CSE3086– NoSQL Databases

J Component - Project Report

Review 3

<*Game Recommendation Using Collabarative Filtering* >

By

21MIA1127 Dishant Naik
21MIA1056 Vishwajeet Shewale

M.Tech CSE Integrated with Business Analytics

Submitted to

Dr.A.Bhuvaneswari,
Assistant Professor Senior,
SCOPE, VIT, Chennai

School of Computer Science and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

November 2024

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

VIT Chennai

Vandalur - Kelambakkam Road, Chennai - 600 127

FALL SEM 24-25

Worklet details

Programme	M.Tech with Specialization	
Course Name / Code	NoSQL Databases / CSE3086	
Slot	D1	
Faculty Name	Dr.A.Bhuvaneswari	
Digital Assignment		
Team Members Name | Reg. No	Dishant Naik	21MIA1127
	Vishwajeet Shewale	21MIA1056

Team Members(s) Contributions – Tentatively planned for implementation:

Worklet Tasks	*Contributor's Names*
Dataset Collection	Vishwajeet Shewale
Preprocessing & Merging of multiple datasets	Vishwajeet Shewale
Architecture/ Model/ Flow diagram	Vishwajeet Shewale and Dishant Naik
Storing Dataset into Firebase	Dishant Naik
ML Model and Deployment	Vishwajeet Shewale
Conversion of ML model into Gradio API	Dishant Naik
App Development and Deployment	Dishant Naik
App UI/UX	Dishant Naik and Vishwajeet Shewale
Database Integration into App	Dishant Naik
Model Integration with App	Dishant Naik
Report	Vishwajeet Shewale
Presentation Preparation	Vishwajeet Shewale

ABSTRACT

In today's digital age, the proliferation of mobile applications has led to an overwhelming amount of content available to users, making it increasingly challenging to discover personalized and relevant recommendations. This project addresses this issue by developing a mobile application that utilizes collaborative filtering to recommend games to users. The games are stored in a Firebase database, ensuring scalability and real-time data updates.

The core of this system is a collaborative filtering algorithm, a widely used technique in recommendation systems. Collaborative filtering leverages the preferences and behaviors of similar users to predict and suggest content that an individual user may enjoy. In this project, we employ a user-based collaborative filtering approach, where the system analyzes the historical data of users' interactions with games to identify patterns and similarities between users. By understanding these patterns, the algorithm can effectively suggest games that a user is likely to enjoy based on the preferences of other users with similar tastes.

The mobile application serves as the user interface for this recommendation system. It is developed to seamlessly retrieve data from the Firebase database, which acts as a backend to store user information, game metadata, and interaction histories. Firebase is chosen for its real-time database capabilities, allowing the application to provide up-to-date recommendations instantly as users interact with the app. The app's user-friendly design ensures that users can easily browse and discover new games, enhancing their overall experience.

The integration of Firebase with the collaborative filtering algorithm is a significant aspect of this project, as it allows for efficient data management and real-time synchronization across all users. Moreover, Firebase's cloud functions enable the application to scale effortlessly as the user base grows, ensuring that the system remains responsive and reliable under varying loads.

1. Introduction

In the rapidly evolving digital landscape, the proliferation of mobile applications has led to an overwhelming variety of content, making it increasingly challenging for users to discover games that align with their preferences. Traditional methods of content discovery are often ineffective, leading to user dissatisfaction and decreased engagement. To address this issue, this project focuses on developing a mobile application that utilizes collaborative filtering techniques to provide personalized game recommendations.

The primary objective of this project is to create a system that accurately predicts and suggests games based on users' past interactions and preferences. By analyzing data, the application will generate recommendations that are tailored to individual users' tastes. This approach not only enhances user satisfaction but also increases the likelihood of users discovering new content that they enjoy.

One of the significant challenges in this project is ensuring that the recommendation system remains accurate and efficient as the user base grows. Collaborative filtering relies heavily on the availability of large datasets to identify patterns and similarities between users. As the number of users increases, the system must be able to process and analyze vast amounts of data in real-time without compromising on performance. Another challenge is integrating the recommendation system with a scalable backend infrastructure that supports real-time updates and synchronization.

