Stage 3 Report

Group Members: Owen Eugenio, Ruby Matheny, & John McLeod

Final choice of relational databases: PostgreSQL
Software platforms/languages: Flask(Python)
Milestones:

1. User Authentication
2. Design how the website looks
3. User Profile
4. Build out database functions
5. Build service to collect data from IMDB
6. Build website

Week 1-3: Complete backend
Week 4-5: Complete frontend

DOL:
John- User Entity + APIs gateways to interact
Ruby- Movies, Director Entities + APIs gateways to interact
Oli- Genre, Actor Entities + APIs gateways to interact

Data source:
IMDB api for movie data
We will generate user data

Relational Schema:

User (uuid, email, password)

Profile(pid, name, sexualPreference, gender, age, match, favoriteActor, favoriteMovie, favoriteGenre, favoriteDirector, FOREIGN KEY uuid REFERENCES User, FOREIGN KEY password REFERENCES User)

UsersProfile (user, profile)

Match (profile1, profile2)

FavoriteGenre(profile, genre)

FavoriteMovie(profile, movie)

FavoriteActor (profile, actor)

FavoriteDirector(profile, director)

Actor(a_id, name, actedInMovie, actedInGenre, isFavoriteOf)

ActedInMovie(Actor, Movie)

ActedInGenre(Actor, Genre)

Genre(genreName, actedInGenre, belongsToGenre, hasDirectedGenre, isFavoriteOf)

BelongsToGenre(genre, movie)

HasDirectedGenre(genre, director)

Director(d_id, name, directedMovie, directedInGenre, isFavoriteOf)

DirectedMovie(director, movie)

Movies(title, year, metaCriticScore, actor, director, genre, isFavoriteOf)