GameSphere: Let us unite

Team members: Bartu Sisman

Project Overview:

GameSphere is designed to treat games as more than just entertainment, viewing them as art deserving of in-depth exploration. This platform isn't just for tracking how many hours you've spent playing games across various platforms; it offers much more. On GameSphere, you can write in-depth reviews, rate games, highlight specific issues using straightforward language, share your favorite characters, and compile a wishlist of games for future exploration. All these elements contribute to your gaming profile, which your friends can follow—and you can follow theirs too. A standout feature of GameSphere is the "What to Play?" suggestion system, which recommends games based on your preferences. Essentially, GameSphere serves as your comprehensive guide for managing your gaming activities across all platforms and engaging with genuine game reviews.

Objective:

To create a centralized platform for gamers to share their experiences, integrate playtime data across platforms, and discover games tailored to their preferences.

Mobile Application (Frontend):

Features and Navigation:

1-) User Authentication Screen: Facilitates user registration and login.

2-) Game Library Screen: Displays games from linked gaming accounts.

3-) Playtime Tracker: Shows aggregated playtime data from various platforms.

4-) Game Reviews and Social Interaction: Users can write reviews, like, comment, and follow other users.

5-) Search Functionality: Enables searching for games and other users.

6-) Game Detail Screens: Provides comprehensive information about each game.

7-) Social Feed: A main page that displays the activities (comments, likes, reviews) of followed users.

8-) Profile Management: Users can view their profiles, follow others, and see their followers and who they are following.

9-) Problem Reporting: Allows users to report specific game-related issues.

10-) "What to Play?" Recommendation System: Offers personalized game suggestions.

Web API (Backend):

1. Account Linking

Link Steam Account

Method: POST

Path: /auth/steam

Description: Allows users to link their Steam account by authenticating.

Request Body: {"steamUsername": "string", "password": "string"}

Response: {"status": "success", "message": "Steam account linked successfully."}

Link PlayStation Network (PSN) Account

Method: POST

Path: /auth/psn

Description: Allows users to link their PSN account by authenticating.

Request Body: {"psnUsername": "string", "password": "string"}

Response: {"status": "success", "message": "PSN account linked successfully."}

Link Epic Games Account

Method: POST

Path: /auth/epic

Description: Allows users to link their Epic Games account by authenticating.

Request Body: {"epicGamesUsername": "string", "password": "string"}

Response: {"status": "success", "message": "Epic Games account linked successfully."}

Link Xbox Account

Method: POST

Path: /auth/xbox

Description: Allows users to link their Xbox account by authenticating.

Request Body: {"xboxUsername": "string", "password": "string"}

Response: {"status": "success", "message": "Xbox account linked successfully."}

2. Playtime Data Aggregation

Aggregate Playtime

Method: GET

Path: /playtime/aggregate

Description: Aggregates playtime data from all linked accounts and presents a unified view.

Response: {"status": "success", "data": [{"gameId": "string", "title": "string", "totalPlaytime": "number"}]}

3. Game Library Management

Get User Game Library

Method: GET

Path: /games/library

Description: Retrieves a list of games owned by the user across all linked platforms.

Response: {"status": "success", "games": [{"gameId": "string", "title": "string", "platform": "string"}]}

4. Game Details and Playtime

Get Game Details

Method: GET

Path: /games/details/{gameId}

Description: Provides detailed information about a specific game, including aggregated playtime from all platforms.

Response: {"status": "success", "game": {"gameId": "string", "title": "string", "description": "string", "totalPlaytime": "number", "platforms": ["string"]}}

5. User Reviews and Interactions

Post Game Review

Method: POST

Path: /games/{gameId}/reviews

Description: Allows users to post reviews for a game.

Request Body: {"rating": "number", "text": "string"}

Response: {"status": "success", "message": "Review posted successfully."}

Like a Review

Method: POST

Path: /reviews/{reviewId}/like

Description: Lets a user like a game review.

Response: {"status": "success", "message": "Review liked successfully."}

Comment on a Review

Method: POST

Path: /reviews/{reviewId}/comment

Description: Enables users to comment on a game review.

Request Body: {"text": "string"}

Response: {"status": "success", "message": "Comment added successfully."}

6. Game Profile Management

Get Game Profile

Method: GET

Path: /games/{gameId}/profile

Description: Retrieves the game's profile, including overall rating, all reviews, reported problems, etc.

Response:

The response will indicate success and provide a detailed profile of the game, which includes:

The game ID and title.

The overall rating of the game.

A list of reviews, each containing:

Review ID.

Username of the reviewer.

Rating given by the reviewer.

Text of the review.

Number of likes on the review.

Comments on the review, with each comment including:

Comment ID.

Username of the commenter.

Text of the comment.

A list of reported problems, with each entry detailing:

Problem ID.

Description of the problem.

7. User Profile Management

<u>Get User Profile</u>

Method: GET

Path: /users/{userId}/profile

Description: Retrieves detailed information about the user's profile.

Response: The response will indicate success and provide a detailed profile of the user, which includes:

The user ID and username.

An array of games in the user's library, each containing:

Game ID.

Title of the game.

Platform the game is on.

A list of reviews posted by the user, each review includes:

Game ID.

Title of the game reviewed.

The review text.

The rating given by the user.

Lists of users the profile owner is following and the users who are following the profile owner, each containing:

User ID.

Username.