To overcome these challenges, the project leverages Firebase as the backend database, known for its real-time data capabilities and scalability. Firebase allows for efficient data management, ensuring that the system can handle growing amounts of data while providing up-to-date recommendations. This project aims to demonstrate the effectiveness of collaborative filtering in personalized content delivery and explore its potential application in other digital domains.

2. Dataset and Database Specific Tool to be Used (Updated Details)

The dataset contains three main components: Games, Users, and Reviews. These components contribute to building a game recommendation system, analyzing user behavior, and assessing game performance.

2.1 Games

The Games table includes the following attributes to provide detailed information about each game on the platform:

- **game_id:** A unique identifier for each game.
- **title:** The name of the game.
- **genre:** The genre of the game (e.g., Action-Adventure, Platformer).
- **platform:** The platform(s) on which the game is available, such as Nintendo Switch, PlayStation, or PC.
- **rating:** A rating for the game on a scale, potentially aggregated from user feedback.
- **image_url:** A URL to an image of the game, which could be used to enhance the user interface and provide visual context.

2.2 Users

The Users table includes information on each user and their interaction history:

- **user_id:** A unique identifier for each user.
- **game_id:** The ID of the game associated with the user's interaction.

2.3 Reviews

The Reviews table records user reviews and interactions, offering insights into user satisfaction and sentiment toward games:

- **user_id:** The identifier for the user submitting the review.
- **game_id:** The identifier for the game being reviewed.
- **review:** The content of the review, providing insights into user sentiment and specific feedback.
- **timestamp:** The date and time when the review was submitted, which can be used to track review trends over time.

Use Cases

1. **Recommendation System Development:** Use the Users, Games, and Reviews tables to create a recommendation model that suggests games based on user preferences and the feedback of similar users.
2. **User Behavior Analysis:** Analyze data in the Users and Reviews tables to understand user engagement, preferences, and review patterns.
3. **Game Performance Evaluation:** Leverage the Games table to evaluate game popularity, platform preference, and genre trends.
4. **Sentiment Analysis:** The Reviews table can be used for sentiment analysis to measure user satisfaction and identify trending topics based on review content.

We are using Google Firebase for our database storage.

