UnitedPurpleNPCs: Henry Bach, Jeffery Tang, David Deng, Shafiul Haque
SoftDev Period 8
P1 – Stalker
2022-12-22
Time Spent: 12 hours
Target Ship Date: 2023-01-05

Current Plan:

- User Feature:
    - Log in/log out features with ability to hide and show password
    - Login page
        - Spaces to input username and password
        - Submit button to lead to home page
        - Sign up button to lead to sign up page
        - Show password checkbox to show/hide current input in password bar
    - Sign up page
        - Space to input username and spaces to enter password twice
            - Passwords have to match for an account to be created
        - Sign up button to create account and redirect to home page
        - Back to site button to return to site
        - Login button to go to login page
        - Show password checkbox to show/hide current input in password bar
- Getting the information:
    - Use **IPify API** get the IP of the user (user consent does not matter, we will get it either way)
    - Use **IPStack API** to get the location of the user using their IP
    - **WorldTimeAPI** to get time of user using their location
    - Use **WeatherBit API** to get the weather in that location
- Submit Button:
    - Use **LoveCalculatorAPI** to calculate user's results using info obtained from aforementioned APIs (gives percentages)
- Output:
    - Similar results feature (User can see other users with results that line up)
    - Use the percentages to display the possibilities of many things (chance you'll find love today, chance you're existence upon this planet that I write this from will end today, etc.)
    - Keeps those results and specifics in db to share as necessary (store most recent)

Possible Stretch Goals:

- DM them their results over Discord through a bot created using DiscordAPI
- Link discord with your account login created through our websites

List of Program Components:

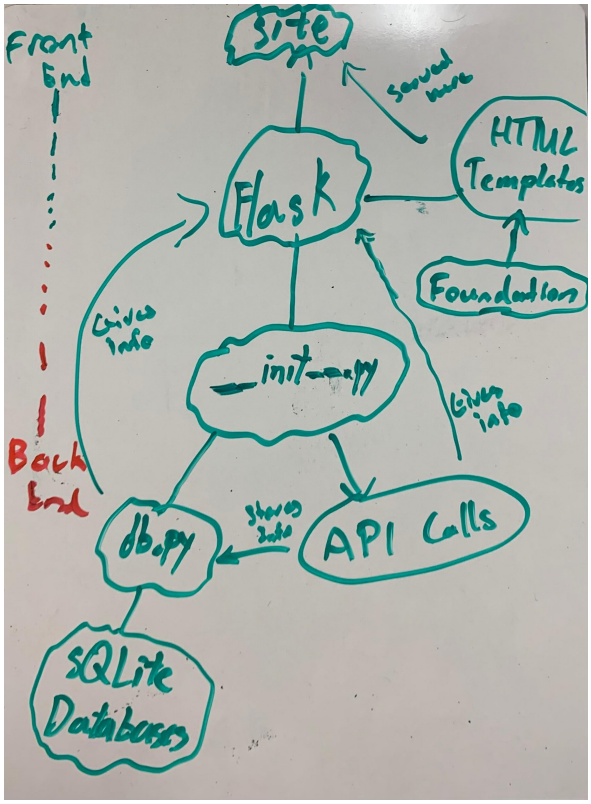
- Templates
    - Use **Foundation** as a Front End Framework to format pages (explanation as to why provided later)
    - Serves as a foundation for front-end to display the website (__init__.py)
    - Database data displays in certain sections of the html
    - The Stalker and United Purple NPCs names are visible at the top of every page and clicking our team name will redirect to our about page
    - index.html
        - The landing/main page (When logged in)
        - The path that the flask would go through is /
        - Welcome message
        - Lists info about the user obtained through APIs
        - Discover section
            - See your results button
                - Calculates results and redirects to results.html
            - Similarities button
                - Redirects to similar.html
        - There would be a logout button at the top to return to index.html
    - main.html
        - The landing/main page (When not logged in)
        - The path that the flask would go through is /
        - Project title and description
        - There will be a login and sign up button
    - login.html
        - The login page
        - The path that flask would go through is /login
        - If a user is already logged in, then this page would automatically redirect to the main page
        - Spaces to input username and password
        - Errors for when username doesn't exist or password is wrong
        - There are signup and back to site buttons for navigations
        - Show password checkbox to show/hide current input in password bar
    - signup.html
        - The signup page
        - The path that flask would go through is /signup
        - A user who isn't registered already would signup here
        - There are login and back to site buttons for navigations
        - Space for username and 2 spaces for passwords

