Project Title: PlayRECS

Project Summary:

Our project aims to aid parents who have children who are interested in gaming by allowing them to search and find games for their kids that would be safe for them. We will be creating a web application that takes in user input data and stores it which we will then use to aid in the search of steam's game database for game titles. An example of this would be how we take age as data from our user and then filter the data that our user searches for based on the age we were given. This project will involve designing the web application to have a page where we can take in a search and display results and also a page where we will visualize collective data from all users. Our project will also include data visualizations for the users which will help them see what kind of things the community of users is searching for and can help parents see what kind of genres are currently popular. In conclusion, the problem we aim to solve is that parents don't want their children being exposed to mature games at a young age so we want to help parents find games for their children that would be suitable for their maturity level.

Usefulness:

The website will be most similar to the Steam store's search function, with some useful upgrades. The Steam store has no option to filter out tags, which is an issue since a user who dislikes a genre cannot remove games with that genre in a search. There is also no filter for movies, since the Steam "Movies" tag includes filmmaking software. Also, there is no convenient way for parents to filter out games by genre and age group. Finally, the visualization of user preferences will be different from Steam, and will be a convenient way for users to find popular games by user age and price.

Functionality:

Allows users (regular games or parents) to search games based on inputted data. Users can filter data based on game title, age, media (movies/tv shows), price, and DLC count. The search will consider all inputs provided by the users. For example, a user can search only by age or search by age, price, and title.

Our application will also consider user types, parents of users under 17 and users over the age of 17. According to the user type, certain game titles will be filtered-out. If the user type is "Parent", they can set child age, banned titles, banned genres, and banned prices ranges. When a child searches games through our platform, the searches will also consider the attributes of the user type. If the user type is "Gamer", they are 17 years old or older and will not have restrictions. In this situation, the user type table will have NULL data for each attribute (child age, banned titles, banned genres, and banned price range). A "Gamer" type will have access to all data. The search data determines what game titles are displayed for the user.

Stored Data/Realness:

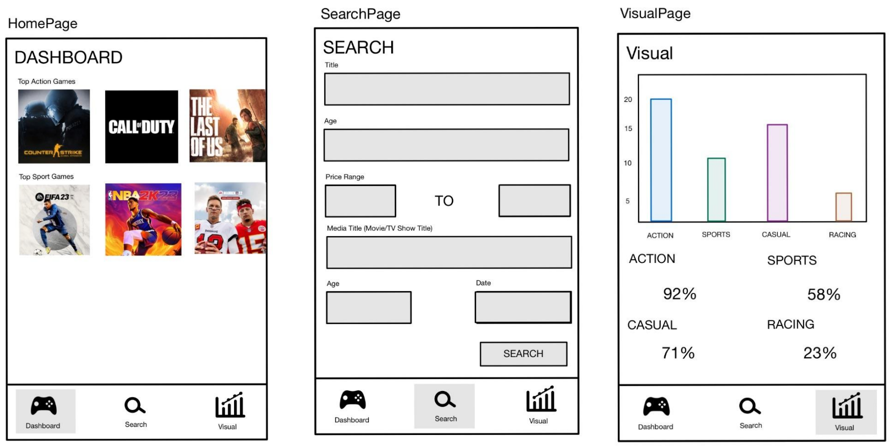
The data we will be using is from a steam data set from 7 years ago. The website we acquired this data from is https://data.world/craigkelly/steam-game-data. The data that we have stored in the database is things like age restrictions, titles, prices, player counts, etc. The attributes of the games available on steam are what we have stored within our database, but we will also be creating our own databases to store data that our users will provide for their preferences.

The data we will be collecting from users consists of age, in order to filter based on age restrictions for parents. We will also be collecting users' liked genres in order to search based on that information as well. We will also be collecting banned titles and genres from our users in order to help filter our results for games that a parent could get for their child. Finally, we will also be collecting a price range from parents that we will use in order to display data to them depending on how much money they are willing to spend.

Creative Component:

We will visualize user preferences through graphs/charts by summarizing search trends sorted by genre, user type, price, etc. For example, we could use a chart to show how much each age group searches for a particular game genre. We will visualize data by showing a graph or percentages of all user data collectively. Our creative component will NOT show personal data for each user.

Low-Fi UI Mockup:



Project Work Distribution:

<u>Peiqi</u> : In charge of creating data visualization and queries concerning the data visualizations.

<u>Jeffrey</u>: In charge of taking inputs and inserting into user tables or managing user data based on queries and in charge of updating and deleting from user tables based on user inputs.

<u>Milind</u> : In charge of creating the general search functionalities and filtering.

Alexis : In charge of front end development and understanding the backend in order to merge the two successfully to allow the users to partake in the functionalities.

Everyone works together but specific people are in charge of understanding the timeline of their overall tasks and figuring out where they will need assistance and then asking for help.

Project Ideas / Notes:

Design Principles

- Purposeful
  - Does it solve an actual problem, meet a real need?
- Unique
  - Is it a mere clone of an existing product?
  - Innovation:
    - UX/UI (think about what you often complain as a user) • Engineering: performance, scalability, reliability
- Real
  - Data source: web crawling, user contribution
  - Realistic

Project Description Principles:

1. Insert new records (rows) to the database and reflect the change on the frontend interface
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
3. Update records on the database and reflect the change on the frontend interface
4. Delete rows from the database
5. As a group, develop two advanced SQL queries related to the project that are different from one another. The two advance queries are expected to be part of your final application. The queries should **each** involve **at least two** of the following SQL concepts:
   - Join of multiple relations
   - Set operations
   - Aggregation via GROUP BY
   - Subqueries

Insert new records:

- store user data (games users liked and played)
- store users likes (favorite genre, game type: free or pay, etc.)
- Store user Graphic cards
- Create a “wishlist” of games they want to play

Delete Records:

- Delete games from wishlist
- Delete games that are not popular

Update:

- Create “parents” that store data about kids and restrictions on what games cannot be shown

Search Database:

- Search by date
- Search by price
- Search by sum price of game and DLC
- Search by title like
- Search by similar media title
- Search by age

*have textbox and checkboxes for input data

Creative component:

- Display a graph of genres vs payments of that genre
- Spotify wrapped like thing
- Display a graph of amount of games per genre

Tables:

GameId, all Genres

Advance Queries:

- SET OPERATIONS: example like (Action games UNION games before 2010)
- Subqueries (this one is kinda obvious)

UI:

- Homepage: top games / user data
- Search: search data from database
- Creative component: visualize data / graph

Child age,

Banned titles

banned genres

Banned price range