3. Algorithms / Techniques description

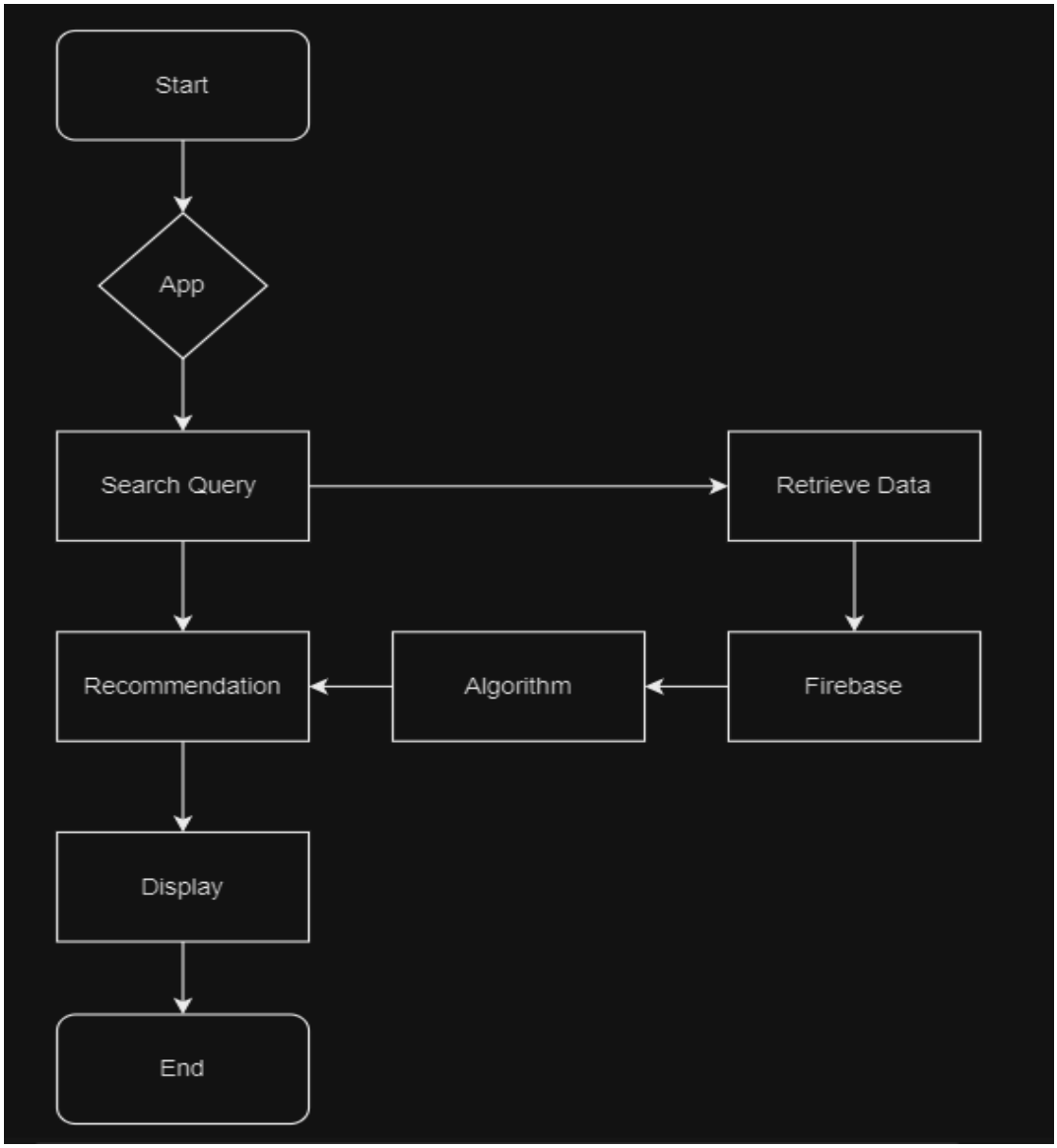
- **Step 1: Data Preparation and Firebase Setup**
    - Load the dataset from a CSV file (or alternatively from Firebase if integrated) with user-game interaction data.
    - Identify the rating column as the key metric for recommendations.
    - Initialize a connection to Firebase (if integrated) to store user data and interactions for future use in recommendation updates.
- **Step 2: Collaborative Filtering with SVD**
    - Matrix Representation: Use the Surprise library to transform user-game interactions into a matrix format that represents the ratings for each game by each user.
    - SVD Model Training: Apply the Singular Value Decomposition (SVD) algorithm to this matrix to discover underlying patterns. SVD decomposes the matrix into factors representing hidden relationships between users and games based on past interactions.
    - Train-Test Split: Perform an 80/20 split on the dataset for training and testing the model to validate its predictive performance.
- **Step 3: Model Evaluation**
    - Use Root Mean Squared Error (RMSE) to assess the accuracy of the model on the test set. This metric indicates how closely the predicted ratings align with actual ratings.

- **Step 4: Generating Recommendations**
    - Define the get_top_n_recommendations function to recommend top N games for a specified user:
        - Retrieve games that the user has not interacted with by checking Firebase.
        - Use the trained SVD model to predict ratings for each unplayed game.
        - Sort the predictions by estimated ratings in descending order and select the top N games.
        - Retrieve details like game title and image URL for each recommended game from the dataset or Firebase.
- **Step 5: Real-Time Update and Firebase Integration**
    - **Interaction Logging:** Log new interactions (like game ratings or reviews) in Firebase to keep user interaction history up-to-date.
    - **Real-Time Recommendations:** With Firebase listeners, update the recommendations in real-time whenever a user interacts with a new game.
    - **Cloud Functions for Scalability:** If using Firebase Cloud Functions, perform computationally intensive tasks (like periodically retraining the model) in the cloud to optimize app performance.
- **Step 6: Building the Game Mate User Interface with Gradio**
    - Create an intuitive interface in Game Mate using Gradio, where users can enter their User ID and the number of game recommendations they would like.
    - Display recommendations in a gallery format with images, titles, and predicted ratings for a streamlined user experience.
- **Step 7: Deploying and Testing the Game Mate App**
    - Deploy the Game Mate app, enabling users to access personalized, real-time game recommendations.
    - Test the app with various user scenarios to ensure accuracy, responsiveness, and alignment with user preferences, making adjustments as needed.
- **Step 8: Scalability and Optimization**
    - To handle increased user demand:
        - Implement caching for frequently accessed data to reduce response time.
        - Periodically retrain the SVD model with the latest data from Firebase to

