MEWS: Eric Sohel, Wilson Mach, Shinji Kusakabe, Maya Mori
SoftDev
P01: ArRESTed Development
Time Spent: 1 Hour
2022-12-05

Target Ship Date: 2022-12-23
Idea: superhero higher lower game. Use superheroAPI and others to create a higher lower game with superheroes as the subjects.

List of APIs (update with links):

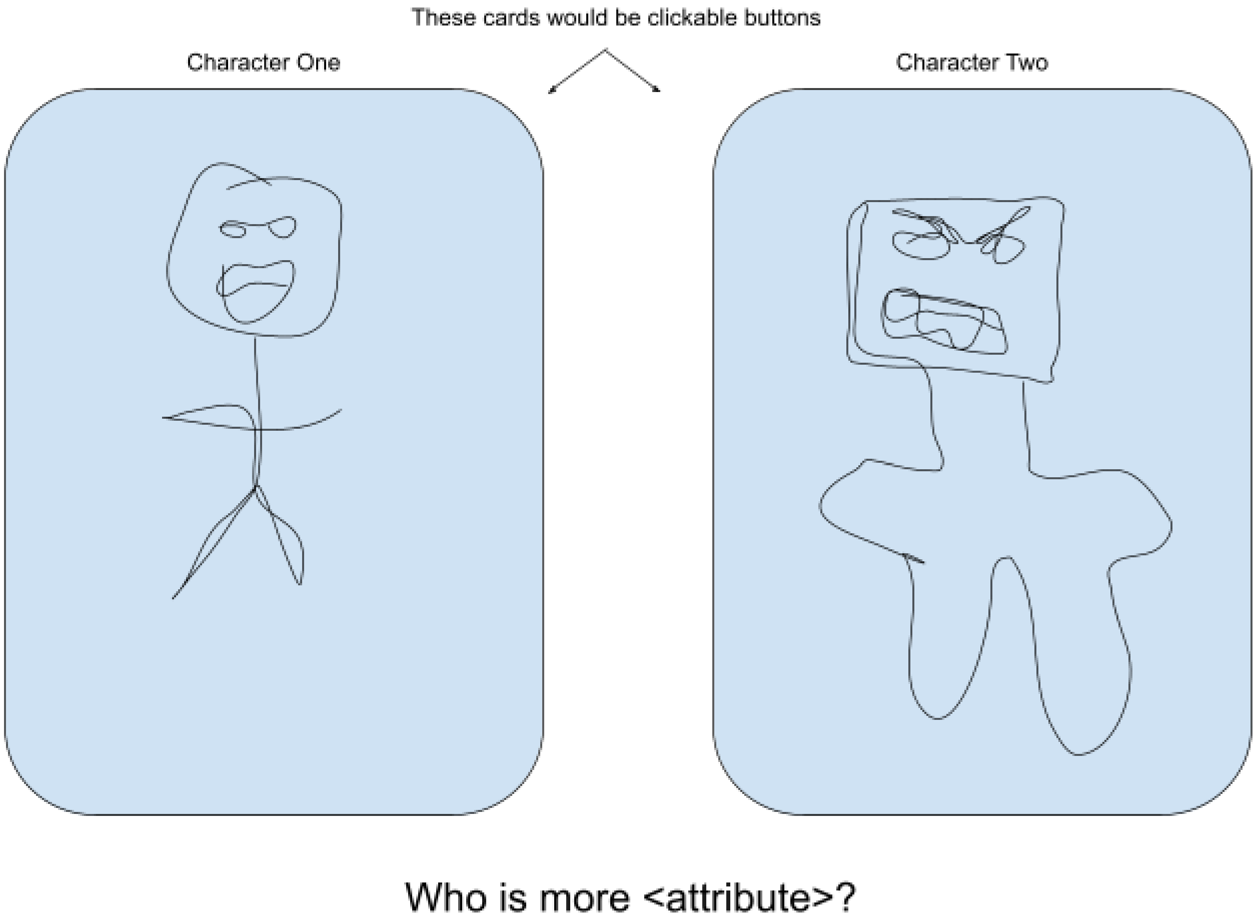
- Superhero API (get superhero stats for comparisons)
- Google analytics API
- Yes/No API - https://yesno.wtf/api
    - Use “force” query to get a GIF for yes, no and maybe
- Background API (for aesthetics!)
- Marvel comic API (gather background information about Superheroes)

