1001
0101
1101
SAT

Are you the one?

SAT Recruiting Task

Task description **2**
Business Requirements 2
Technical Requirements 4
Tech stack: 5
Criteria and steps **5**
Estimation 5
When it will be ready 5
Implement it 5
Let’s talk 6
Summary **6**

Task description

Business Requirements

As the crisis regarding supply and demand of used vehicles in Poland gets bigger and cars are getting more expensive, a market for software solutions regarding the evaluation of used cars is emerging.

Due to that fact, as a SAT potential Junior Software Engineer, you need to face that challenge, and answer the call from the market.

SAT is asking you to prepare an API that will expose endpoints to calculate average use of diesel fuel and probability of engine breakdown - in one of the most popular car on the used market.

Due to NDA's and legal obligation we can not use the real name of the car, instead we would use an imaginary car. All similarities are results of coincidence and not intended.

Car that will be evaluated by your software is PeopleCar PasWagon C6.

The Software Development market has no mercy, and the only way to survive in it, is through quality of the code. Although it is not a strict business requirement, it is that much important that it's in this part of the document.

Technical Requirements

REST API with following endpoints:

- GET: /calculateDisselUsageForDistance
    - The endpoint should return a number, which is the fuel consumption on specified distance.
        - Inputs in request url(query parameters):
            - distance- total distance between point A and point B. Provided as a natural number. Please assume that unit measurement here is KM.
            - yearOfProduction - year of production of the car. Provided as a number.
            - fuelUsagePer100KM - natural number that represents approximate fuel consumption per 100KM. Provided as a number.
        - Returns:
            - fuelUsage- based on input, please perform calculations that will allow to define fuelConsumotion.
- GET: /probabilityOfUnitInjectorFail
    - The endpoint should return a percentage of the chance that the engine of the C6 model will fail on the Unit Injector element. Meaning “0” means there is no such possibility, and “0,77” means that there is a 77% chance that the Unit Injector will fail.
        - Inputs in request url(query parameters):
            - VIN - not relevant, but customer really wants it here
        - Returns:
            - failProbability - beforehand there were extensive R&D tasks performed, AI was used, we even searched through Google, including third and fourth page. All effort in the name of finding a way to calculate such a chance. Results show that randomly generated percentages that do not base on anything, accurately represent the chances of potential failure. Please use some method to generate random numbers from 0 to 100 and convert to the correct format.

Tech stack:

- Rust
- Whatever you need

You can use tools that are available in the wild(github.com/google), but you need to have a grasp of feeling what they are doing.

All moves allowed, besides one. Please do not copy paste code, be inspired by one but do not copy paste it, as it was mentioned before - we have rules.

Criteria and steps

Estimation (work-hours)

Please estimate how long it will take to implement such an API, in work-hours. Meaning real time spent on code. Send it back as a reply to the mail that you received with this PDF - before starting the task.

When it will be ready (date)

Please estimate when you will deliver the finished task. Once again reply back with a concrete date, before starting the task.

Implement it

Code the solution, put it on the repo(github/gitlab). README file should explain how to start the project.

Let’s talk

After reviewing the task we will either contact you through mail or ask you for a call, when we will go through the API and ask some questions, to make sure that you understand what is happening there(don't worry we will ask about your code not instruction sent to the CPU by the compiled code).

Summary

Good luck.

Do not be afraid to ask any questions, it will not be seen as negatives, if we decide to not answer, we will just say it.