Talia Fight Club (Sponsored by Yusha)

: Yusha Aziz, Talia Hsia, Anthony Sun, William Vongphanith

P01: INSERT

Target ship date: {2022-12-23}

Program Components:

- Flask app (__init.__py)
    - User input is stored from website and served to html pages for functionality.
- HTML templates
    - Login/Signup
        - login.html (landing) – user is presented with a home page with login form (asking for username and password) and signup page redirect button.
            - Login form uses DB information to verify user input and login message displayed after form.
            - Bad password/username will be displayed in the login message
        - signup.html – form presented to user for entering username, password, and age. Signup message is displayed below form.
            - Signup form uses DB to make sure no pre-existing username.
            - If the username already exists then (bad username) returned to the signup message.
    - Landing page/Main site
        - landing.html – main screen where user can select which of the 4 databases (anime, league, valorant, pokemon) they want to include in their compatibility test
            - Displays past characters that they've matched with
            - Form with the areas they want to include and submit button directs user to test.html
        - test.html – user presented with a form consisting of (x) multiple choice questions which upon filling out the user will be able to submit
            - Questions will be randomly selected from a DB of questions
            - User cannot submit the form until all questions have been filled out in the form
            - Input answers will add points to personality categories which are added to the DB → characters are selected from category with highest points
        - result.html – result character is printed with message and generates image and information on character
            - Uses image apis to gain image

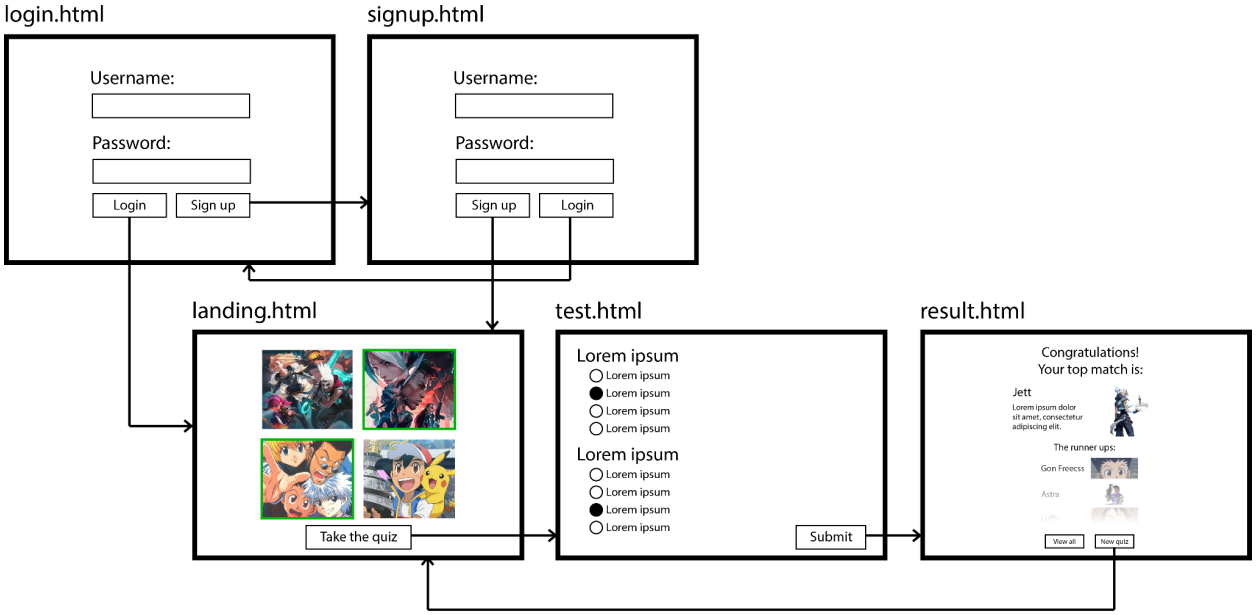
        - Result stored in user DB
        - Button to take test again with same categories → test.html
        - Button to return to landing.html
- Js (/js)
    - Includes necessary javascript to help CCS files
- CSS files (/static)
    - Personal edits to bootstrap front-end
- API Keys (txt file)
    - txt files will all necessary API Keys (see APIs)

Database Organization:

Table: Users	
id	int
username	string
password	string (are we hashing?)
previous_characters	string (separated by commas)
qualities	string (JSON)

Table: characters	
id	int
name	string
category	string
qualities	String (JSON)

Site Map:



APIs:

- Kitsu
  - https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_Kitsu.md
  - We will use this API to find anime characters that match the user's preference.
- PokeApi
  - https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_pokeapi.md
  - We can use this API to find pokemon that have characteristics that the user prefers, for example color, native home region(hot, cold, ocean, city, etc..), body shape, pokemon type(fire, water, dragon, fairy …)
  - Our preference quiz will allow us to find the information of the color, home region, body type, and type of the perfect match. we will then search the database of pokemon in the API for the pokemon that meets all of the criteria.
- RiotAPI
  - https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_riotAPI.md
  - We can use this API to gain information about league of legends and valorant characters in order to match them with the user's preferences.
- LoveCalculator API
  - https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_LoveCalculator.md

    - After we have a list of potential matches from the above 3 APIs, we will run them all through the LoveCalculator API and find the highest percentage of compatibility.

Front-end:

- Bootstrap Front-end
    - We are using bootstrap because of better handling ability and 2 of our group members have experience with utilizing the framework.
    - Components
        - Flex properties
        - Grid layout
            - Input forms are arranged in grid boxes
        - Cards
            - Past character compatibility results
        - Navigation bar
            - Logout ability and home

Tasks:

- Yusha:
    - Character categorization, backend
- William:
    - API response parsing, backend
- Talia:
    - Front-end / html
- Anthony:
    - Database, cleanup