Wishlist Application

An application where you can create a wishlist for various things.

The Python Flask web application will connect to the MySQL db on GCP through configurations

**How to generate the “production” dataset and load it into database.**

Raw Data Link:

<https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset/version/1>

We downloaded the movie data from the above link, the form of a CSV file with 5k data.

After getting the movie data, we created a database on CloudSQL and defined the schema for the data we would populate. The DDL code are presented in the **project.sql** file. After defining the schema, the movie data was imported to the database movies table. The recommendation data is essentially a subset of this data based on dfferent type of queries.

Other production data related to the app include user data and wishlist data. When a user register to the application, we obtained data from the user such as their chosen “username” and “password”, and our backend built out custom database tables to account for this. A unique “userID” will be created for each user based on uuid function to establish a primary key. This is present in the app.py file under login and register function.

A user session is created once a user logs into the web application. When user clicks the "Add to wishlist" button, the movieID, userID in session, and time will be inserted to the wishlist table. When a user clicks on delete from wishlist button on the wishlist page, this would delete the movie from the user’s wishlist table.

**6 Features implemented and files that contain the implementation**

**Feature 1: Register Functionality**

When the user first arrives at our webpage, they are greeted by a generic login page. Here, the user will be able to either register or login.If the user presses the “Register” button, they are taken to a register page, where they can enter in their username and password. Once they have completed their sign up and presses the “submit” button, they will be successfully registered in the database. There will be error messages to guide the user throughout the registration process, such as “username is not long enough” if they try to press “Submit”

**Files:**

app.py (register function) , templates/register.html , templates/land.html

**Feature 2: Login/Logout Functionality**

User login to the movie wishlist app and remember login session. Once a user finishes register, they can then go to the login page to login to their new account. User can also logout by clicking on the button in the top right.

**File:**

app.py (login function and is\_user\_logged\_in function), templates/land.html, templates/login.html

**Feature 3: Returning all the movies available and sorting by various requirements**

After a successful login, the user will be brought to the homepage. At the top of the homepage, there will be dropdown menus allowing the user to filter their movie search based on various attributes like “genre”, “ rating”, “director name”, etc. Once they have completed their selection from the drop down menu, they will be able to click on the “Submit” button, and the selection tables at the bottom of the page will refresh, displaying all of the movies that match the user’s search criteria. Here is a snapshot showing off the filter feature using the “genre” category. In addition to drop-down menus, the top selection menu also contains a text field search bar. This is intended to allow users to search for movies directly by name. Users can enter text into the search bar, and the selection table will update automatically (no need to press submit) to display movies matching the text search. The text search will still abide by the current selection criteria. So for example, if the user has currently filtered for “Romance” movies, and types “abc”, the selection table will display romance movies containing “abc” within the title. Here is a snapshot of this feature where the user searches for the film “wolf of” and “Wolf of Wall Street” appears because it’s the only text search match!

**File:**

app.py (dashboard function) and home.html

**Feature 4: Adding movies to wishlist**

Within some movie items of the selection tables, there will be an “add to wishlist” button to the right of the rows. If a movie has this button within its row, it means that the movie is not a part of the user’s wishlist. Users can click on the button to add the movie to their wishlist. Once clicked, the “add to wishlist” button’s text changes to “remove from wishlist”

**File:**

app.py (processing\_wishlist function and dashboard function), templates/home.html and templates/layout.html

**Feature 5: Displaying wishlist**

On the top selection bar of the web app (where users filtered for movie selections), there will also be a button representing a user’s wishlist. If users want to view all movies within their wishlist, they can simply click on that button, and the page will refresh, override existing search queries, and display all the movies added to the user’s wishlist.

**File:** app.py (wishlist function), templates/wishlist.html

**Feature 6: Providing different recommendations**

On the recommendation tab, the user can view various recommendations such as recommening most popular movie with respect to director, most popular movie of the year, top rated movie in each genre.

**File:**

app.py (recommendation function), templates/recommendation.html

**Feature 7: Changing Username**

User can change their username.

**File:**

app.py (profile function), templates/profile.html