Description:

TrashTalker is a web-based application designed to promote responsible waste disposal by educating and engaging users through a socially-driven and rewarding platform. Improper waste management is a significant environmental issue, contributing to pollution and environmental harm. This application seeks to address that problem by providing users with an application that helps to motivate and educate users, helping them be more sustainable.

Primary Features Descriptions:

Users should be able to create an account and login with a federated identity service such as Google. Their data should be stored in a secure cloud hosted database.

Users should be able to earn points by properly disposing of their trash, as well as earn points for consistent logins to the platform.

Users should be able to upload an image of an item to find out how they can properly dispose of it. This feature should also be gamified to encourage user participation.

Users should be able to view their scores and the scores of others on a public leaderboard to facilitate competition.

User interface should be designed in a modern way, and should be easy and intuitive to navigate.

Users should be able to access informational pages about waste disposal for several categories of items such as plastic, glass, metal, etc.

Requirements:

- **Functional Requirements**
    - User account creation with federated identity service
    - Point based reward system

    - Leaderboard with user scores
    - User account page displaying score etc.
    - Search bar for waste items
    - Image upload and integration with image detection model
    - Info pages for common waste categories
    - Reward redeeming system
    - Reward users for consistent logins
    - Classification Game system for image detection and point distribution
    - User info stored in cloud database

- **Non-Functional Requirements**
    - System should be responsive and fast
    - ML model should deliver classification results quickly
    - User interface should be modern and easy to navigate
    - Code should be designed in a way that is maintainable and modular
    - Database should be remotely hosted (3rd party cloud provider)