League Companions (LoC)

Capstone One: Project Proposal

[GitHub Repo]

Goal

The primary goal of our application, League Companions (LoC), is to:

- Enhance Player Engagement:
    - Keep players engaged with the game and the community by building features that allow them to connect with friends, track their performance, and compete in friendly tournaments.
- Community Building:
    - Encourage users to connect with their in-game friends, compare statistics, and engage in friendly competition. We aim to create a space where players can interact, learn from each other, and strengthen their in-game relationships.
- Statistical Insights:
    - Provide players with in-depth statistical insights into their performance and that of their friends. It allows users to identify areas for improvement, strategize, and celebrate achievements.

User Demographic

LoC is specifically designed for League of Legends players but will cater to a diverse demographic. However, the MVP will be viewable only in English. Most of our users will fall into one of the categories below.

- League Enthusiasts:
    - These are dedicated players who invest significant time and effort into League of Legends. They are likely to be regular players, participating in ranked matches and looking to improve their gameplay.
- Friends and Teams:
    - LoC is also ideal for groups of friends or gaming teams who want to strengthen their bonds, track their collective progress, build some bragging rights, and organize friendly competitions.

API / Data

Most data will be sourced from Riot's League of Legends API. This API is very comprehensive and will allow us to access much of the data mentioned below. Other data will be capture via cookies.

- Player Data:
    - Summoner information (usernames, account IDs, levels).

        - Needed for LoC friends list and account creation. Allows us to look up specific information based on summoner information.
    - Match history (match IDs, timestamps, game modes).
        - Sourced via summoner information to gather number of games played user is filtering by
- Match Data:
    - Match details (duration, queue type).
        - Needed to determine if user is filtering by all game modes or specific game modes
    - Participant details (champion selections, statistics like kills, deaths, assists, damage).
        - Necessary for us to build a leaderboard and associate metrics with proper summoner.
    - Match events and timeline.
        - Helpful for showing other interesting stats like how many multi-kills a user has gotten over a certain number of games.
- User Preferences:
    - Customization settings
- Authentication and Authorization:
    - Table stakes needed for proper authorization (sign in)
- Tournament Data (Future Feature):
    - Tournament creation info.
    - Tournament results (scores, categories).

Project Outline/Approach

- Database Schema:
    - User Table:
        - Stores user data, like usernames, user's league account and authentication tokens.
    - Leaderboard Table:
        - Contains saved leaderboard view of previous user query. Will have details like game mode, metric being ranked by (ie: damage, kills, etc), leaderboard standings.
    - Friends Table:
        - Manages friends lists
        - Will need to find the unique id for friends accounts as their summoner name can change.
- API Considerations:
    - Potential rate limiting and API call restrictions from Riot's League of Legends API.
    - Handling exceptions and ensuring graceful degradation if API requests fail.

- Sensitive Information:
    - Securing user authentication tokens and personal data.
- Functionality:
    - User registration and authentication.
    - Connecting League of Legends account via the Riot API.
    - Managing friends lists
    - Customizable leaderboard creation to view statistics to based on specific metric to build out rankings
- User Flow:
    - Users register and authenticate.
    - Connect League account
    - Add friends to friends list
    - Customize and create leaderboards.
- Site Beyond CRUD:
    - Customizable leaderboards and tournament creation add unique interactivity.
    - JavaScript based interactions for interactivity within the leaderboards.