Final Project Report

Team 117: Netflix Wrapped

Group Members: Megha Chada, Krish Patel, Kaylan Wang

Please list out changes in the directions of your project if the final project is different from your original proposal.

- When we first completed the proposal, we were hoping to show users an analysis of their different viewing habits as well as offer more recommendations based on their history. However, this is one aspect that we did not include in the final application as we wanted to focus on the analysis portion and ensure that it was accurate. Additionally, instead of using a dataset from kaggle.com like we had initially proposed, we decided to use the imdb develop datasets to do our project. These datasets are constantly updated and offer more variety in metrics than the ones we found online from Netflix which helped us in designing our database properly.

Discuss what you think your application achieved or failed to achieve regarding its usefulness.

- The application is able to show the user interesting trends in their watch history, such as the genres they watched or the total hours watched of different types of media like movies and tv series. The website can identify trends like which directors the user likes the most and display other titles from the director. Additionally, the user is able to add and delete to their watch history with the insights updating in real time, as well as keyword search for media. Something we did not implement was the content recommendation based on watch history.

Discuss if you changed the schema or source of the data for your application.

- At the beginning of the project, we were going to use a dataset from kaggle.com that took information from Netflix itself on what content they offered. However, we decided to use the imdb datasets instead because they were more thorough and we were able to combine different columns from different tables to create the exact metrics we wanted to see for each title.

Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?

- In stage 2, when we first completed the ER diagram, there were some attributes to the main Title table that we included which were later removed because they were unnecessary. These attributes were originalTitle, language, endYear, and genres. These attributes were already being used in different tables and didn't need to be repeated or were redundant to other attributes in the table. Additionally, we added unique identifiers to the genre, director, and language tables because those were missing in the initial ER diagram. The relationships between Title and these three tables were also changed from initially being weak to being strong. However, the cardinality of these relationships remained the same. When creating the actual DDL commands for the application, some of the naming conventions were different than what was in the ER diagram. However, the same tables are in the application as what was in our second edited ER diagram. The second ER diagram that we produced was more accurate to the application we were hoping to create and took into account the SQL queries we wanted to implement. The first one was more based on the dataset we were looking at but wasn't specific to what we were trying to achieve. By editing the diagram in stage 3 based on what queries we wanted to complete, it became more specific and accurate.

Discuss what functionalities you added or removed. Why?

- One functionality that we added was the ability to sort the output data by year. This way the user is able to see their trends and other interesting information about their habits over the course of each year. The reason we added this functionality is because the inspiration behind this project was Spotify Wrapped, which releases a synopsis of the user's listening history over the course of one year. We ended up removing the recommendation functionality due to some time constraints and we wanted to make sure the analysis of the data was correct and made sense.

Explain how you think your advanced database programs complement your application.

- Using advanced database programs like MySQL helped in the efficient data storage and access. It optimized the data storage and ensured efficient use of the resources and faster access to information. Storing millions of records on a local drive or just some cloud platform and using programming languages like python would have taken significant time to compute the desired results. Using indexes in a database and queries to access the data optimized the use of ram and cloud to compute this data. Moreover the scalability of this application increased with the help of MySQL . It enabled seamless scaling of applications as data volume increases. Adding data to the user table for instance when multiple users are on the same page would require a store cloud platform and long compute times but with the use of MySQL indexing directly into session will keep the compute time short.

Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.

- Krish: One technical challenge that I faced was trying to isolate the user data between each session. Making the website atomic with respect to other users was quite important as otherwise the data would exchange between users and display incorrect results. To solve this we implemented sessions in the code. The sessions were linked with each account that tried to access the website. This was then referred to using a key in the user table and the temporary display table that was created. Keeping track of these sessions also allowed for more data about the website like minutes they spend on the website and how many visits are on the website per month.
- Megha: One technical challenge I encountered was navigating the GCP and utilizing that throughout the project to edit code and test queries. I have never used a platform like this before and was hesitant with it at first which slowed me down a lot. I also struggled with understanding the connection between the server.js file to the other files in the repository. I think physically going to the workshops in person rather than watching from home is important to fully understand these new concepts and get more comfortable with them especially if you're someone who isn't familiar with it at all.
- Kaylan: One challenge that we faced was making the frontend of the data queries look presentable. For example, we wanted to display the total hours the user had watched on Netflix over the course of a year and sort it by type of content (movies, tv shows, etc.). Displaying this data on the frontend of the application in a way that was clear and looked good was difficult. Over the course of this class, we learned how to create queries and relational schema to interact with our databases but did not learn how to properly display the information that we query.

Describe future work that you think, other than the interface, that the application can improve on.

- One aspect of the application that can be improved in the future would be to include another page after analysis that gives users detailed recommendations on what other content they would be interested in seeing. For instance, there could be a button at the bottom of the analysis page that leads to a recommendations page and this would take the analysis and give them new movies and tv shows to watch based on the genres they enjoy, the shows they've binged in the past, and their favorite directors.

Describe the final division of labor and how well you managed teamwork.

- The work was split evenly between the three team members. Creating the database schema, UML diagram and working on vital functions of the projects like triggers, frontend backend connect was done by Megha. Working on normalization, vital functions of the projects like transactions, and frontend backend connections was done by Kaylan. Krish worked on data cleaning, python script for customizing this data to be used to compute the queries, front end back end work and working with server.js files with sessions. We all worked together on the advanced queries.