Team Discarded Reindeer - Karen Chen, David Doktorman, Sasha Fomina, Angelica Zverovich

<u>Summary:</u> Our project is a dating website for computer scientists that learns more about 'your type' the more you use it and gives you more compatible matches

Program Components:

APIs:

-Face++ : facial recognition service

-IBM Watson Personality Insights: analyzes texts and returns personality vars (emotional range, conscientiousness, openness, introversion/extraversion, agreeableness) , needs (structure, curiosity challenge, ideal, stability), values (stimulation, tradition, helping others, taking pleasure in life, achievement)

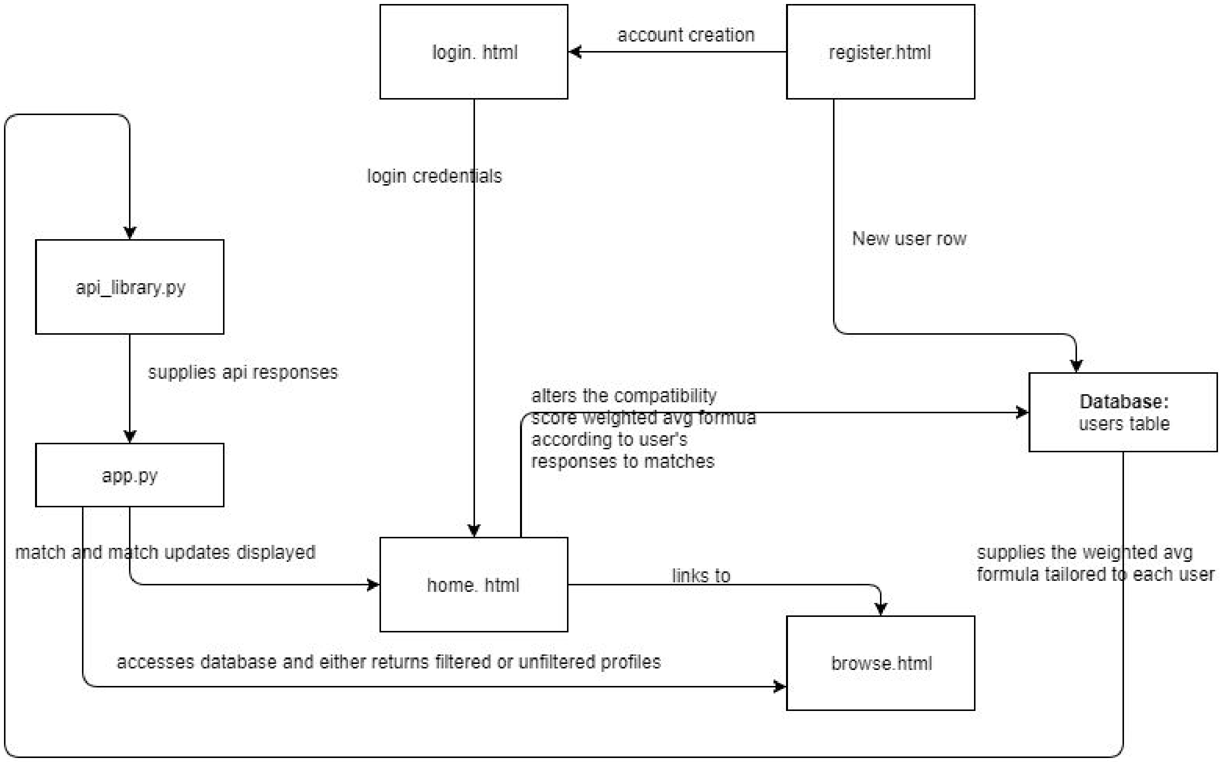
-Google maps api (**if time allows)**

- login.html
    - Simple login, using hashed password storing
- register.html
    - Enter username and password and add to your profile:
        - Write a personal bio, answer multiple choice questions pertaining to computer science to create a personality profile and user inputs their gender and sexual orientation
        - Who is your celebrity crush?
- home.html
    - Search bar at the top to search for profiles
    - Button to browse all uses to manually choose a match
    - Displays your profile information with the option to edit any of your info
    - Displays your current match, with their picture and a link to their profile
        - Asks user if they would like to be matched with this person or not

