Assessment – Senior Software Engineer (ExpressJs/NestJs)

Problem Statement:

1. Develop an API that returns all the Employee Information (hierarchy by position) under any given position in the organogram.
2. Call another API (Can be another endpoint of your current api) with JWT token authorization.

Scenario:

Employee id 1: CTO
Employee Id 2: Senior software eng
Employee Id 3: Software eng
Employee Id 4: Junior software eng

Input:

Id : 1

Output:

JavaScript

```javascript
[
 {
   id: 2,
   name: "Name 2",
   positionId: 2,
   positionName: 'Senior software eng',
   child: [
     {
       id: 3,
       name: "name 3",
       positionId: 3,
       positionName: 'Software eng',
       child: [
         {
           id: 4,
           name: "name 4",
           positionId: 4,
           positionName: 'Junior software eng',
           child: null
         }
       ]
     }
   ]
 },
 {
   id: 5,
   name: "Name 5",
   positionId: 2,
   positionName: 'Senior software eng',
   child: [
     {
       id: 6,
       name: "name 6",
       positionId: 3,
       positionName: 'Software eng',
       child: null
     }
   ]
 }
]
```

Things to consider:

1. Build a web API project with all standard industry practice
2. Have to use migration file
3. Create a proper unit test

4. Create end-to-end test
5. Make sure the API can support 5000 calls at the same time and for 1m data the api response should be less than a sec.

Recommended Stack:

1. NestJs/Express JS
2. RDBMS(MySql/PostgreSql)
3. ORM (TypeOrm/Sequelize)

Rules:

1. If you do not complete the test please indicate how you would intend to finalize it in the README.
2. The team is looking to see how you approach a problem with a broad spec which could have a number of different solutions and then explain your approach? Keep the implementation simple, but make sure you have automated tests, logging (structured logs with JSON), and include information in the README about how you'll scale the solution to thousands of users.
3. How you'd approach logging & monitoring at scale so that you can actually debug the system as it increases in complexity.
4. We are not expecting the solution to be deployed, but we expect you to understand the process and best practices around the deployment process. It’s enough if you could provide our engineers clear and easy instructions on how to deploy your application.