

TNPG: MegaNuclearSuperJesus

Roster: Jacob Guo, Marc Jiang, Nicole Zhou, Sam Cowan

Soft Dev

2022-12-05

Target ship date: 2022-12-21

### Our Project:

-----  
-

College picker and it offers weather/climate of that region, commute or travel to that location, and other stuff TBD.

Search bar (with dropdown if possible) on homepage (and smaller on other pages) which you can type a college name into. If there is a dropdown, you can select an option to autofill and search.

Query Results page gets the latitude and longitude of the college from geonames and plugs into a few apis. OpenTripMap (and perhaps google maps/places) finds points of interest, and lists the top five out on the page. Uses some api to determine travel time by car, train, plane, and price of all. Should include a picture of campus, and maybe of the places around it too. Lists events in the next year, places to travel to on weekends/breaks and prices. Indicate weather.

Purpose is mainly to help a high school senior prepare for their first year of college, helping explain logistics of getting to and from college, etc

### Program List

1. **\_\_init\_\_.py** - Basic driver with Flask and Foundation
2. static/css/ - Basic formatting that FEF doesn't take care of
3. js/ - Keeping it to a minimum
4. templates/ - our HTML files for the respective pages we need
  - a. login
  - b. signup
  - c. home
  - d. results
    - i. College name, location, distance/travel cost, weather/climate of location

e. (Please recommend more)

### Component Map

- **/:** users can login in with an existing account. If information entered is incorrect, the user stays on this page. If the correct username and password are entered, the user gets directed to the home page (/home). If the user does not have an account, they can go to the signup page and signup.
- **/signup:** where users create an account. After creating an account, the user can then login.
- **/home:** user can search for colleges (default tab). The user can also go to the "favorite" tab and check colleges in their favorite list. After searching for a specific college, it brings the user to the results page (/results).
- **/results:** shows information about the specific college researched.

### Database Organization

2 tables:

1) To store user login credentials:

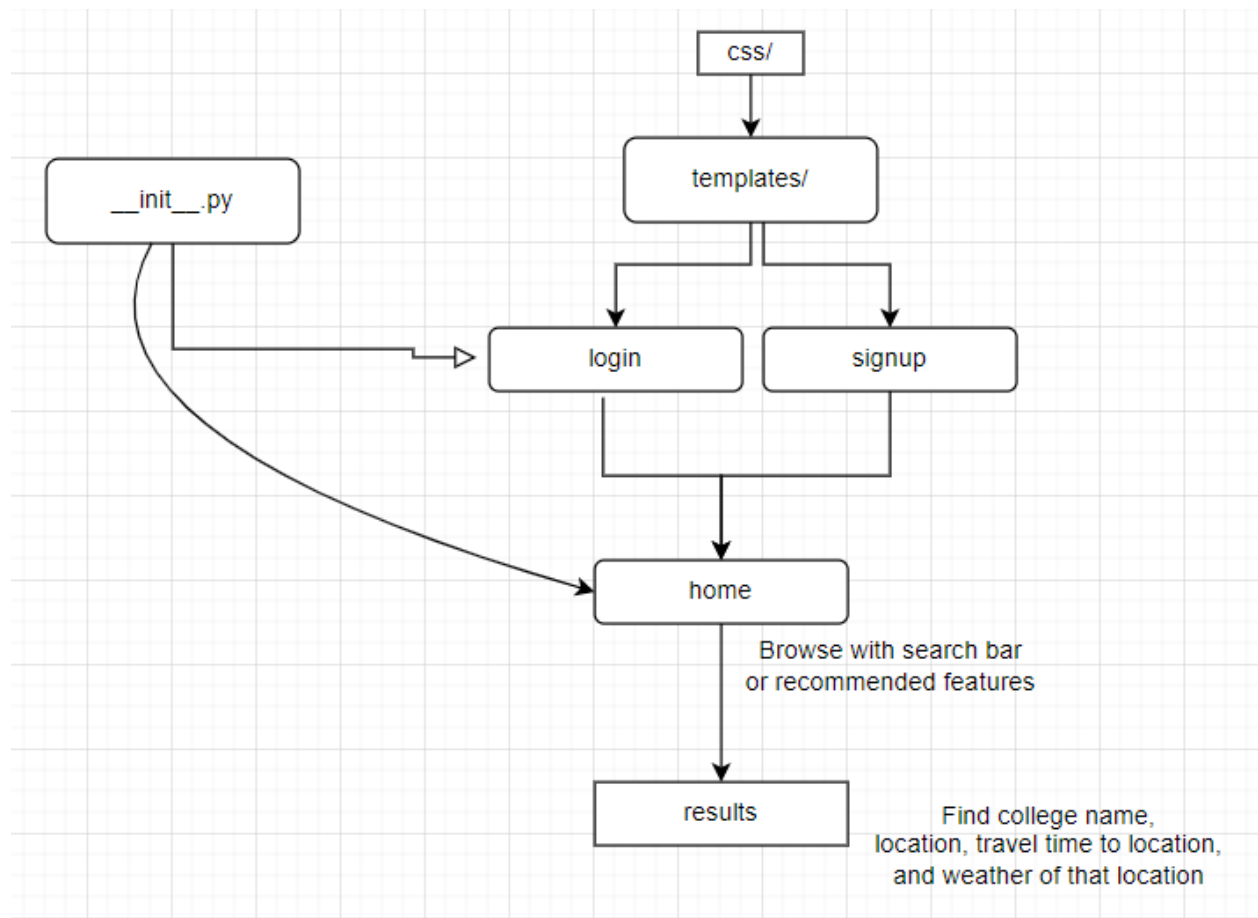
| Username<br>(TEXT_UNIQUE) | Password (TEXT) |
|---------------------------|-----------------|
| marc123                   | password        |
| jguo30                    | guoster123      |

