

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

Summary: 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:

-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)

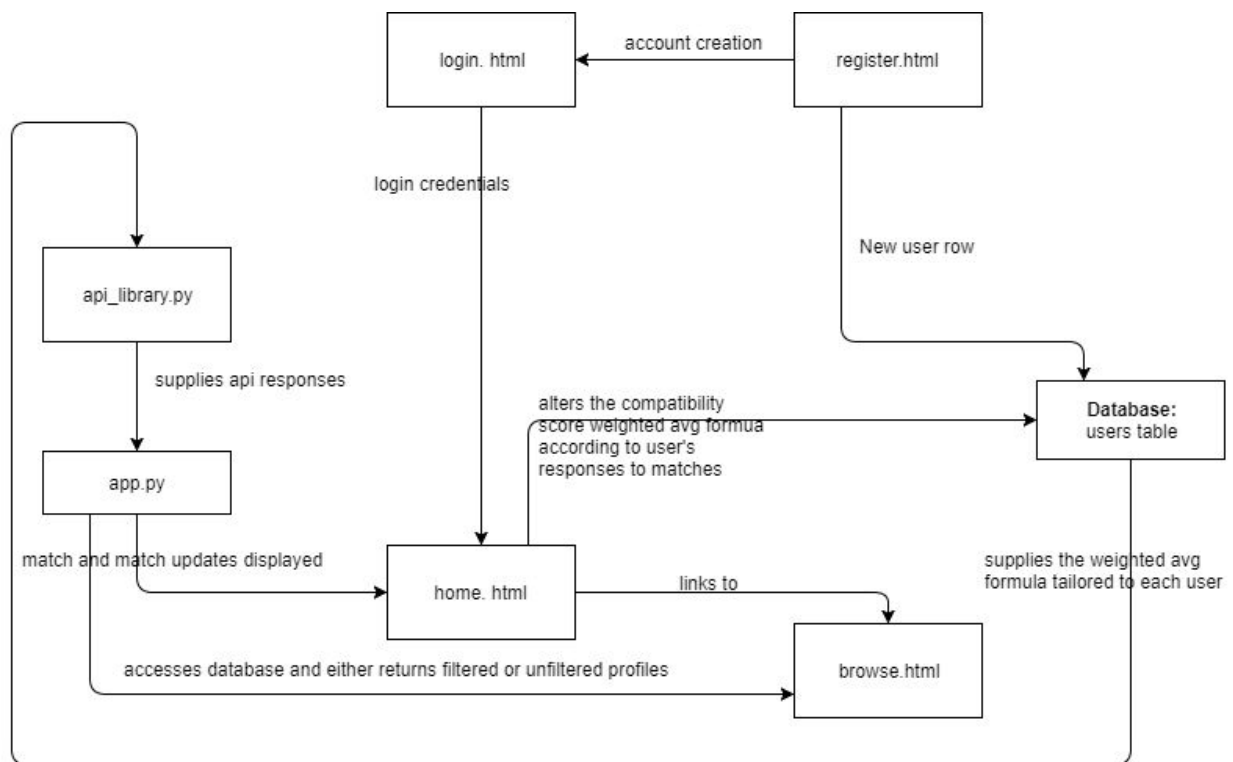
- login.html
 - Simple login, using hashed password storing
- register.html
 - Enter username and password and add to your profile:
 - Write a personal bio, answer multiple choice questions pertaining to computer science to create a personality profile and user inputs their gender and sexual orientation
 - 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 then one of two things can happen:
 - If their percentage of incompatibility in the category in which they differ the most is greater than 50%, we assume that the user wants to be matched with people more similar to them. Because of this, we increase the weight of that
 - the category which received the lowest score in the rejected match will now be weighted a certain number of percentage points higher
 - If user says 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 if there have been any successful matches (both people said yes) with the match's phone number
- bootstrap
 - CSS framework
- style.css
 - Our custom style for the website
- login.py
 - Flask App
 - Includes all routes
- Utils
 - `__init__.py`
 - `dbLibrary.py`
 - Handles database stuff
 - `api_library.py`

- Sends api requests and parses responses
- Will have function that returns the current match
- dating.db:
 - Users Table
 - Personality Table
 - Formula table

Component Map:



Site Map:

Home	Login
------	-------

Signup Page

Username:

Password:

Home	Register
------	----------

Login

Username:

Password:

Account Setup/Signup:

Personality Test

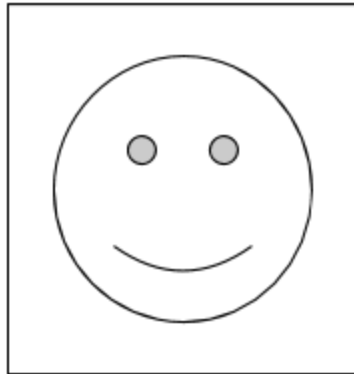
Write a short bio...