Documentation:
-https://www.superheroapi.com/index.html
-

Notes:

- Players would be shown 2(+) characters and a category, such as intelligence, strength, speed, durability, power, combat and google search results and the player would try to pick which character has the higher metrics for that category. The values in these categories would be given in a number that we would compare between the different characters.
- Google analytics api can be used to see what character is trending more.
- We can use the Yes/No API to tell the users if they got the question right with a gif to make it more interesting
- Background api to make the website look better
- Can use the database to record the player’s high score and if we have time, maybe let them customize the settings that they want to play the game with.

Example of a round:



Implementation:

-For each round, the program would select 2(+) random IDs from the character IDs in the superhero API (or random ones from a list of those that we choose to use) and randomly pick a metric to compare the characters.

-We would then use the information we have on those metrics given by the superhero API for those character IDs (or search results from the google analytics API for that metric) and mark the correct one (might need either a tie as an option or do some coding to make it reroll the characters/metrics if there is a tie).

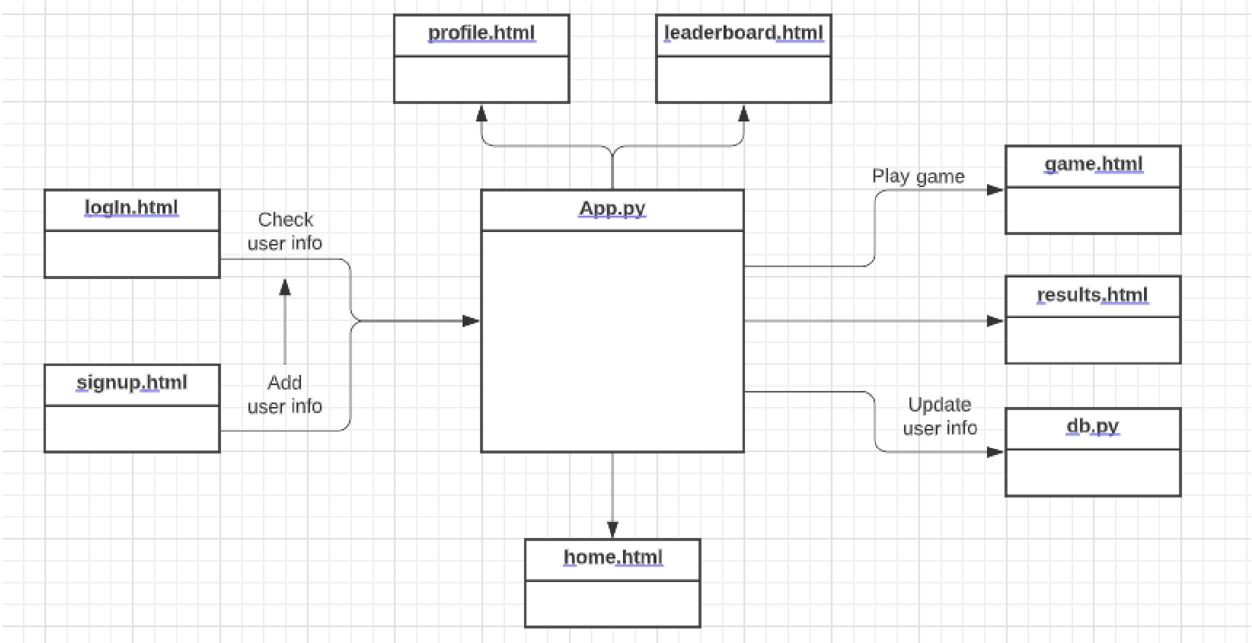
-We would render choices on our website and the player would select whatever they thought was the answer and will be told if it's correct or not. If they are correct, repeat and continue with another round. If they are wrong, it takes them to the results page where they can play again or go back to the home page/their profile.

