

Do You Suck At Overwatch?

An application by
CJ LaFleur and Nathan Purpura

An Introduction to Overwatch

- 6 v 6 first-person shooter
- Each character has different weapons and abilities
- Teams fight to capture/complete or defend an objective.
- There is a competitive mode where players are matched against others of similar rank and skill tiers.

Competitive Ranking System

- Players play 10 placement matches.
- Players are assigned a skill rating based on their wins/losses, and performance in those games.
- Players gain or lose points based on wins and losses.



BRONZE

1 - 1499



SILVER

1500 - 1999



GOLD

2000 - 2499



PLATINUM

2500 - 2999



DIAMOND

3000 - 3499



MASTER

3500 - 3999



GRANDMASTER

4000 +

The Problems

- Parts of the Overwatch community have become toxic.
- Too many people are insulting and mean.
- People blame others wrongly for their losses.
- Players claim to be better than they are with characters, and refuse to switch or give up your main character.

The Solutions

- A way to look up users who call others out and insult people.
- Our service will show players' average stats, totals, and game highs.
- This service would include a head to head comparison feature.
- Our last feature will be a character by character comparison.

The Process

- We started out thinking about the ways we could solve the problem.
- At first our thinking was simple, but we wanted to give users more options.
- We added more search and compare options to accomplish this goal.
- We came up with more problems to solve by playing the game.

Design

- For the design, we went simple, similar to a material design.
- We intentionally avoided services like Bootstrap in order to increase our knowledge of CSS.
- We used CSS, and JavaScript, and PHP to make tables, a nav bar, a banner, etc.
- Our color scheme ended up being a soft grey background, purple nav bar, different shades of black and grey for the tables, with white text all around.

Implementation

- In order for this project to work, we had to use the LootBox API.
- This allowed us to get data about users and put them into tables.
- The tables are generated upon search, based on the type of search.
- We used JavaScript, PHP, and JSON to make this possible.

Implementation Details

- We used PHP and JavaScript to generate HTML tables which displays the requested data from the API.
- This caused our code to become modular so we didn't have to repeat HTML.

Design/Implementation Challenges

- Lack of Bootstrap.
- Lack of any outside frameworks besides Lootbox.
- Server certificate issues.
- Git.
- Time management.

Partner Roles

- Nathan:
 - Implementation of LootBox API
 - Majority of HTML, JavaScript, and PHP
 - Consulted on CSS and design
 - Consulted on write up
 - Consulted on presentation
 - Application feature design
- CJ:
 - CSS and design
 - Some HTML
 - Some PHP
 - Application feature design
 - Consulted on JavaScript and PHP