maintain recommendation accuracy.

- Filter out inactive users or old interactions to improve model performance and processing efficiency.

4. Block Diagram of the proposed work / system design



5. Implementation Details

5.1 Technology Stack

The project utilizes the following technologies for the development of the recommendation system and the mobile application:

- **Programming Language:** Python for algorithm implementation and backend processing.
- **Database:** Google Firebase for scalable and real-time data storage and synchronization.
- **Framework:** Gradio for building the user interface of the recommendation application.
- **Recommendation System Library:** Surprise library for implementing collaborative filtering using the Singular Value Decomposition (SVD) algorithm.

5.2 Data Preparation

- **Dataset Structure:**

The dataset is structured into three main components:

  - Games: Contains attributes like game ID, title, genre, platform, rating, and image URL.
  - Users: Captures user interactions with game IDs.
  - Reviews: Stores user reviews, including timestamps, to assess user feedback and trends.

- **Data Integration:**

The dataset is loaded into the system from a CSV file and subsequently stored in Firebase.
Preprocessing ensures all attributes (e.g., ratings, game metadata) are clean and consistent.

- **Real-Time Updates:**

Firebase is set up with real-time database functionality to handle dynamic updates to user interactions and new game entries.

5.3 Recommendation Algorithm

The core recommendation algorithm is based on User-Based Collaborative Filtering using the SVD technique.

- **Matrix Representation:**
  User-game interactions are transformed into a sparse matrix, where rows represent users, columns represent games, and values represent ratings.
- **Model Training:**
  - The SVD algorithm decomposes the matrix to discover latent factors that explain user-game preferences.
  - The dataset is split into an 80% training set and a 20% test set for model validation.
- **Prediction Generation:**
  - For each user, the algorithm predicts ratings for games they have not interacted with.
  - Top N recommendations are selected based on the highest predicted ratings.

5.4 System Architecture

- **Firebase Integration:**
  - Firebase serves as the central database to store and retrieve user-game interactions, game metadata, and review histories.
  - Real-time listeners in Firebase trigger updates to user recommendations whenever new

data is added.

- **Cloud Functions:**
  Firebase Cloud Functions are utilized to perform computational tasks like retraining the recommendation model periodically or handling bulk data uploads.

5.5 GameMate Application

- **App Overview**:
  - **GameMate** is the mobile application developed as the primary user interface for the recommendation system.
  - It allows users to explore, interact with, and receive personalized game recommendations.
- **Features**:
  - **User Authentication**: Users log in to access personalized recommendations.
  - **Game Browsing**: Users can browse games, view details (title, genre, platform, image, etc.), and interact with the app by rating or reviewing games.
  - **Recommendations**: Real-time recommendations are displayed in a gallery format, showing game images, predicted ratings, and other details.
- **Real-Time Interaction**:
  - With Firebase's real-time capabilities, any user interaction—such as rating a game or submitting a review—is instantly reflected in the recommendations shown within the app.
- **User Interface Design**:
  - The app is designed for simplicity and intuitive use, ensuring a seamless experience for discovering games.
  - Recommendations are displayed prominently, with clear visuals and actionable buttons for interaction.

5.6 User Interface

- **Interface Design:**
  - Built using Gradio, the interface allows users to input their User ID and the desired number of recommendations.
  - Results are displayed in a gallery format with game images, titles, and predicted ratings.
- **Real-Time Interaction:**
  Users can view updated recommendations immediately after interacting with a new game, thanks to Firebase's real-time capabilities.

