Coding Challenge

Thank you for showing interest in us. We appreciate you taking the time to take up this challenge.

Objective:

The key things we are looking for in your submission are:

- Good code performance (Without giving up readability). We are looking for code that is a simplistic and conventional route to the solution.
- You should imagine you are going to deliver this to your team and thus it shouldn't be overly complex and of Production quality.
- A consistent engineering style throughout.
- Using correct programming concepts is as important as solving the problem.

Description:

Build a CSV to JSON convertor API. Each row in the CSV file will represent one object and a file with multiple rows will be converted to a list of objects.
The fields in the csv files will be properties inside the object. A complex property will be named with a dot (.) separator.
Following properties will mandatorily be available in each record at the beginning.
name.firstName, name.lastName, age

For example:
A csv file with following content:

name.firstName, name.lastName, age, address.line1, address.line2, address.city, address.state, gender Rohit, Prasad, 35, A-563 Rakshak Society, New Pune Road, Pune, Maharashtra, male

JSOB Object Representation

```
1   {
2       "name": {
3           "firstName" : "Rohit",
4           "lastName" : "Prasad"
5       },
6       "age" : 35,
7       "address" : {
8           "line1" : "A-563 Rakshak Society",
9           "line2" : "New Pune Road",
10          "city" : "Pune",
11          "state" : "Maharashtra"
12      },
13      "gender" : "male"
14  }
```

You need to take a csv file from a configurable location(define an env config) in a node application (ExpressJs or NestJs) and upload the data into a postgres DB table with following structure

Please note that we expect that the candidate uses custom logic to parse csv file and covert the data to JSON rather than using publicly available NPM packages (For eg: csv-to-json)

```
CREATE TABLE public.users (
      "name" varchar NOT NULL, (name = firstName + lastName)
      age int4 NOT NULL, address
      jsonb NULL,
      additional_info jsonb NULL, id
      serial4 NOT NULL
);
```

Map mandatory property to designated fields of the table and put the remaining ones to additional_info filed as a JSON object.

Once the records are uploaded to the DB the application should calculate the age distribution of all users and print following report on console.

Age-Group	% Distribution
< 20	20
20 to 40	45
40 to 60	25
> 60	10

Please note
- First line in the csv file will always be labels for the properties
- Number of records in the file can go beyond **50000**
- You can have properties with infinite depth. (a.b.c.d……..z.a1.b1.c1…..)
- All sub-properties of a complex property will be placed next to each other in the file.

Submission Details:

Please share with us a working version of the application via Github. Also, let us know your assumptions if you have taken any.

What Next?

Once you submit your solution/code to us, Our team would review your code and schedule a technical interview within the next 2 to 3 days. The interview would be beyond this challenge. We would use this interview to understand your overall technical skills.