Chariot (Github) (template)

Team: The Cavalry

Joseph Yusufov (PM)
Moududur Rahman (Backend focused)
Coby Sontag (Backend focused)
William Cao (Frontend focused)

Objective:

Generate a detailed itinerary given the user's location, preference, and amount of time available. This itinerary will lead the user on a path to see relevant events, landmarks, and other establishments of interest within the given parameters.

Background:

Travel is one of the luxuries of life, and tourists often find it difficult to make the most of their time in a city or region without help from someone that knows the region well. For this reason, those traveling abroad often consult a concierge or tour guide, using precious time and money that could be better spent getting a feel for the city or region that they're in. With an application that a user can run on the go, the need for a concierge or a Travel Guide is eliminated, and users are able to save time, and money.

Outline:

- Flask: Backend framework
- Python request: HTTPS handling (rather than urllib)
- Bootstrap: Frontend framework
- Sqlite3: Database
- JQuery: Javascript framework for frontend dynamicism
- APIS:
    - We can do what Brian Moses' group did and have the user enter API keys when logged in.
    - Finding directions and time to get from one place to another:
        - https://developers.google.com/maps/documentation/directions/intro#traffic-model
    - Finding places to visit and placeID for directions:
        - https://developers.google.com/places/web-service/search
        - -- Note: This requires a billing account
    - Yelp for reviews?:
        - Uses GraphQL -- not 100% if we can use this
        - https://www.yelp.com/developers/documentation/v3/get_started

Detailed Plan:

- Bootstrap used to create a responsive front-end
- Database, keeps track of user and their current location (for now)
- Requests for current address upon user creation
- Checkbox/ Text Input for user parameters
- NYC Events API
    - Parse the (limited) user input and print out nearby events
- Multiple Text Inputs that return all relevant events
- Incorporate events into the Google Maps API (Maps, Routes, Places)
    - Plot out where events are
    - Calculate travel time (walking/ driving for now)
    - Draw path from different events
- Generate timetable from event1, to travel time, to event2
- Display timetable along the path
- Implement itinerary constraints
- Restructure API calls to account for constraints, generate new timetable

Timeline:

2019-12-21 to 2019-12-28

- Drafting foundational Design Documentation, writing boilerplate code.
- Having frequent calls using Discord to coordinate and streamline work.

2019-12-29 to 2020-01-05

- Designing UI / UX of the service using Bootstrap 5, jQuery, JavaScript. Model UI/UX: http://zbib.org
- Implementing APIs and base features, without implementing any of our own algorithm
- APIs include:
    - Google Places API
    - Google Routes API
    - Google Places API
    - (QAF post and API card made for all APIs)

2019-01-06 to 2019-01-13

- Construct the Itinerary generator algorithm using a combination of proximity, user preference, and time available.
- All front-end and back-end work should ideally be in place at this point, so we can focus on having many calls / a lot of face time at the dojo, and write this algorithm. Pseudocode coming soon.

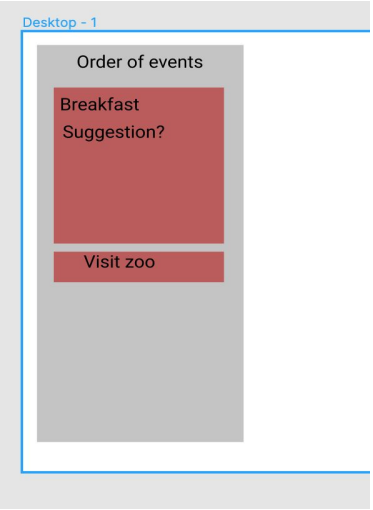
How to use the program:

1. Login
2. Preliminary information
    a. City
    b. Time to leave and finish by
    c. Tags (what you want to focus on sports/eating/watching movies)

3. Give a list of possible options for each event and user must choose

   List of events: [
   Breakfast,
   Watch movie,
   Visit a zoo,
   Lunch,
   Go home
   ]

