CS 160 Software Engineering Sec 4

Professor Dominic Abucejo

TrashTalker: Sprint 3 Report

Green Team (Team 4)

Apple Ko, Jaewon Kim, Katherine Yee, Riley Short, Vu Ai Van Trinh

Introduction

Waste classification is one of the most essential mechanisms needed to encourage recycling and minimize negative environmental impact caused by improper handling of hazardous waste. Despite being waste producers, most people often fail to classify their own waste before throwing it into the trash, due to difficulty differentiating types of waste or lack of incentive to comply with waste management rules. In order to help remove this barrier to effective waste management, TrashTalker is a web app designed to help users quickly classify the waste they produce by leveraging image detection technology. Our goal is for our application to encourage users to be more conscious about where their trash ends up and to help guide users to the proper disposal methods for the waste they produce.

Statistics

~How many scrum meetings occurred during sprint 3:

We had 3 meetings during Sprint 3, which lasted between 15 to 25 minutes.

~Shortest task duration time and longest task duration time:

Shortest task was cleaning up the UI and adding photos to the category pages, which took about 1 hour. The longest task was integrating the image detection model to the project and troubleshooting some errors related to setting it up on every member's devices, which took about a day.

~Completion Rate:

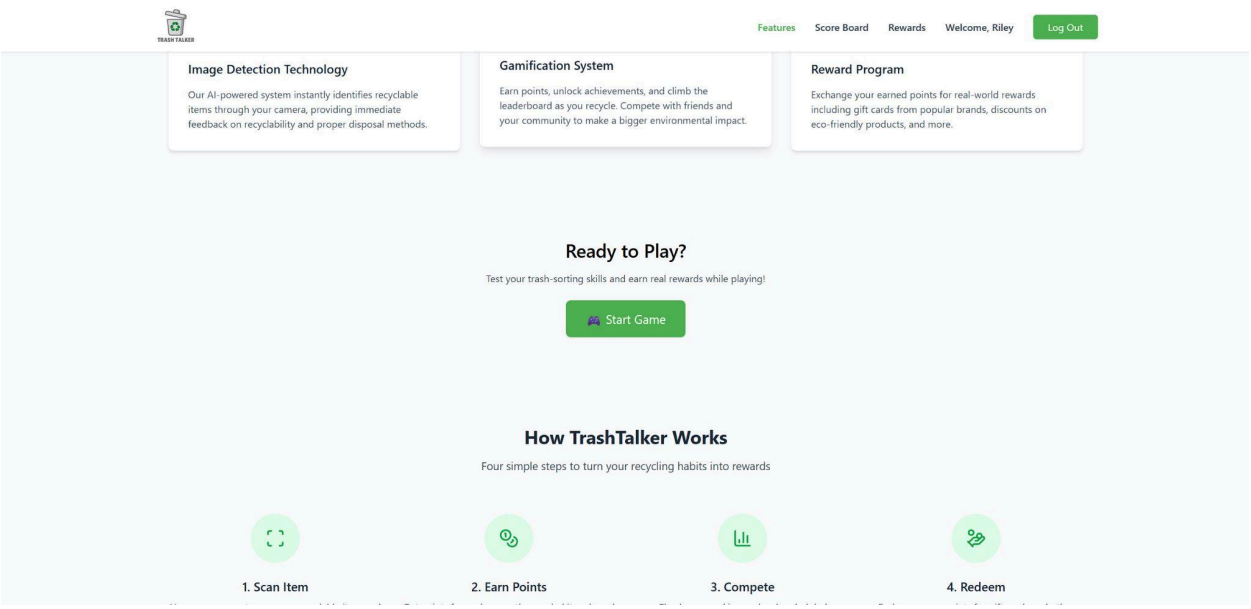
We had a 100% completion rate this sprint, as the team was able to implement all the features listed on the backlog on time.

