SEAT:CODE

Hello,

This is our code test. Here you will find the description of the problem, what is required from you, and an explanation of any input and output to be produced by the code.

Hopefully the exercise will be intellectually interesting. In your answer we'll look for the patterns used, the architecture, the project infrastructure, modularization, your Clean Code practices, SOLID principles, code testability and testing. Do your best and and be prepared to discuss and argue your choices during the follow up interview.

Good luck. Any questions let us know. One final thing: please make sure that the code compiles and runs, and include any instructions on how to build and run it. No matter how incredible your solution, if it does not run, it does not count.

You are free to use external dependencies, but should be ready to justify the use of them.
Regarding the design, if you use auto-layout, the requirement is to not have any breaking constraints.

We expect you to deliver the final project containing the git history.
Thanks

The SEAT:CODE team

SEAT:CODE

Front End Problem Description

You are asked to build a simple trip manager for our bus on demand solution. This tool will be used by the operators of the service in order to see the trips available in the system. The problem has three tasks and you should solve each one before the next.

Task 1: Trip list:

- **Goal:**

The initial screen should show a map and scrollable list of the current available trips.

- **Guide**:
  - Each trip card should show some information: driver name, timestamps,... You can choose what you think is more important.

Task 2: Select trip:

- **Goal:**

When we click on one trip, it should be shown on the map. (map centers and zooms towards it)

- **Guide**:
  - The route of the trip and the points of each stop should be shown on the map. Also the start and finish points of the trip.
  - The map should be centered on the route and zoomed.
  - It should be clear to the user which trip has been selected.
  - The route returned by the API is a google encoded polyline. (check this)
    - You can use this third-party library to represent the polyline on the map if you want to.

Task 3: Stop info:

- **Goal:**

When we click on one stop, a popup with the information of the stop should be shown.

- **Guide**:
  - Each stop bubble should show some information about it: passenger, time,... You can choose what you think is more important.

SEAT:CODE

Task 4: Contact form:

- **Goal:**

On the main screen we should be able to open a contact form in order to report an issue.

- **Guide**:
    - The form should ask the name and surname of the user, email, phone (non-mandatory field), date and time of the reporting bug and a multiline input text (200 characters max) for the report description.
    - This data should be validated and stored locally (you don't need to send it to a server).
    - The application's icon badge number should display the total number of stored issues.

SEAT:CODE

API DOCUMENTATION

1. Trips

- Endpoint:

GET https://europe-west1-metropolis-fe-test.cloudfunctions.net/api/trips

- Response template:

```
[
    {
        "description": "Barcelona a Martorell", // description of the trip
        "driverName": "Alberto Morales", //name of the driver
        "route":
"sdq{Fc}iLj@zR|W~TryCzvC??do@jkKeiDxjIccLhiFqiE`uJqe@rlCy~B`t@sK|i@", //encoded
polyline of the trip
        "status": "ongoing", // status of the trip. Can be: ongoing, scheduled,
finalized or cancelled
        "origin": {
            "address": "metropolis:lab, Barcelona",
            "point": {
                "_latitude": 41.38074,
                "_longitude": 2.18594
            } // address and point in the earth for the trip origin
        },
        "stops": [
            {
                "id": 1,
                "point": {
                    "_latitude": 23,
                    "_longitude": 4
                } // each entry is the point and the id of every stop of the trip
            }
        ],
        "destination": {
            "address": "Seat HQ, Martorell",
            "point": {
                "_latitude": 41.49958,
                "_longitude": 1.90307
```

```
            } // address and point in the earth for the trip origin        },
        "endTime": "2018-12-18T09:00:00.000Z", //timestamp of the trip start
        "startTime": "2018-12-18T08:00:00.000Z" // timestamp of the trip end
    }
]
```

2. Stops

- Endpoint:

GET https://europe-west1-metropolis-fe-test.cloudfunctions.net/api/stops/{stopId}

- Response template:

```
{
    "stopTime": "2018-12-18T09:00:00.000Z", // timestamp of the stop
    "paid": true, // if the user has paid or not
    "address": "Ramblas, Barcelona", // address of the stop
    "tripId": 1, // id of the trip related to the stop
    "userName": "Manuel Gomez", // the name of the passenger
    "point": {
        "_latitude": 41.37653,
        "_longitude": 2.17924
    }, // latitude and longitude of the stop in a map
    "price": 1.5
}
```