

Garlic Fries: Samantha Hua, Diana Akhmedova, Vivian Teo, Gitae Park

SoftDev

P01 -- ArRESTed Development -- NBA Love Story

2022-12-02

time spent: 1.5 hrs

Target Ship Date: 2022-12-14

Files Needed:

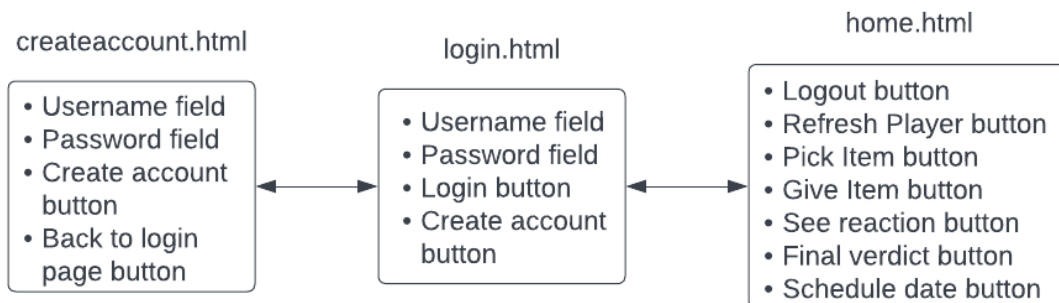
- HTML Files:
 - login.html
 - createaccount.html
 - home.html
- CSS Files:
 - style.css
- JS Files:
 - script.js
- Python Files:
 - __init__.py
 - user_db.py
 - refresh.py
 - api_calls.py
- SQLite Database:
 - tables.db
- API Files:
 - key_love_calculator.txt
 - key_NFL_schedule.txt
 - Note: the rest of our APIs do not need keys

Program Components:

- Login Button
- Username field
- Password field
- Logout Button
- Create Account button

- Back to Login Page button
- Pick Item Button
- Refresh player button
- Give item button
- See reaction button
- Final verdict button
- Schedule date button

Component Map:



Database Organization:

- Login Info
 - Username
 - Password

Site Map:



login.html:

- This is the first page the user navigates to.
- The user must login with a username and password to access the other pages (besides createaccount.html).

createaccount.html:

- The user can navigate to this page through the login page as well as navigate back to the login page through the create account page.
- The user has to create a username and a password.

- The username cannot be the same as another user's username.
- The username and password will be added to the users database in tables.db.
 - The password will use hashing to hide its content.
- Once an account is created, the user will be navigated back to the login page.

home.html:

- The user will be navigated to this page upon successfully logging in.
- This will include the NBA love story, which will utilize a total of five APIs.

style.css:

- This will have any custom CSS we use to stylize our site.

script.js:

- Buttons and what the buttons do
- Allow for more elements to be displayed on the page with the user's actions

__init__.py:

- Routes, managing server, Flask and connecting with html.

user_db.py:

- Dedicated to log in system and account creation

refresh.py:

- Algo to ensure the two NBA players aren't the same

api_calls.py:

- Retrieve stats on the NBA players
- Get the random item from the Fake-Store
- Return yes/no and the corresponding gif
- Retrieve data from the love calculator

- Retrieve the NFL Schedule date

Tasks:

- ☐ Create account page
 - ☐ SQLite table (Gitae)
- ☐ Login page (Diana)
- ☐ Home page
 - ☐ Narrative of the love story (All)
 - ☐ Creating buttons that continue the story (Vivian)
 - ☐ Retrieving NBA name, age, and stats from the NBA API (Samantha)
 - ☐ Retrieving an item from the Fake-Store API (Samantha)
 - ☐ Returning a yes/no GIF from the Yes/No API (Samantha)
 - ☐ Retrieving data from the Love Calculator API
 - ☐ Retrieving a date from the NFL Schedule API (Samantha)
- ☐ Flask, connecting components (Diana)
- ☐ Bootstrap/CSS formatting/JavaScript(?) (Vivian/Diana)

Stretch Goals:

- Password requirements (ex: 12 char min), hashing password

5 APIs:

- [NBA Player API](#)
 - Name/Age/Stats (display players with cards - Bootstrap)
- [Fake-Store API](#)
 - Items one of the players will give to the other player
- [Yes/No API](#)
 - Return a GIF
- [Love Calculator API](#)
 - Calculate the likelihood of the two players working out
 - If previous answer was yes, multiplier for percentage (~1.5x)
- [NFL Schedule API](#)

- Use the times to schedule a date between NBA player

FEF: Bootstrap because the aesthetic is nicer, has cool features like cards, which is useful for our specific storytelling and characters.

Storyline:

Two NBA players try to start a relationship -> one of them tries to buy something for the other person -> yes/no to see whether the other person likes it -> love calculator to see compatibility -> if compatibility/percent is high enough, we use NFL schedule to schedule a date.

- Each step/scenario will be on the same page (home.html). With each progression, more elements will display on home.html. The user will not be able to edit any of their past actions but will be able to restart the process (match two players again).