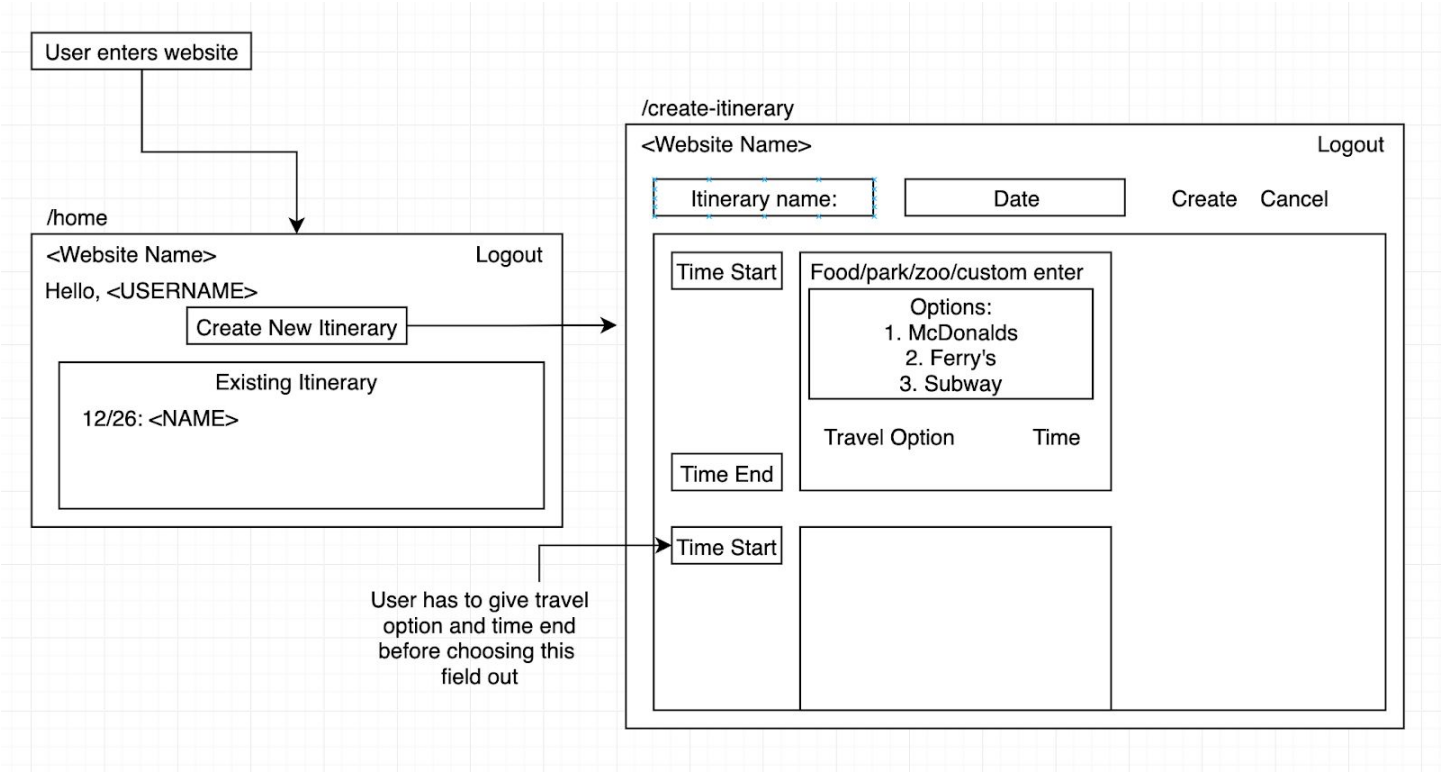
   List of tags:
   [
   Lunch,
   Movie,
   Visit zoo,
   …
   ]



Features:

- Itinerary Generator
  - Function: Generator given tags, time available **MOST DIFFICULT**
  - Function: Route planner given multiple locations, given current location
  - Function: Generate Itinerary given an order of things that you want to do (eat, then see a museum, then see a movie, etc.), and a zipcode.
- User System
  - ** EXTRA FEATURE ** Keep track of places that the user has already been to (User can enter events that they've been to in a "Track Activities" function). ** EXTRA FEATURE **

Sitemap



Database Design

users	
id	int
password	text
name	text
locationsVisited	text

location	
id	int
fullName	text
tag	int

APIKeys	
key	text
value	text

itinerary	
id	int
events	text
times	text

tags	
id	int
name	text