Full Stack Coding task

Below is the coding task assigned to you. All other interview rounds are designed based on your coding task. So take time to fully complete the task.

Here are some mistakes that the candidates have done in our past experience, which should be avoided during the coding task:

- There is no documentation about the project. At Least a basic structure/workflow of the project in the Readme file
- No tests, which has already been mentioned in the mail.
- Few functions have been defined and written without using it anywhere in the project.
- The app is not modeled properly
- Not using basic programming practices

Code submission:

1. Even before starting to complete the task, please make sure to create a public github repo
2. Please commit your changes as you go along solving the task. When we review your git commits we should be able to get a fair idea about how you went along solving the problem
3. ReadMe for your project should include setup instructions to execute your solution or any other instructions/assumptions that can make it easy to review your code
4. Submit your github repo link once the coding task is completed.

Modelling Problem:

You are the proprietor of füber, an on call taxi service.

- You have a fleet of cabs at your disposal, and each cab has a location, determined by it’s latitude and longitude.
- A customer can call one of your taxis by providing their location, and you must assign the nearest taxi to the customer.
- Some customers are particular that they only ride around in pink cars, for hipster reasons. You must support this ability.
- When the cab is assigned to the customer, it can no longer pick up any other customers
- If there are no taxis available, you reject the customer's request.
- The customer ends the ride at some location. The cab waits around outside the customer’s house, and is available to be assigned to another customer.

Notes:

- You can build this in any programming language of your choice
- We expect good unit tests
- Unfortunately, you skipped Geography, and believe the earth is flat. The distance between two points can be calculated by Pythagoras’ theorem.
- We don’t expect a front end for this, but try to build an restful API.

Extra Credit:

- When the customer ends the ride, log the total amount the customer owes
- The price is 1 dogecoin per minute, and 2 dogecoin per kilometer. Pink cars cost an additional 5 dogecoin.
- HTML front end showing me all the cars available

Please Note:

- What we care mostly about is OO modeling, readability, and simplicity.
- Do not spend time tacking on databases, dependency injection, ORMs, authentication,admin modules for setting up data, and the like.
- We do not even require a UI or a web framework.
- It does not add anything to the review process and sometimes detracts from it.
- We're happy to see you store all your data in memory. We will, most likely, not even execute your code; only read your code and see whether it communicates its intent