

SWE - Full Stack Interview

Background

Our team in Palo Alto are on a quest to discover the nearby food trucks! Your challenge is to make it possible for us to find a food truck no matter where our work takes us in the city.

This assignment consists of two components:

- 1) writing a web API that returns a set of food trucks and
- 2) write a web frontend that visualizes the nearby food trucks.

The requirement for the assignment is that it gives us at least 5 food trucks in a radius provided by the user or can keep a default radius of 5kms to choose from a particular latitude and longitude.

Use the code snippet below to calculate the distance between two coordinates-:

```
1 from math import sin, cos, sqrt, atan2, radians
2
3 # Approximate radius of earth in km
4 R = 6373.0
5
6 lat1 = radians(52.2296756)
7 lon1 = radians(21.0122287)
8 lat2 = radians(52.406374)
9 lon2 = radians(16.9251681)
10
11 dlon = lon2 - lon1
12 dlat = lat2 - lat1
13
14 a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
15 c = 2 * atan2(sqrt(a), sqrt(1 - a))
16
17 distance = R * c
18
19 print("Result: ", distance)
20 print("Should be: ", 278.546, "km")
```

Tackle this problem using a backend built in Python, the front end using React.js and use CSS or available components for styling.

Palo Alto Food Truck Dataset: https://drive.google.com/file/d/1sJImU2xuCL-OvaWXM95oG_TSYWF_OVOG/view?usp=sharing

Use this link to seed the data in any database of your choice and then create the backend APIs using python/Django. You're in charge of the UI - design one that you believe will enable the best experience to represent these results.

Also allow the user to share their current location from the browser as well (instead of user input long lat)

Remember to handle error cases and handle them well!

Validation

User Interface should return the locations of nearby food trucks based on the latitude and longitude provided by the user or by their location.

Estimates

The estimated time to finish this challenge is 2.5 hours. Breakup is as follows:

1. Creating backend application using Python (can use django, flask or any other framework) - 75-90mins
2. Creating frontend application using React.js - 30-45 mins
3. Adding unit-tests (good to have) - 30 mins

