1. **Project Title**: GameGenius

2. **Project Summary**: GameGenius will use Steam data, user preferences, and user budget to recommend a game in their price range. The user can put in games they like or genres along with their budget and our application will generate a list of apps based on Steam data that the user might enjoy.

3. **Description**: Our application aims to help a typical gamer out on finding new and interesting video games to play by providing a platform that assists personalized game recommendations. Users can figure out their next game(s) based on their preferences, the provided reviews, and tracking their gaming interests.

4. **Creative Component**: A cool creative component that would improve the functionality of our application is implementing a like-dislike feature for games, allowing users to see how a game is perceived by the public. We could implement this by adding additional tables that are considered when recommending games and giving lower weight to games that are disliked and higher weight to games that are more liked.

5. **Usefulness**: As of 2024, there are over 5 million video games in existence, and video game advertisements are everywhere. As the years go by, gamers are more and more overwhelmed with choices about which games to buy and play.

   With our web app, we could create a service that does most of the decision-making for the user– streamlining the process of choosing what games to buy while also saving the user money. The web app will take some data about user budget and preferred games (such as specific genres), then attempt to give the user a "best choice" for what game they could buy. Users could create a profile, put in some preferences in regards to genres and price range, then get curated recommendations.

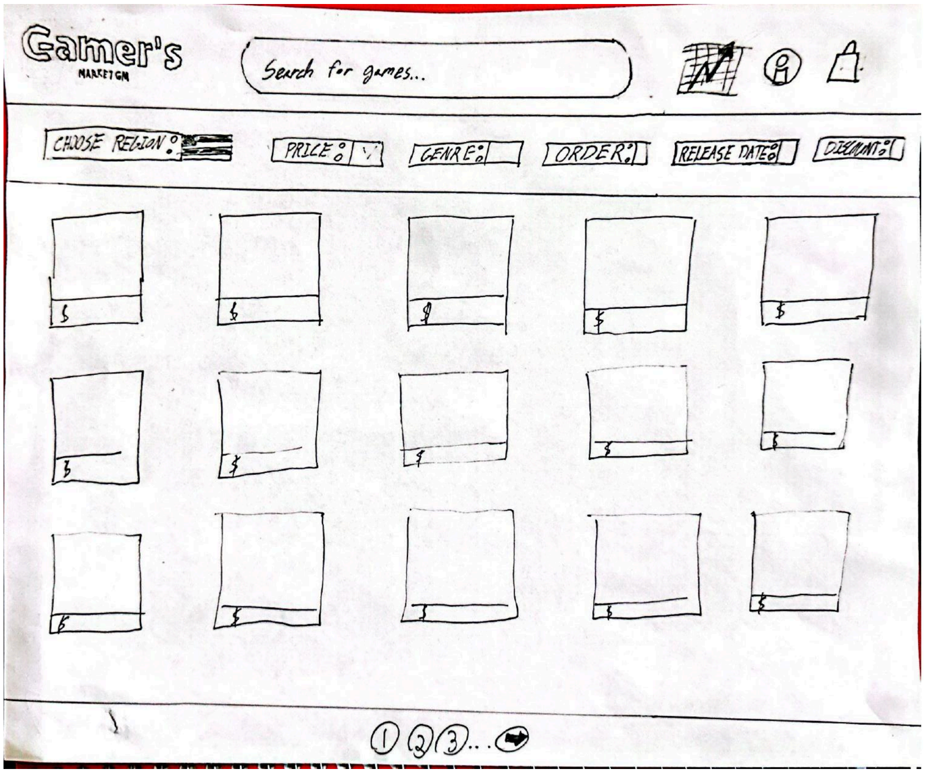
6. **Realness**: Game Genius will use a dataset to provide the recommendations for its users:
   - For the Steam Data, our dataset derives from a user named Craig Kelly that is available to the public on Github. The user made a combination of data based on Steam's public API and steamspy.com. The format is mainly in JSON and also spread out on Microsoft Excel. The size of the data is 47,504 KB in a compressed zipped file (the .csv file is 52.6 MB), it holds 13,357 records, and the information that the data source captures is things like: QueryID, ResponseID, QueryName, ResponseName, ReleaseDate, AchievementCount, RequiredAge, DLCCount, RecommendationCount, IsFree, a bunch of category options and boolean field list, etc…
   - The dataset has a bunch of genres as columns and tells which genres each game falls into. It also gives the metacritic score for each game, as well as recommendation count (how many users "upvoted" the game and recommended it), whether the game supports controller use, the price of each game, what platform it can run on, whether it is single and/or multiplayer, and some other useful information for recommendations.

