Technical Challenge

Dear candidate,

Thank you so much for taking your time for going through this exercise.

Please consider this a way that we can get familiar with your technical knowledge. We would like to know the cool things you have been doing, and how you do think about technology.

We thank you in advance for doing this, and we hope you get to join our team.

Sincerely,

Team members.

EXERCISE

Goal

The purpose of the exercise is to evaluate your knowledge about Angular, Angular Material, Node.js, Express, and testing.

Introduction

Bayer invests heavily in seed technology to maximize yield in any soil and weather conditions. Our commercial department works very closely with farmers. These farmers plant seeds with our technology and expect maximum yield.

Description

Following a product workshop, where business needs are identified and captured as high level stories, The development team then grooms all high level stories and breaks them down into small vertical slices. The following story is prioritized and moved from our backlog to the sprint:

"AS Commercial User, I want to know the hectares planted divided per Lots from each farmer, So with that I can know how much the farmer needs to pay."

After analyzing the requirement, the team have sliced the story in small tasks. The following is one of the tasks that need to be tackled to complete the story.

- One webpage component, that could be reused, with the following:
  Angular 7, Angular Material

- Component to search farmers by document # or Name.

Technical Requirements

Website:

- Technologies:
  - Angular 7
  - Angular Material
- Pages:
  - Search form:
    Farmer search form should match mockups below.

    If the search has more than one result, return any result that match's.

    Once the farmer is selected, the farmer's details should be populated in the correct inputs.

    The component should have, at least, one input and one output:

```
@Input() farmerSearchAbstractProvider: FarmerSearchAbstractProvider;

@Output() onPartnerSelectedEvent: new EventEmitter();
```

This abstract provider should be implemented to allow us to reuse the same component for different source of farmers

```
import { Farmer } from '../model/Farmer;
export declare abstract class FarmerSearchAbstractProvider {
  abstract searchFarmers(params: SearchParams): Promise<Farmer[]>;
}
```

Use this farmer model as sample:

```
export interface Farmer {

    id: string;

    document: Document;

    name: string;

    address: Address;

}

export interface Address {

    street: string;

    state: string;

    address: string;

    country: string;

}

export interface Document {

    documentNumber: string;
    documentType: string;

}
```

The component should looks like this:

```
<farmer-search-card

            (onFarmerSelectedEvent)="mySelectedFarmer($event)"
            [farmerSearchAbstractProvider]="myFarmerSearchProvider">

</farmer-search-card>
```

Backend:

- Technologies:
  - Node.js
  - Express
  - PostgreSQL

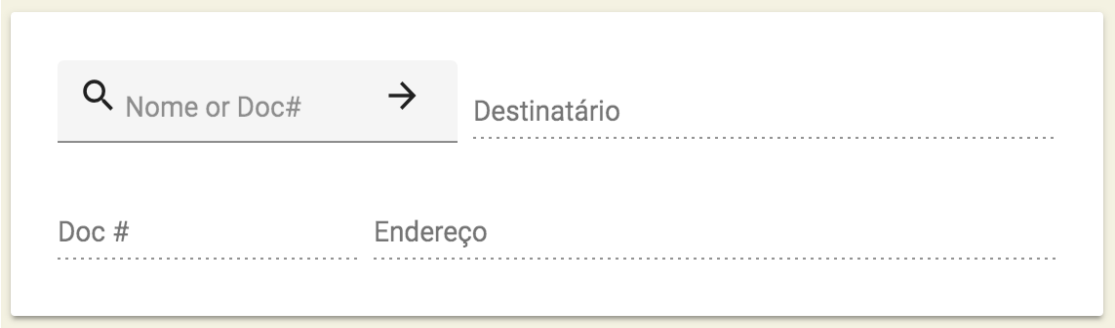
- Endpoints:
  - GET – Farmer search

    Endpoint should accept one search parameter.
    Endpoint should take parameter and search against farmer's document # and name.
    Endpoint should return list of search results.

Mockups

UX team collaborates with users and comes up with a mockup. This is the look and feel we need to follow. **This is a sample mockup, It is not mandatory to follow it as is in your solution and also it would need some extra CSS that is not required to follow for this exercise.**

Card for farmer search:



What is needed?

Delivering an implemented and working component.

The source data (Farmers) can be hardcoded. (not in the component but it could be in the backend side)

Using any open source library.

Include all tests (unit, integration tests is considered bonus)

Publish the solution to a public GitHub repository and share the URL as response at pre-interview form.

Include a README.md about how to install and run.