~*Team Velocity:*

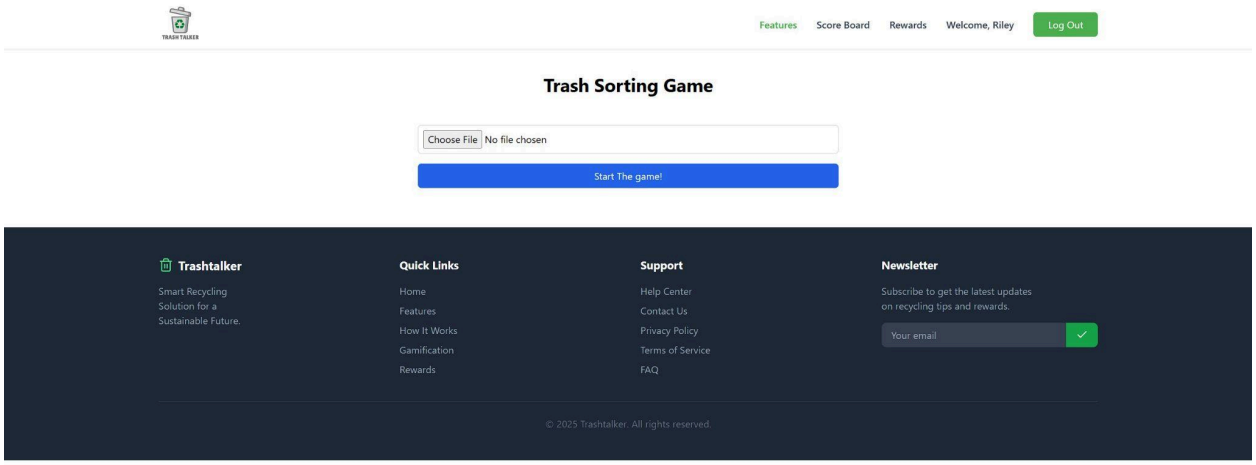
There were 10 items on the product backlog for sprint 3 and we were able to complete them in 2 weeks.

Functionality

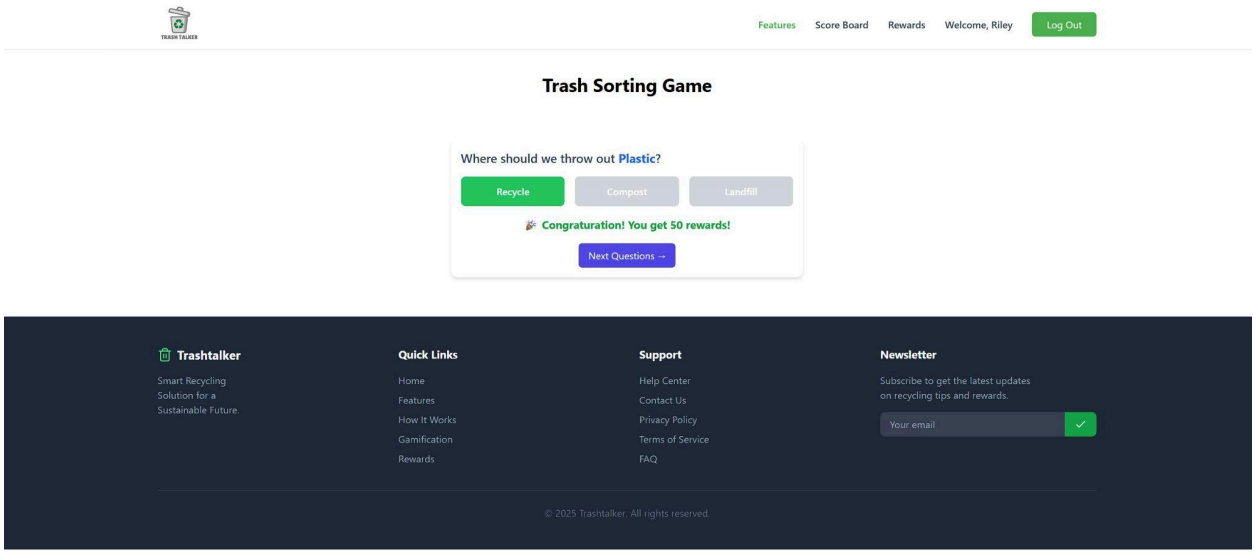
The major functionality that was added during sprint 3 was the integration of our image detection model into the UI and the backend. Users can now upload an image of the item they wish to dispose of to the webpage, which will then provide a category classification for the item. This classification is one of three categories: recycle, compost, and landfill. The feature was implemented as a mini-game, where users attempt to guess the correct classification after uploading their image. If they guess correctly, they will earn points, which will be posted to their account via a database API in the backend. Additionally, the image classification machine learning model was also integrated into the search bar feature of the application. Here, users can either type in the item they are looking for or upload the image of the item for automatic classification. In either case, the users are redirected to the appropriate page detailing disposal instructions for the item. Both of these features involved updates to the user interface as well as the backend API connections to the database. Additionally, a Flask server was added to the application to support and enable the communication between the frontend and the machine learning model running in the backend.