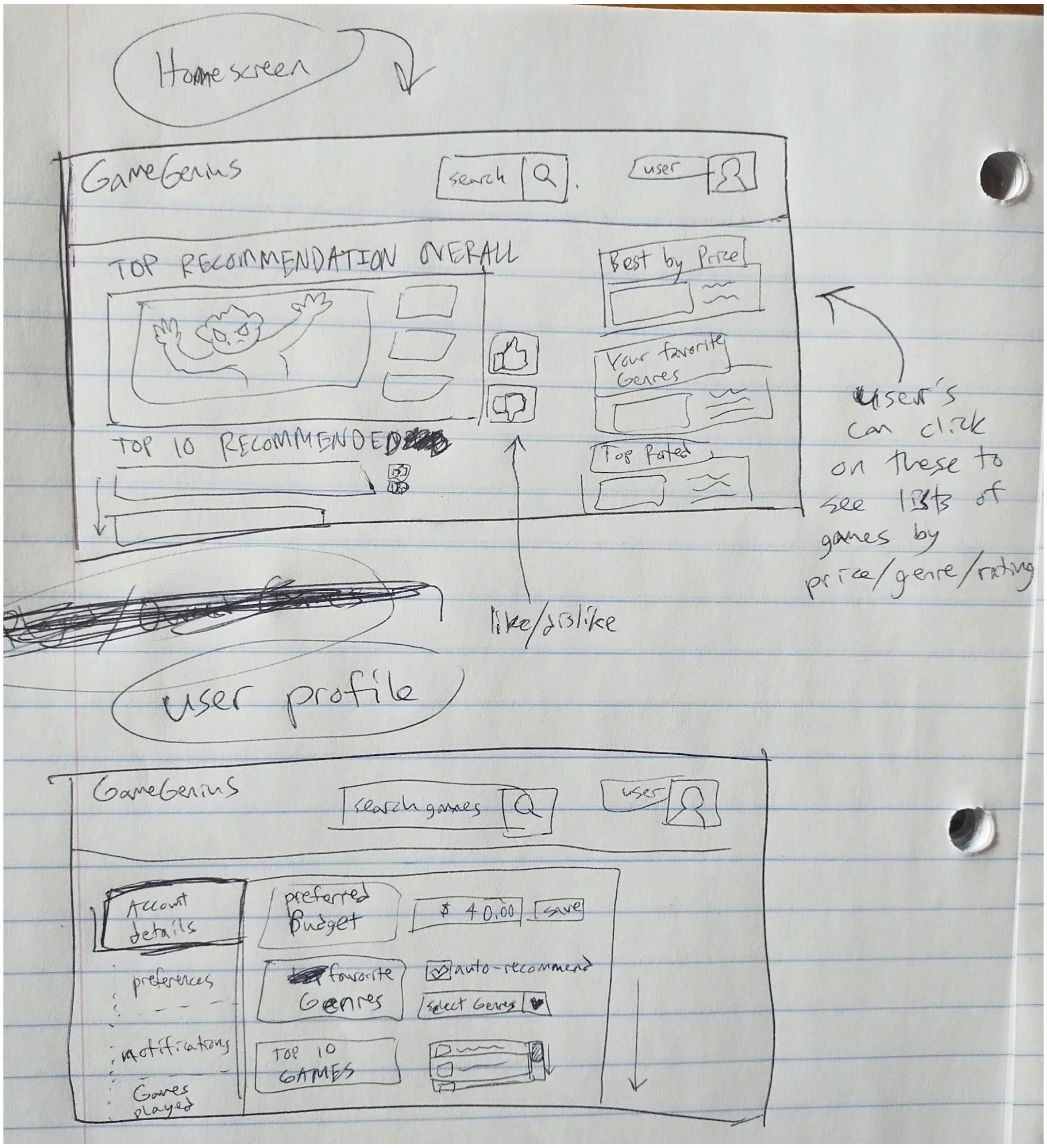
7. **Functionality:** Users can search for a game and view real-time price comparisons across different regions and stores. Can let users know when interesting games get priced dropped. Can give historical price trends and receive predictions for potential price drops. Filter and sort games by price, region, and store. We can try to implement a game recommendation system based on their Steam profile/ search interests.

Users can also input some user data on their profile, such as preferred budget, favorite genres, and favorite games.

a. UI Mockup

i.



ii. 

b. Each of the members will be responsible for parts of the backend and frontend. All members will be responsible for some of the endpoints of the API. Mike will be responsible for the DB schema design and implementing the search functionality and game recommendations. Mohammad will be responsible for integrating the dataset into the database and partly responsible for implementing functions related to user profiles. Uriel will be responsible for the DB schema as well and for creating user profiles. Jonathan will be responsible for integrating the dataset and implementing the search function and recommendation system.