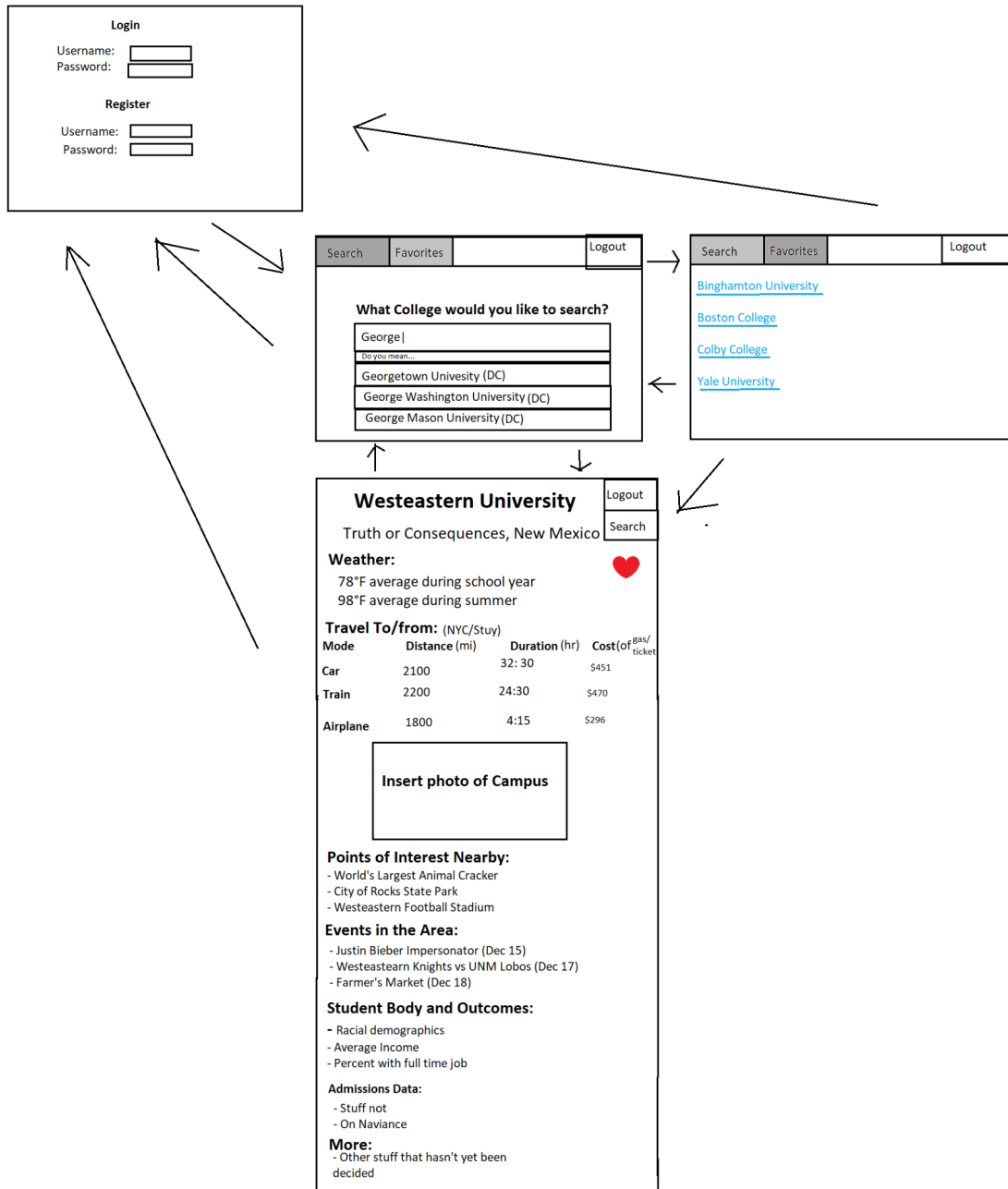
2) To store favorited colleges:

| Username | Favorited colleges (append name of college to string) |
|----------|---|
| Marc     | Westeastern, Hardvard, Princesston, UPeen             |
| jguo30   | Bing  |



## Site Map (Front-end)





### Assignments to Members

Jacob

- PM (the best)
  - Talking to Mr. Mykolyk and organizing the group
- Flask worker
  - Write flask things

Marc

- Database Engineer
  - Write database functions and understand database structure

Sam

- API Specialist
  - Write API kbs, learn how to use each API to get desired results
- Flask incorporator
  - Write flask things (with specialty in connection to APIs)

Nicole

- Front end framework (foundation)
  - HTML and CSS in conjunction with Foundation

MVP:

- Login page w/ corresponding DB
- Search bar on homepage which works with college names
- Query page providing basic info (as presented in site map) about colleges

Goals:

- Dropdown under search bar as autofill field
- Favorites Section added to navbar on "homepage"
- Calculations for certain fields with API generated numbers (such as price of driving with current gas prices)

Stretch Goals:

- Recommend the best (easiest and most cost-effective) form of travel
- Search for college given features instead of name (requires a lot of DB tinkering)
- Keep track of past searches

API Section

- 1) Bing Maps
  - a) Calculate distance and travel time from one point to another
    - i) Walking, car, plane(?)
- 2) College Scorecard
  - a) Contains tons of info on colleges
    - i) College name
    - ii) location (latitude, longitude to be used w/ Bing Maps)

- iii) type of area (urban, rural, big city, small city, etc.)
  - iv) General info (amount of students, cost, etc.)
- 3) Weather API
- a) Fetch local weather and climate info based on location (latitude, longitude, or city)

FEF Features Why and How?

#### Foundation

- Search bar
- Dropdown bar
- Checkbox to add colleges to the favorite list

We chose foundation because it was conceptually easier. We also liked the grid features.