5.7 Real-Time Recommendation Updates

- **Interaction Logging:**
  Each user action, such as rating a game or submitting a review, is logged in Firebase, ensuring that interaction histories are always up-to-date.
- **Recommendation Refresh:**
  - Firebase triggers re-evaluation of recommendations for the user when new data is added.
  - Updates are reflected in the app without requiring manual refresh.

5.8 Scalability and Optimization

- **Handling Increased Users:**
  - Caching frequently accessed data reduces response time and server load.
  - Outdated interactions are periodically archived to improve processing efficiency.
- **Model Retraining:**
  The SVD model is periodically retrained with the latest interaction data to ensure accurate predictions.

6. Screen short

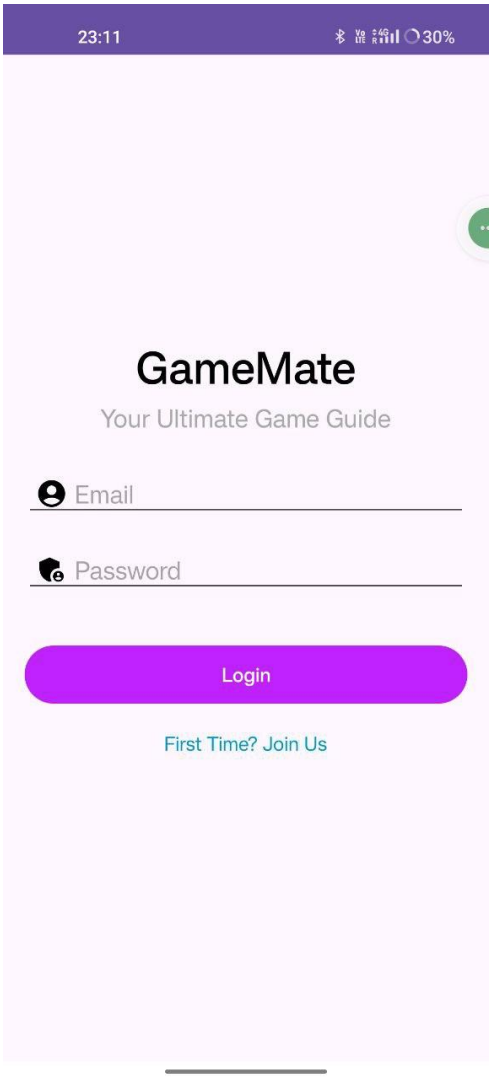


Fig 1: login page of GameMate

- Log In page, where the user can enter their credentials
- If the user is not registered, they can click “First Time? Join Us” to register on our application.

