GRINDR: Brian Wang, Donald Bi, Jian Hong Li, Brian Yang
SoftDev
P01: ArRESTed Development
2022-12-02

Target Ship Date: {2022-12-19}

Propo:
Our website will be a website that utilizes information from various APIs about the user to determine how grass-deprived the user is. The information will all be willingly given by the user.

Database (*SQlite3):*
db.py

- Login information

ID	Username	Password	Did_Questions

- Insult database

Insult_ID	Insult_Text	Grass_Level	API_Info (if relevant)

- Grass meter

ID	Quiz Grass (need to store grass earned from quizzes separately)	Grass

- Game accounts (for profile page)

ID	Game	Game_Username

Front End:
FEF: Bootstrap
style.css
- Styling Sheet

index.html
- Grass Profile title
- Login form

questionnaire.html
- Contains a list of basic questions such as how many sports do you play, how much time you spend outside, how much time you play games, etc… and calculates initial grass score
- This only happens once for each user before they enter the real website, checks by using the Did_Questions value from the login information table

register.html

- Registration form

profile.html

- Grass o'meter at the top telling you how much grass you touch (usually negative)
    - Calculated based on the usernames of different games that you put by getting stats in the api
- Insult box at the top that has different levels of insults based on your grass level, insults are taken from insult database
- Places to input usernames of different games
  <u>After you have already inputted username of game</u>
- Containers of profiles of different games detailing stats
    - Ie. you are level 420 in League of Legends then you grass meter will be -4200

pokequiz.html

- Uses pokeapi to get a random pokemon and display its sprite on the site
- There's an input to submit the pokemon's name
    - If the answer is correct, then it -10 grass on your grass meter and goes to profile
    - If the answer is wrong, then it +1 grass on your grass meter and goes to profile
- There are also extra inputs to submit the pokemon's typings that decrease grass further
- Could also include extra inputs such as pokedex id and stuff, but thats if we have time

aniquiz.html

- Basically the same as pokequiz.html but for anime and without pokemon typings

Back End (*Flask):*
__init__.py

- Main web server file: provides navigation between few sites

API keys:

- Provides python backend with access to assorted APIs

Python:
grass_calc.py

- Calculates the grass meter based on user inputs and makes the necessary changes to database
- Accesses assorted API to acquire grass information

db.py

- Contains functions to insert new data into the database, ie. when logging in
- Contains functions to retrieve data from the database, ie. getting how much grass a user has

api.py

- Contains functions to make getting information from apis easier

APIS
pokeapi

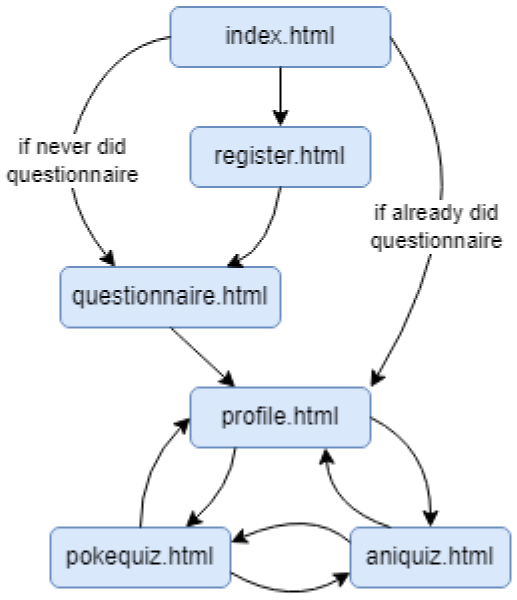
- For getting pokemon information for the pokequiz myanimelist api

- For getting anime information for the aniquiz

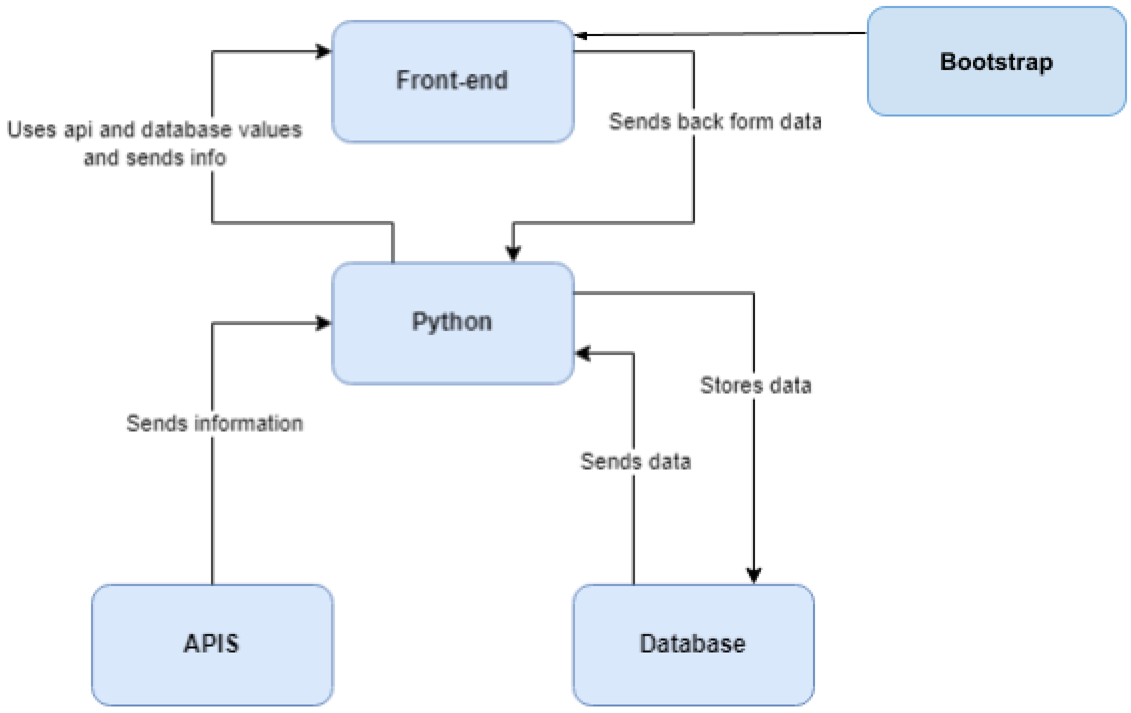
Hypixel, RIOT games, other game apis

- For making profile information and calculating the grass you touch based on it

Site Map:



Concept Map:



Task Breakdown:

Front End**:** (Donald)

1) Create Login and Register pages
2) Create questionnaire page
3) Create profile page
4) Create pokequiz and aniquiz.html
5) Style the stuffs

Database Management: (Jian Hong)

1) Create db.py to store account information
2) Create functions to set up database file
3) Create functions to set up tables required
4) Create functions needed to add new information and retrieve old information

Backend: (Brian, Brian)

1) Make init and grass_calc python files
2) Route everything in init according to the sitemap
3) Finish backend for login and register
4) Create functions to get data from APIs in api.py
5) Create functions for grass calculations in grass_calc
6) Use the functions created in 4) and 5) to send the data needed to the frontend