 3.
   - Integrated the advanced SQL queries to allow users to search based on given parameters as described in section 2.
   - SQL Statements:
     1. “SELECT name, m.releaseYear, value genre

        FROM Movies as m Natural JOIN MovingRating as mr JOIN (

        SELECT MAX(value) as mxs, releaseYear

        FROM Movies Natural JOIN MovieRating

GROUP by releaseYear

) as maxs ON maxs.releaseYear=m.releaseYear

WHERE genre LIKE '%${genre}%' and mr.value=maxs.mxs and m.releaseYear between ${sYear} and ${eYear}

ORDER BY name'

2. "SELECT name, firstName, lastName, value
   FROM Users
   NATURAL JOIN MovieRating
   WHERE country
   IN (
   SELECT country
   FROM Platform
   WHERE platform = '"
   + platform +
   "') ORDER BY movieName;"