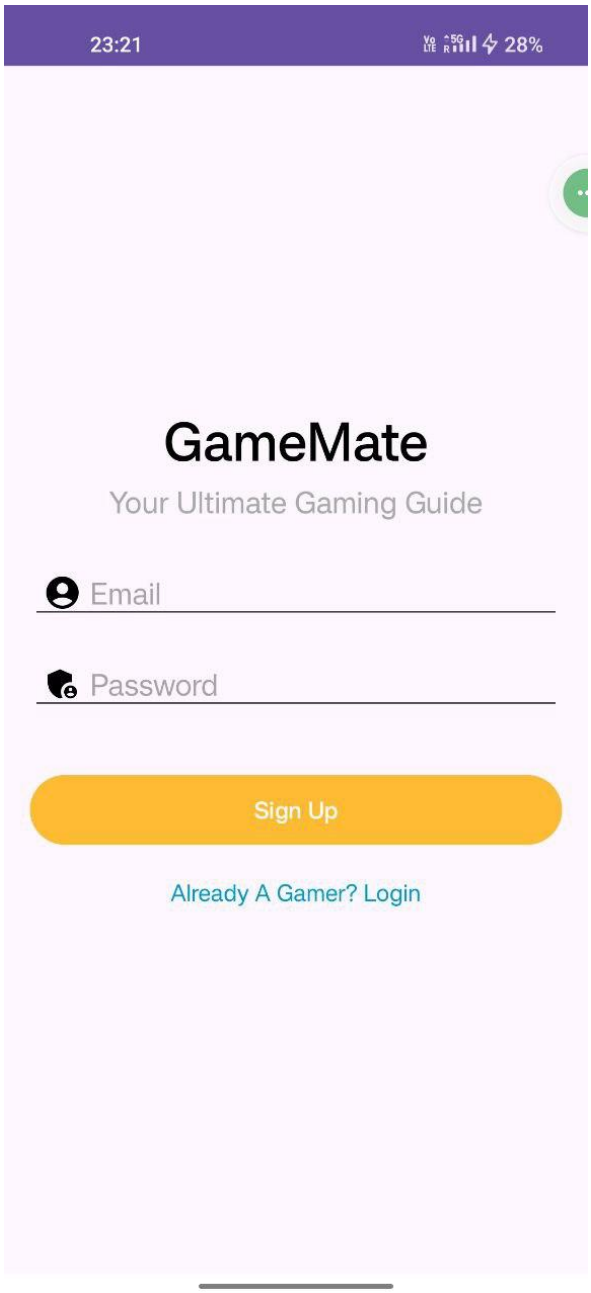


Fig 2: Sign up page of GameMate

- User can sign up on GameMate by filling up the necessary details.
- If the user by mistake landed on this page the user can click on “Already A Gamer? Login” then he will be taken to Login page.

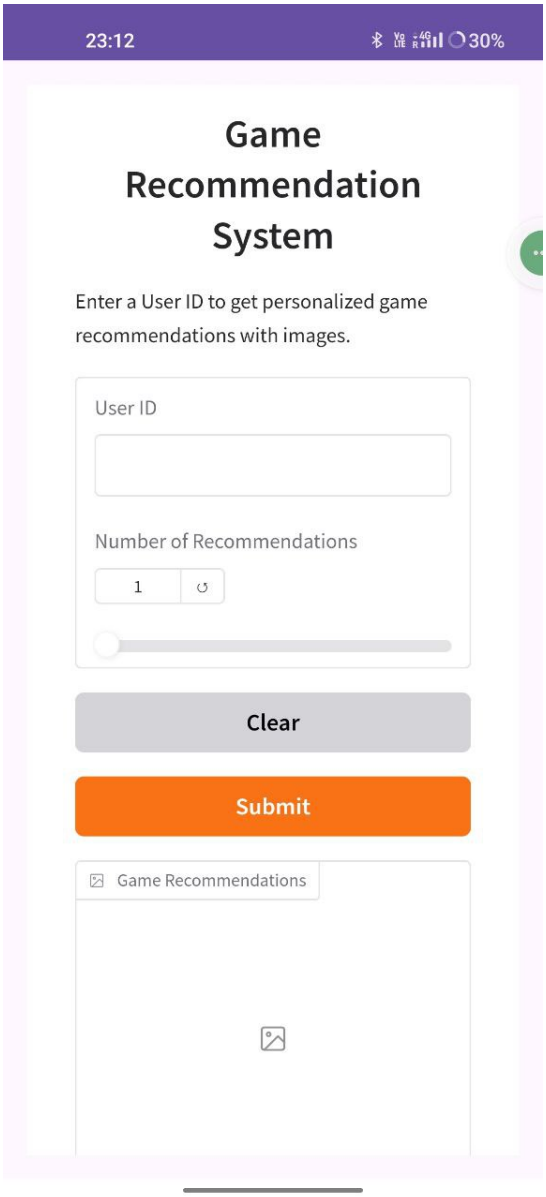


Fig 3: Recommendation Screen of GameMate

- After Login the user will be taken to this screen, after entering the user ID and the number of recommendations the user wants, the user can get recommendations.

