REDKK Project Ideas

Contributors: Reid, Hengyuan, \_\_\_\_\_\_,  \_\_\_\_\_, \_\_\_\_\_\_

* **LOL Esports Dashboard**
  + **Description:**
    - A web application serving League of Legends (A popular PC video game) esport-lovers. Providing the latest information for esport teams, matches and schedules. It supports a highly-customized data dashboard and standing visualization for users who would like to dive deep into their favorite teams and leagues.
  + **Features:**
    - **Schedule, Vods and Standings**: Our overall dashboard provides insights with the latest LOL esport statistics and news. Various filtering conditions are supported from the users’ end for customized information inquiries.
    - **Match History Tracking:** Track the historical match record between two teams for any incoming match. Giving out winning rate predictions based on the match history and contestants‘ recent form.
    - **All-in-One interface:** All related information and links are integrated together for users’ easy navigation. Beginning from a single match history, users are able to be redirected to the detailed player stats, match Vods, team info and so on. Thorough support on subpage navigations gives users the best exploration experience and lower use cost.
    - **Customized Watching List:** Pick your favorite team and players, then store them in our cloud servers. Log in with your personal account to retrieve the search history on any portable devices.
  + **APIs:**
    - [API for LOL Esport Standings](https://rapidapi.com/snldnc-kpCtDKbxo_F/api/league-of-legends-esports)
    - [API for LOL Game Info](https://rapidapi.com/snldnc-kpCtDKbxo_F/api/league-of-legends-champions/)
* **RelativityRadio/Trailblazer**: Location map of songs listened throughout city, filtered by time / song info / user
  + Description:
    - Permissions: users of this web app will link their Spotify account (OAuth) and grant location permissions (possibly using a mobile app).
    - Data Architecture: the app will automatically approximately geotag and timestamp each track as it is listened to, and upload it to the server, potentially along with (anonymous) user ID. Each tracked object will contain links to the next and last song played by that user.
      * if the start and end timestamps of a track are not almost the same as the
    - data sources will be map data from google and general listenership statistics from Spotify.
  + Features:
    - *Music Map:* visualize which songs were listened to when & where! (as well as other stuff based on [spotify data](https://research.atspotify.com/datasets/).)
      * *Track Tracker:* see where a specific song has popped up.
    - *Local Listening*: compile songs from within a space-time radius into a static playlist.
    - *Dynamic Queue*: filter, weight, order, and auto-play nearby songs!
    - *User Link*: share a playlist with someone nearby who has similar tastes
      * *Anonymous Chat:* say hi to a nearby user, only knowing their recent song history.
      * *Song Pong*: interlace queues with a nearby user!
    - Safety features:
      * Private Zone: automatically stop sharing data based on specific criteria, like being within a certain radius, or during certain times.
      * Obfuscation: all user IDs will be completely absent from the front-end, and only accessible on the backend. It will be impossible to directly query user IDs, and any grouping or association related to listening patterns will occur serverside using dynamic linking. the speed and number of linkage queries will be limited to a human rate (perhaps one every 15 minutes).
      * Location inaccuracy: geotagging will have some noise introduced purposefully—approximate distance will be given, or coordinates will be approximate (rounded to the nearest 50 feet, or something similar.)
  + APIs & frameworks:
    - [SpotifyAPI](https://developer.spotify.com/documentation/web-api/)
    - [Google Maps API](https://developers.google.com/maps/apis-by-platform)
    - [HTML5 Geolocation](https://crate.io/blog/geolocation-101-get-users-location)?
    - [Anonymous Chat](https://github.com/MRHRTZ/Anon-Chat/blob/main/README.md)?