-Scores would be recorded and stored in the database and there could be a leaderboard page to see the best scores/players.

-Profile page could show some of the player's stats.

Frontend Framework: Bootstrap
Reason: While Foundation is focused on the visual and customizability aspects of a project, Bootstrap is primarily focused on functionality and not so much in the way the final project will look. For the purposes of this project, we are more concerned with the functionality that is offered by bootstrap rather than the aesthetics offered by foundation.

Concept Map:



Data Base:

username	password	high_score
Each user will be able to create their own account!	Their progress/ scores will save!	Other users will be able to compete for the highest score!

Python Files:

- data.py - deal with the database
- __init__.py - run Flask app

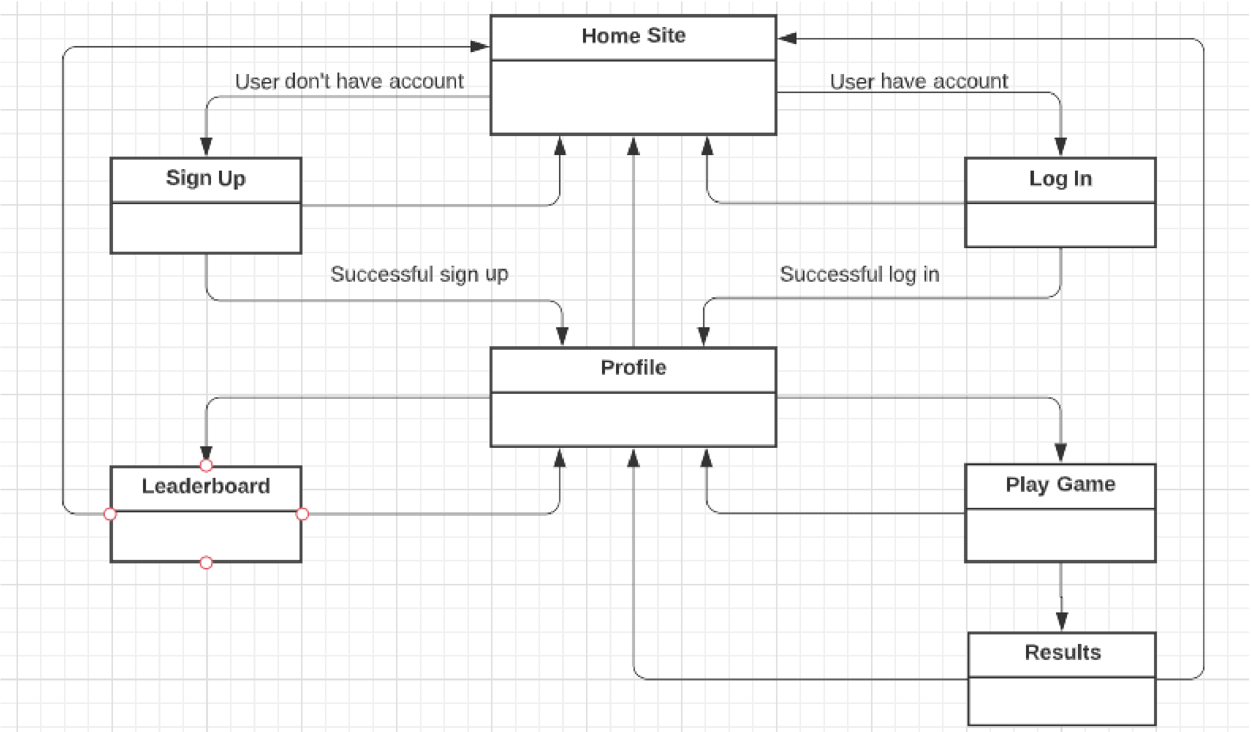
Html Files (templates):

- Home.html
- Signup.html
- Login.html
- Profile.html
- Leaderboard.html
- Playgame.html
- Results.html

CSS File:

- Style.css

Site Map:



Roles:

Databases - Maya
Front End- Eric/Shinji
Flask- Wilson/Shiji/ Maya