

Crazy Joe Speaks Frogs:: Kevin Wang (PM), Shreya Roy, Faiyaz Rafee,
Justin Mohabir
SoftDev
P01 -- ArRESTed Development
2022-12-4
time spent: 1 hrs

Target ship date: W **2022-12-21**

Design V.0 for GOOGOL MAPS!

Goal:

Create boot-leg Google Maps. An address inputted from the client is then converted into coordinates to be passed into the yelp restaurants API and overpass API. Returns nearest restaurants and amenities within 5 miles of the client's inputted address.

Roles:

API stuff: Shreya

Frontend: Justin

DB management: Faiyaz

PM & moral support: Kevin

Frontend Framework Chosen: Bootstrap

- Why:
 - Bootstrap is easier to work with & looks cleaner.
 - Bootstrap containers are epic
 - Foundation is bad :D
 - Using only flexbox rather than grid/flex grid/etc
- How:
 - Containers to make sure everything is neatly organized in accordance with screen size
 - Forms to accept user input for a location or address

APIs Utilized:

- https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_PositionStack.md
 - Input Address -> output coordinates
 - 25000 quota & key

- https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_Yelp.md
 - Input coordinates -> output nearest restaurants
 - Maximum of 5,000 calls/day
- https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_OpenStreetMap_overpass-api.md
 - Input coordinates -> output nearest transit & places
 - no quota no key

DB:

Authentication

Username	Password
devol	hehehaha
devo3	gabagoo

Past Searches

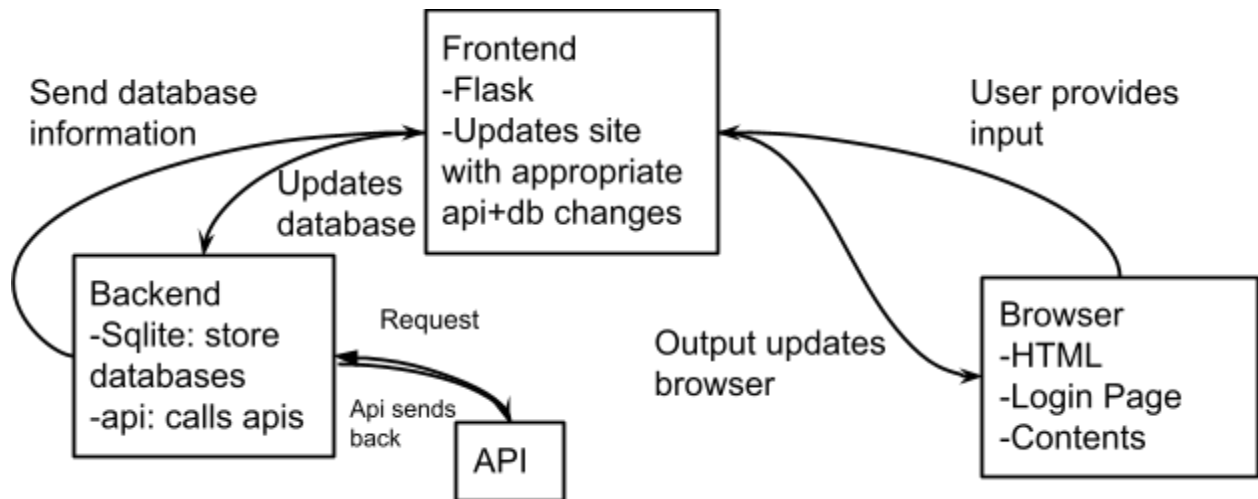
username	Past Searches
ThonnySinclair	345 Chambers Street
ThonnySinclair	1 Empire Avenue

Program Components:

