FLASCC
Fabiha Ahmed, Cynthia Cheng, Stanley Lin
Eatables

API links:
Zomato: https://developers.zomato.com/documentation#/
Directions: https://developers.google.com/maps/documentation/directions/

Components:
Front End

- home.html - User is forced to input their address then shown a search bar. Their address is checked to see if it is valid then stored in a cookie. (address check using maps api) If the address is valid, the user is shown a search page where they can search for a restaurant (can search according to cuisine, names, type of food).
    - Form which will ask for user's address
    - Search bar for searching for restaurants

- results.html - User is shown a list of desired restaurants according to the keyword they searched for. It is listed in order from least to greatest the distance from the user's location. Each restaurant will have a small blurb with some information on the restaurant. (data is from API)
    - List of links to Possible Restaurants with Information on Restaurants

- info.html - Once the users select which restaurant they would like to go to, they are taken to a page with a full description of the restaurant (data is from API) and a list of directions on how to get to the restaurant from the address given in home.html.
    - Full description of the restaurant
    - Directions to restaurant
- design.css

Flask

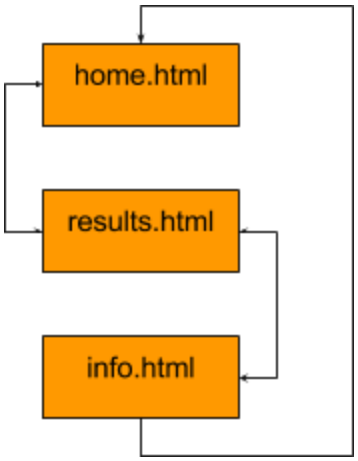
- app.py
    - root()
        - Renders home.page
    - result()
        - Renders results.html
    - info()
        - Renders info.html
    - get_parameters()
        - Gets form data (user location)
- maps.py
    - call_api(src, dest):
        - Params:

                - Src (string): starting address
                - Dest (string): destination address
            - Returns (object): a dictionary formatted version of the JSON response (to reduce number of API calls)
                - Returns None if geocoder-status is ZERO_RESULTS
    - get_time(response)
        - Params:
            - Response (object): python dictionary obtained from callApi()
        - Returns (float): the approximate total time (in minutes) needed to reach the destination
    - get_distance(response):
        - Params:
            - Response (object): python dictionary obtained from callApi()
        - Returns (float): the approximate distance (in miles) to the destination
    - get_directions(response):
        - Params:
            - Response (object): python dictionary obtained from callApi()
        - Returns (object): python list containing html formatted directions to the destination
- food.py
    - get_restaurantsInfo(usersKeyword)
        - Params:
            - User's keyword they used to search for restaurants on homepage
        - Returns (dictionary): python dictionary of restaurant info
    - get_rating(dict)
        - Params:
            - Dictionary of specified restaurant info from API
        - Returns (string): rating of restaurant
    - get_address(dict)
        - Params:
            - Dictionary of restaurant info from API
        - Returns (string): address of restaurant
    - get_menu(dict)
        - Params:
            - Dictionary of restaurant info from API
        - Returns (string): menu of restaurant
    - get_cuisines(dict)
        - Params:
            - Dictionary of restaurant info from API
        - Returns (string):
            - Cuisine of restaurant
    - get_name(dict)
        - Params:

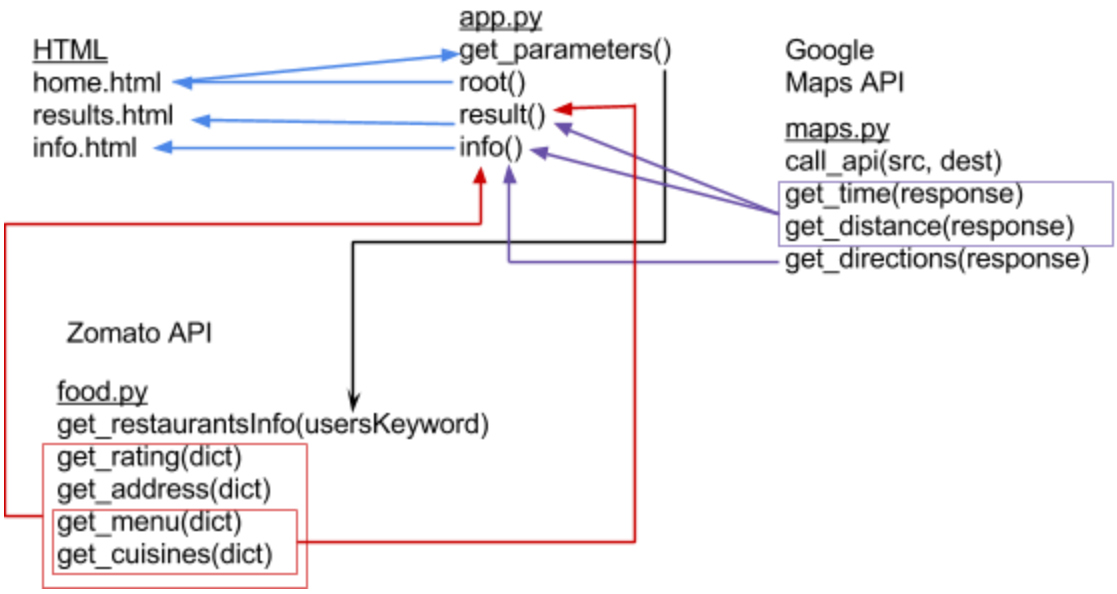
        - Dictionary of restaurant info from API
    - Returns (string):
        - name of restaurant

- get_numOfReviews(dict)
    - Params:
        - Dictionary of restaurant info from API
    - Returns (string):
        - Num of reviews of restaurant

Site map:



Component map:



Tasks:

Person	Task
Fabiha	Project Manager/ front end
Cynthia	Homepage/ Info Page, food API
Stanley	Results Page, maps API