            - Passwords have to match to create an account
        - Sign up button to create account and redirect to home page
        - Show password checkbox to show/hide current input in password bar
    - results.html
        - The results page
        - The path that flask would go through is /results
        - A user will be able to see their results
        - You can navigate back to the main homepage or see similarities
    - similar.html
        - The similar page
        - The path that flask would go through is /similar
        - Shows similar users
- Flask App
    - Helps display the html files through paths for certain linkages
    - SQLite database functions references in order to add/delete data in the database
    - __init__.py
        - routes to connect to HTML files
    - db.py
        - Log in and out
        - Grab data from API and retrieve and set database information
- SQLite Database
    - Displays information in the html templates
    - Functions are used in the flask app to add/delete data in the database
    - p1_info.db
        - Log in and out
        - Store data from APIs (so we can find users with similar results)

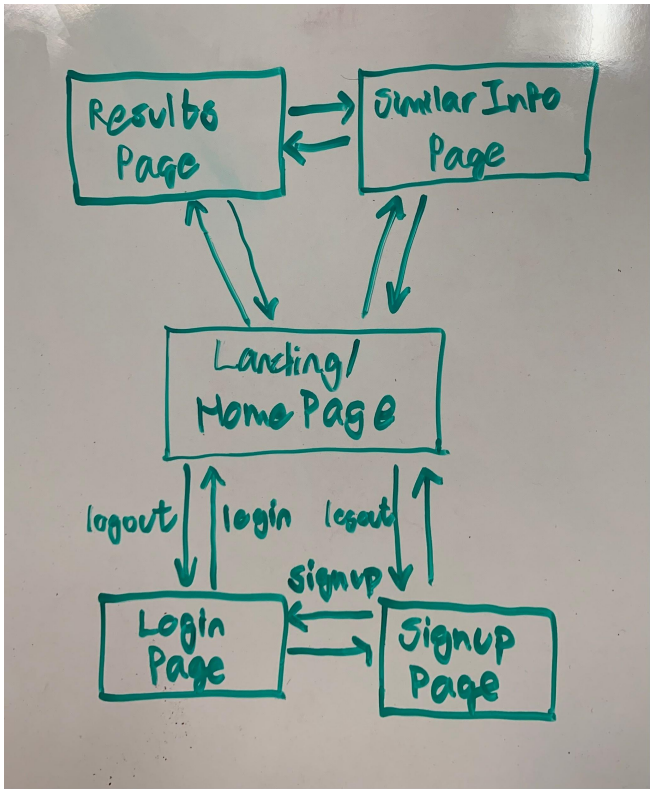
Database Structure:

- users table with usernames (must be unique!) and passwords
- info table that contains username, city, weekday, weather, temperature, town, cont, country, abbr, time

Component Map:


Front End
Site
Served from
HTML Templates
Flask
Foundations
Given info
__init__.py
Given info
Back End
Stores info
db.py
API Calls
SQLite Database

Site Map:


Results Page
Similar Info Page
Landing/ Home Page
logout
login
logout
signup
Login Page
Signup Page

Tasks:
- create the database and functions required for the app
- create html templates to display information and linking redirects from one page to another
- create the python app file in order to have the app running
- import the API data successfully, and use it in order to display data
- Maybe read from database and display previous results ~~leak other people's information~~ on the website as a possible frontend feature since right now, we only have a single form that will take their Discord username and their IP (I think we can get it even if they don't enter it so…) as website content.

General Focuses (per member):
- Jeffery (sqlite)
- Shafiul (html)
- David (api/flask)
- Henry (api/flask)

APIs: IPify API, IPStack API, WorldTimeAPI, LoveCalculatorAPI, DiscordAPI (possibly)

Front-End Framework: Foundation
- the do it yourself style offers more unique and custom designs for websites
- more freedom on components such as grids and cells, which we will make use of