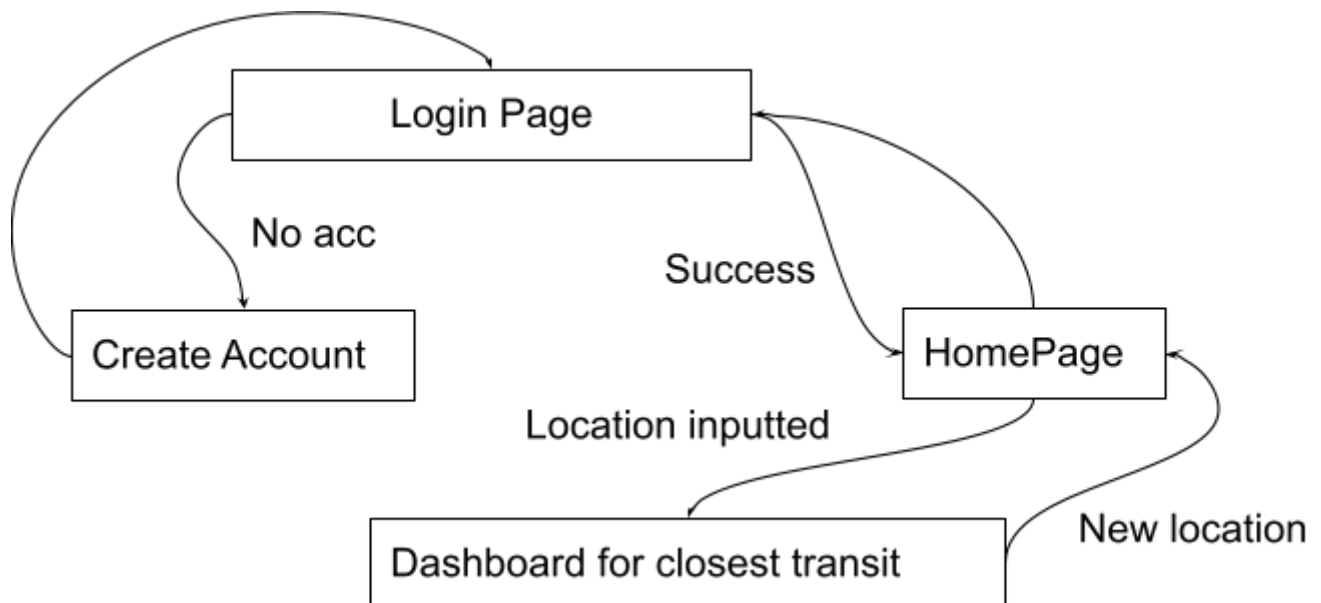
- `__init__.py`
 - Where the flask magic happens
 - Renders templates
 - Uses `database.py` to interact with the db and handle username/password authentication, as well as past searches
 - Takes in user changes -> sanitizes it from methods in `api.py` -> retrieves data from apis from methods in `api.py` -> renders templates from it
- `database.db`
 - A table for usernames and passwords
 - A table for storing username and their searches
- `db.py`

- Methods for updating, viewing, and creating new entries in the db
- api.py
 - retrieving info from apis
 - Cleans and parses data from api response to be used in __init__.py

Component Map:



Site Map:



Templates:

login.html

- Bootstrap form asking for username and password
- Displays error if incorrect inputs

register.html

- Bootstrap form asking for username and password
- Displays error if the username is already taken
- Password/username checking

home.html

- Has a nav bar
- Shows previously searched locations
- Bootstrap form for navigating to a new location
 - Accepts: address (best) / place name (worse)

dashboard.html

- Has a nav bar
- Map showing the relative directions of restaurants, transit/amenities. (relative to the central point of the searched address/location)
- Names of amenities on the side
- Restaurant carousel on the side
- Relevant cool statistics & data included in the apis

Flask:

__init__.py

/home:

- Access argument username and all searches
- Display all search results in home.html template

/login:

- Access arguments username, password
- Checks with form inputs
- Logs in/error message if incorrect username/pswd

/register:

- Accesses arguments username, password and appends that data to users table
- Ensures that username/password meets requirements

/dashboard:

- Shows closest locations
- Shows relative distances to restaurants and amenities on a map
- Shows restaurant names and information