**Project Name: Popular Apps!!**

Team Members: Shweta Sharma, Addison Ji, Bezawit Abebe

Project Goal:

Our group is trying to extract Google Play Store and Apple Store’s Application reviews and ratings data, transform them to one dataset that contains unique Apps found in both stores. We will then compare them by total reviews from both stores and the average ratings of the two stores. This will now be loaded to a MySQL database. From the database, we are able to inquiry on the most popular **Apps,** their **Reviews** and **Ratings** detail. To showcase our web scraping skills, we decided to web scrap an article from Mashable that has the top 20 downloaded Apps and load it to another table separately. Then we can query using SQL to see if an App is in both tables or not.

Data Source:

* Google Play Store dataset: Kaggle (<https://www.kaggle.com/lava18/google-play-store-apps>)
* Apple Store dataset: Kaggle (<https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps>)

They are both can be downloaded as a CSV file from the link provided above.

* Mashable Article ( <https://mashable.com/article/apple-most-popular-iphone-apps-2018/> )

Transformation:

1. Select the columns we need from all columns on both datasets.
2. Trim all the spaces on all fields.
3. Change the font of the app names to lower case for future merge need.
4. We found there are duplicated App names with either the same “Total Review” and “Ratings” or different numbers. We want to consolidate the duplicates to one unique record per App by selecting the most “Total Reviews” and the best “Ratings” per App if they have multiple records. In this way we can remove the duplicates and have one unique line of record per App.
5. In order to add up the total reviews in the next steps, we need to convert “Total Review” to a numerical data type for both tables. When we converting it, we found our dataset has text such as 3M instead of a number 3,000,000 which is throwing out an error. We using filter to filter out the record and change the text to actual number and then covert the column to numerical again.
6. In order to take the average in the next steps, we convert the “Ratings” column to numerical for both tables.
7. We adjust the column names identically for Google Play Store table and Apple Store table.
8. Then we perform the inner join of two tables on the App Names. We found App Names can be spelled so differently in both stores, but due to the limitation of the knowledge of the regular expression and the timeframe, we decided to take the results that spells exactly the same in both tables.
9. We created two new columns, one stores the total reviews, calculated by adding up App store total reviews and Google Play Store total reviews, and the other one stores the average ratings of the two.
10. The final table, we take the App Name, the new total review and ratings column as well as the Genres from the Google Play Store dataset as we found it’s more accurate.

Loading to a Database

After creating our final table, we created and loaded this information into a SQL database.

1. Created a database called PopApp
2. Within PopApp we created 2 tables:
   1. Applications – this table will store the merged app table. We created 4 columns, App\_Name which is a string, Genres which is also a string, Total\_Reviews which will hold integers and Ratings which we set as float since these were decimals.
   2. Top Downloads – This table will hold the list of top 20 apps which we scrapped from the Mashable article. We created only 1 column in this table called Top\_App\_Names which will hold strings for the list of apps.
3. Load the pandas dataframe “final” into the SQL table Apps and the dataframe “Top App” from the webscrape into the SQL table Top Downloads.

Because both of these tables are using MySQL they are both relational tables.