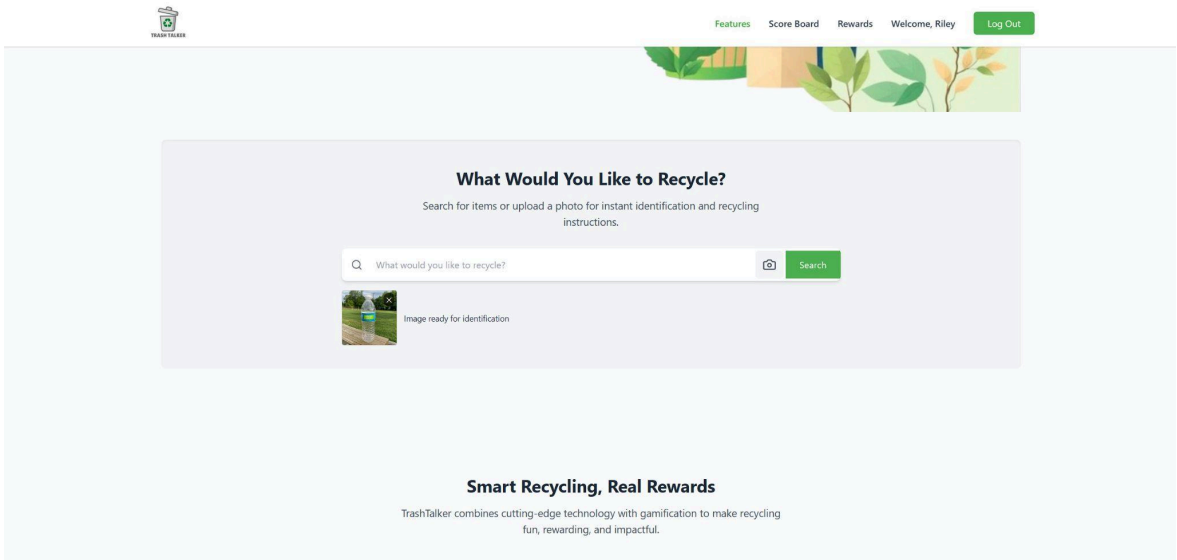
(Image shows the trash sorting game welcome screen on our home page)



(Image shows the trash sorting game image upload portal)



(Image shows results of image detection (item identified as plastic) as well as the users correct guess and points earned)



(Image shows the item search bar with AI image detection option. The results of the search with image detection navigates users to the correct informational page corresponding to their image)

Assessment of Sprint 3

Overall, sprint three went well. There was a limited set of features on the sprint backlog since this was the final sprint. This sprint broadly consisted of wrapping up the final features of the application in order to satisfy the project goals. The primary feature that was integrated was the object detection system, which is the focal point of the project. We overcame the challenge from the last sprint, which was the file size of the model being too large to store in GitHub. Our solution was to instead store the file in a shared Google Drive folder. This enabled project team members to download the necessary files onto their local Git repository. While this was not the most optimal solution, it enabled our team to finish the final feature of the application. Throughout the sprint, we maintained our strong digital communication, providing status updates to other team members and assisting each other with setup issues when new features were integrated. We held brief in-person meetings after most class sessions to provide status updates for the project and ensure all team members were on the same page about the deliverables for our final sprint. As we conclude this project and reflect on our work, we consider this project to be a success. We met all of our major goals for the application, both in terms of function and form. One key takeaway from the project is the importance of the feature planning stage. Dedicating additional time to this process in future iterations could help strengthen our development process and improve the final deliverables of the application.