Do you prefer...?

- Java
- Python
- Ruby

...

A dating site for computer scientists!



Upload Photos

...

Update Bio

...

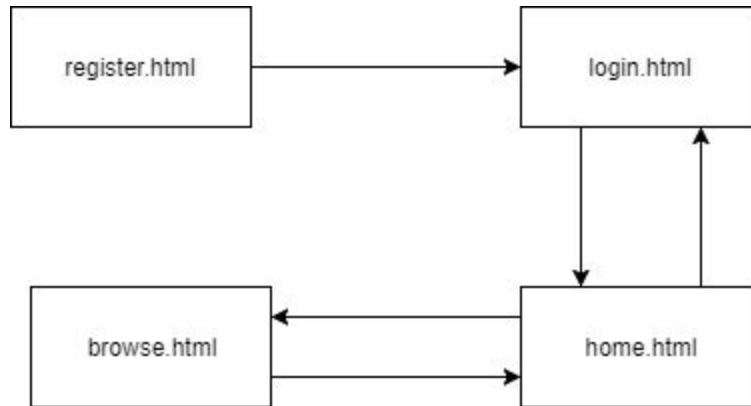
welcome.html

Suggested Match:
Picture
Name
% Similarity
Bio

Dislike

Like

Navigation:



Database Schema: love.db

- Users table

```
create_users = "CREATE TABLE users (username TEXT PRIMARY KEY,  
password TEXT NOT NULL,phone TEXT, name TEXT NOT NULL, age  
INTEGER NOT NULL,gender TEXT NOT NULL, prefGender TEXT NOT NULL,  
lang TEXT NOT NULL, sortAlg TEXT NOT NULL, type TEXT NOT NULL,  
bitcoin TEXT NOT NULL, nameCase TEXT NOT NULL, braces TEXT NOT  
NULL, bio TEXT NOT NULL, img_name TEXT NOT NULL, posMatch TEXT,  
percent INTEGER, od DECIMAL, cd DECIMAL, ed DECIMAL, ad DECIMAL,  
emd DECIMAL, challd DECIMAL, curd DECIMAL, exd DECIMAL, hd  
DECIMAL, ideald DECIMAL, libd DECIMAL, lod DECIMAL, pd DECIMAL,  
exprd DECIMAL, stabd DECIMAL, strucd DECIMAL, csPercent DECIMAL,  
suggested TEXT, queue TEXT, liked TEXT, secured TEXT);"
```

- **Personality**

```
create_personality = "CREATE TABLE personality (username TEXT  
PRIMARY KEY, open DECIMAL NOT NULL, consc DECIMAL, extra  
DECIMAL, agree DECIMAL, emotRange DECIMAL, challenge DECIMAL,  
closeness DECIMAL, curiosity DECIMAL, excitement DECIMAL,  
harmony DECIMAL, ideal DECIMAL, liberty DECIMAL, love DECIMAL,  
practicality DECIMAL, expression DECIMAL, stability DECIMAL,  
structure DECIMAL);"
```

- **Formula**

```
#table of coefficients  
create_formula = "CREATE TABLE formula (username TEXT PRIMARY  
KEY, openCo DECIMAL, conscCo DECIMAL, extraCo DECIMAL, agreeCo  
DECIMAL, emotRangeCo DECIMAL, challengeCo DECIMAL, curiosityCo  
DECIMAL, excitementCo DECIMAL, harmonyCo DECIMAL, idealCo  
DECIMAL, libertyCo DECIMAL, loveCo DECIMAL, practicalityCo  
DECIMAL, expressionCo DECIMAL, stabilityCo DECIMAL, structureCo  
DECIMAL, csCo DECIMAL) "
```

Task Breakdown:

- Karen
 - Bootstrap, css, HTML, javascript
- David
 - Image processing
- Sasha

- Project manager. Keeps track of devlog, oversees distribution of labor, keeps project on schedule
 - Database
 - Api stuff
- Angelica
 - flask routing
 - javascript