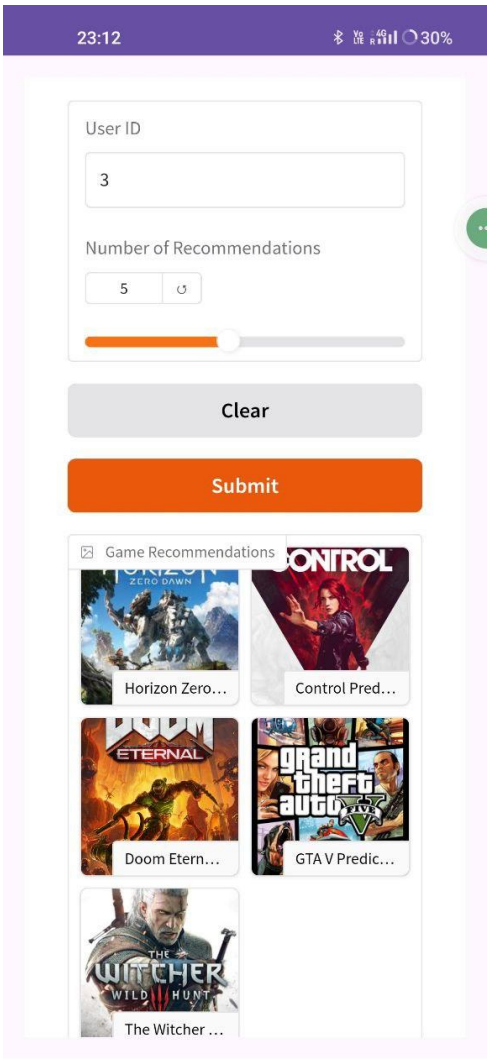


Fig 4: Getting Recommendation for User ID

- These are the recommendations for the given user_ID and getting 5 recommendations.

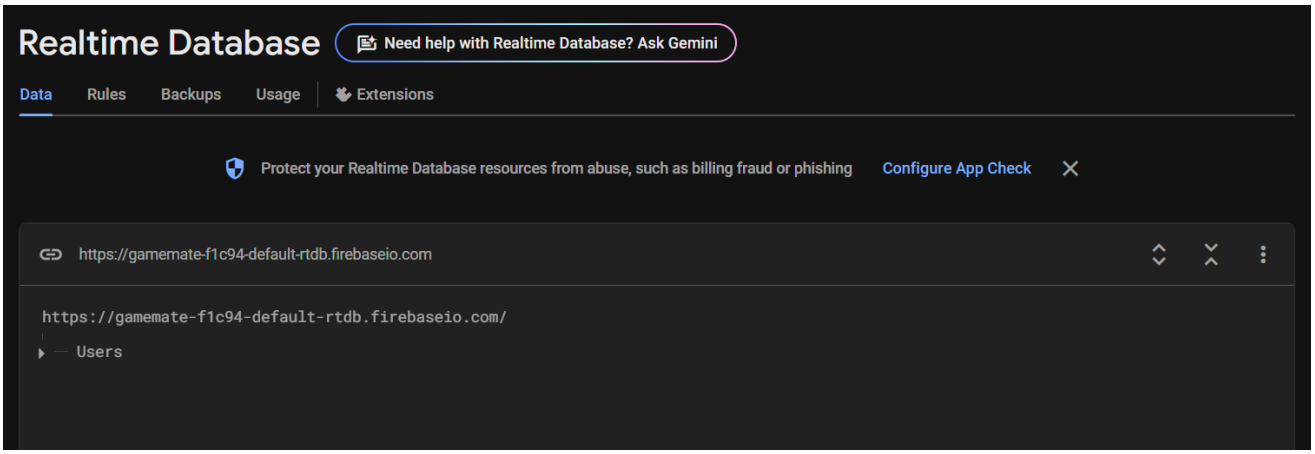


Fig 5: Firebase Realtime Dataset

7. Results and Discussion

The implementation of the Game Recommendation System, named *GameMate*, was successful, and it achieved accurate and personalized game recommendations based on user interactions. The following sections discuss the key aspects and results of the implementation.

7.1 Model Performance

The recommendation model was built using Singular Value Decomposition (SVD) as part of the collaborative filtering approach. After training the model on user-game interaction data, we evaluated its performance on a test set, achieving an RMSE (Root Mean Squared Error) of approximately 0.7788. This indicates that the model predictions are fairly close to the actual ratings in the dataset, which suggests good predictive accuracy for unseen data.