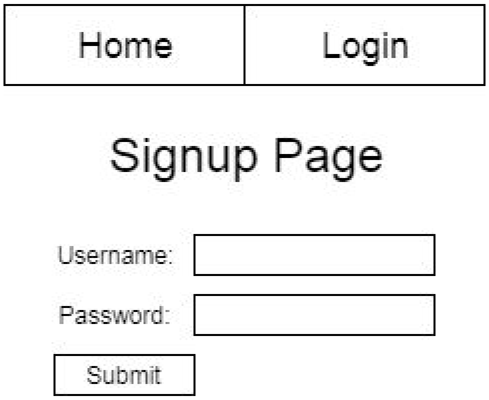
      - A match is chosen based on certain requirements such as the user’s gender preference (and, if we have time, proximity to user) and because this match had compatibility score higher than <some baseline number yet to be determined> based on personality insights and similarity appearance-wise to the user’s celebrity crush
      - Compatibility score is a weighted average of all the factors that go into consideration, initially all factors receive an equal weight. If a user rejects a match than the category which received the lowest score in the rejected match will now be weight a certain number of percentage points higher
    - If user say yes to a match, it shows your next best match based on previously used compatibility score average without any recalculations
  - User will receive a notification on their timeline in there have been any successful matches (both people said yes) with the match’s phone number
- browse.html
  - Either displays profiles from search results, or unfiltered all profiles
- bootstrap
  - Css framework
- style.css
  - Our custom style for the website
- app.py
  - Flask App
  - Includes all routes
- Utils
  - __init__.py
  - db_library.py
    - Handles database stuff
  - api_library.py

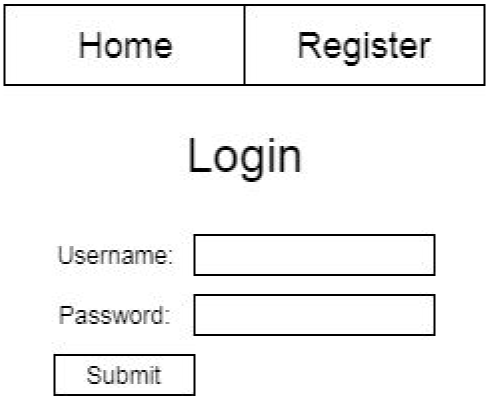
        - Sends api requests and parses responses
        - Will have function that returns the current match
- love.db:
    - users Table

Component Map:

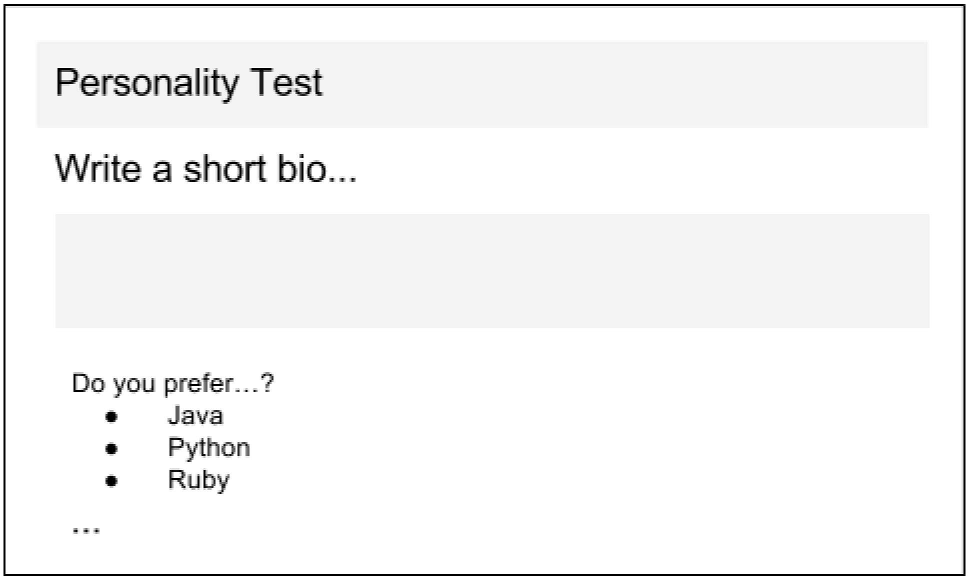


Site Map:





Account Setup/Signup:



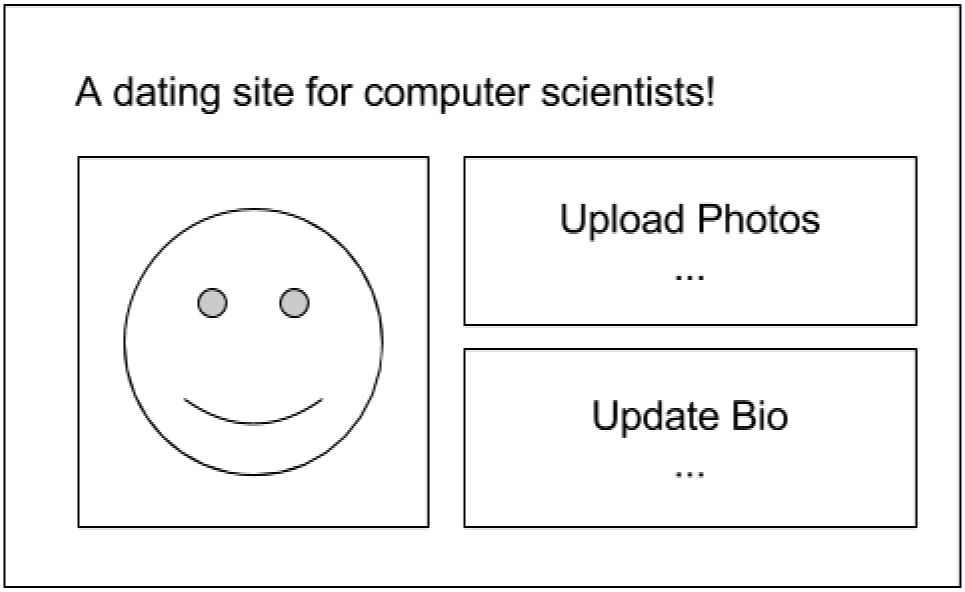
Personality Test

Write a short bio...

Do you prefer…?

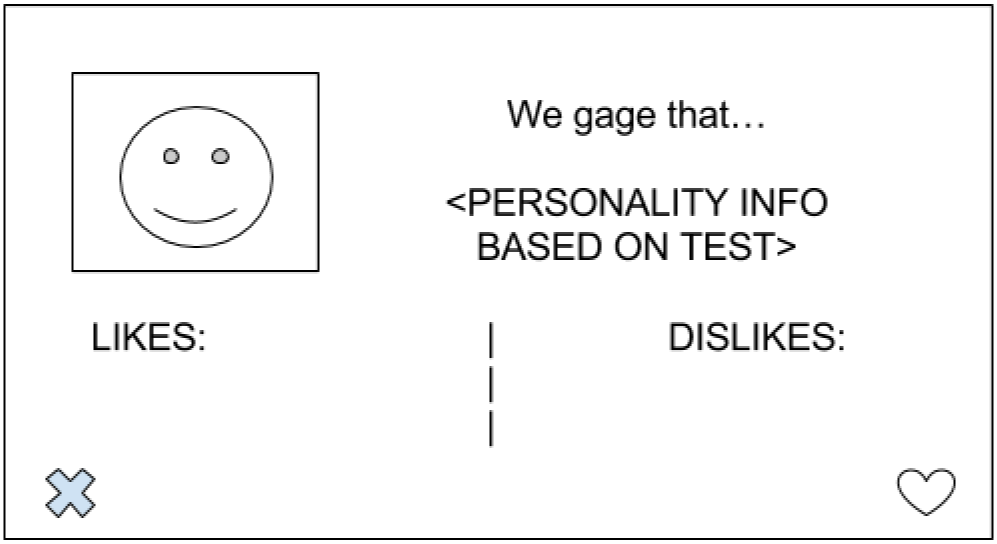
- Java
- Python
- Ruby

…



A dating site for computer scientists!

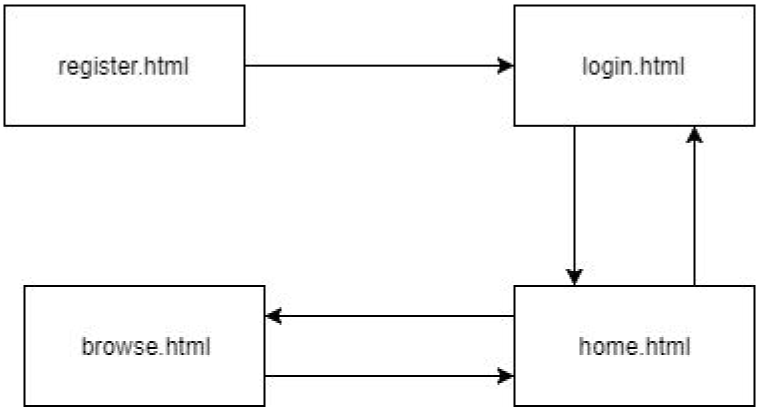
Home Page:



We gage that…

<PERSONALITY INFO BASED ON TEST>

Navigation:



Database Schema: love.db

- Users table
  - Username|Password| bio | all your multiple choice answers for your profile| current match| current match score| weight of attribute 1(in compatibility score formula) | weight of attribute 2| weight of attribute 3 …

Task Breakdown:

- Karen
  - Bootstrap, css, HTML, javascript

- David
  - Database

- Sasha
  - Project manager. Keeps track of devlog, oversees distribution of labor, keeps project on schedule, handles minor coding tasks

- Angelica

- flask routing
- API parsing