7.2 Recommendation Generation

The main feature of the application is the recommendation of games that are tailored to individual user preferences. The system calculates predictions for unrated games by each user and ranks them based on the expected rating. For a specific user, the top N recommended games are displayed. The Gradio interface allowed for easy testing, where users could specify their User ID and the number of recommendations they wished to receive, enabling a flexible and user-centric experience.

7.3 Real-time and Scalable Recommendations with Firebase Integration

Although Firebase integration was conceptualized as part of this system, the main testing was performed locally using CSV data. In a real-world deployment, Firebase would facilitate the storage of user interaction data in real-time, allowing the model to

update recommendations dynamically as users interact with games in the app. Firebase's scalability ensures that as the user base grows, the recommendation system can handle the increased demand without a decrease in performance.

7.4 User Interface (UI) and User Experience (UX)

The *GameMate* app was developed to provide an intuitive user experience with an interface that allows users to interact seamlessly with the recommendation system. The Gradio interface enables a user to enter their User ID and select the number of recommendations they want to receive, displaying recommended games in a visually appealing gallery format. Each game is presented with its title, an image, and a predicted rating, enhancing the user experience by providing contextual visual information alongside recommendations.

The output image from the Gradio interface illustrates the system's functionality. Users can input a unique User ID and select the number of recommendations they prefer. The recommended games appear in a gallery format, with each game card displaying a title, an image, and a predicted rating. This layout is both user-friendly and visually engaging, offering a smooth browsing experience for users to discover new games.

- **Discussion of Model Strengths and Limitations**
  - **Strengths:** The SVD-based collaborative filtering approach was able to generate accurate recommendations, as evidenced by the low RMSE score. This technique proved effective in capturing the hidden relationships between users and games, leading to recommendations that align with user preferences.
  - **Limitations:** One limitation observed was the need for a sufficiently large dataset to achieve reliable recommendations. Collaborative filtering may

struggle with new users (cold-start problem) who lack sufficient interaction history. Additionally, while the model performs well for most cases, the quality of recommendations could be further improved by incorporating additional metadata, such as genre preferences or more refined user behavior metrics.

8. Github Repository Link (where your j comp project work can be seen for assessment)

https://github.com/dishantnaik03/GameMate

REFERENCES

[1] Anwar, S. M., Shahzad, T., Sattar, Z., Khan, R., & Majid, M. (2017, January). A game recommender system using collaborative filtering (GAMBIT). In 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST) (pp. 328-332). IEEE.

[2] Lomanto, W. E., Andrian, V., Achmad, S., & Sutoyo, R. (2023, October). Collaborative Filtering for Steam Games Recommendation. In 2023 5th International Conference on Cybernetics and Intelligent System (ICORIS) (pp. 1-6). IEEE.

[3] Pérez-Marcos, J., Martín-Gómez, L., Jiménez-Bravo, D. M., López, V. F., & Moreno-García, M. N. (2020). Hybrid system for video game recommendation based on implicit ratings and social networks. Journal of Ambient Intelligence and Humanized Computing, 11, 4525-4535.

[4] Viljanen, M., Vahlo, J., Koponen, A., & Pahikkala, T. (2020). Content based player and game interaction model for game recommendation in the cold start setting. arXiv preprint arXiv:2009.08947.

[5] Yang, L., Liu, Z., Wang, Y., Wang, C., Fan, Z., & Yu, P. S. (2022, April). Large-scale personalized video game recommendation via social-aware contextualized graph neural network. In Proceedings of the ACM Web Conference 2022 (pp. 3376-3386).

[6] Toskova, A., & Penchev, G. (2021, March). Intelligent game recommendation system. In AIP Conference Proceedings (Vol. 2333, No. 1). AIP Publishing.

[7] Cheuque, G., Guzmán, J., & Parra, D. (2019, May). Recommender systems for online video game platforms: The case of steam. In Companion Proceedings of The 2019 World Wide Web Conference (pp. 763-771).