cook·unity
A Chef Collective

Case Study - Chef Experience Squad

The Challenge

Please be aware there are no perfect solutions - we want to see how you find solutions, process your ideas, structure your thoughts and how you make your decision paths.

There are two thing we will see with this case study:

- Your coding skills: how you architect things, and how you actually implement your design in code.
- How you think about problems, and how you would create products that appropriately solves these problems.

At CookUnity, we evaluate candidates with respect to three different things:

- Personality:

  It's important to us that we work with people we like and respect. We spend much of our interviews trying to get a sense of a candidate's personality.

- Judgment:

  We look to see that candidates show good judgment, what features to implement, what priority should be placed on the different features, what the best use of their time is, etc. Judgment can blend in with knowledge, but we so often don't know the right answer, we need to depend on having good judgment to guide us.

- Coding skill:

  The primary job function is still writing code, and making a highly successful product. The ideal candidate would be teaching us how to improve our skills.

Exercise description

You have the option to choose between frontend or backend depending on your preference:
- If you choose frontend, please have your preferred IDE set up with a fresh install of ReactJs (skeleton app).
- If you prefer backend, please have a fresh install of Node Express or a similar framework.

Good luck!

Intro

Cookunity is looking to develop a new application that allows Customers to rate meals that are prepared by Chefs, and for Chefs to view the average rating for each of their meals.

Here are the high level application requirements:
- Chefs and Customers are able to log in
- Customers will see the full list of meals by default and are able to filter meals by chef
- Customers able to rate a meal once (1-5 stars)
- Chefs are able to create new meals
- Chefs are able to see a list of their meals and the average rating for each meal

The task

- Implement a small but scalable NPM express project and add the various components needed for the application, Ex: routing, components where certain business logic would be implemented etc.. It is important to call out how the components will be used and integrated with other components
- Business logic: build the business logic needed to calculate return list of meals for a chef and display the average meal rating: Assume the data model is an array of objects that looks like this:

```
Const meals =
[
  {
    Chef_name: 'John',
    Meal: 'EggplantParmesan',
    Rating: 5
  },
  {
    Chef_name: 'Tom',
    Meal: 'NoodleSoup',
    Rating: 4
  },
  {
    Chef_name: 'John',
    Meal: 'EggplantParmesan',
    Rating: 4
  },
  {
    Chef_name: 'John',
    Meal: 'ChickpeaSalad',
    Rating: 3
  }
]
```

- Bonus: Design a database schema for the application.(Diagram)

Technical requirements

Please use **express** as the project language. Consider using clean architecture, unit tests, dependency injection and whatever else you consider important.

Where to focus

We know that developing a case study takes time, so, please focus on the next quality areas:

• Design a clean architecture solution.
• Use the benefits of the technology, to write easy-to-read code.
• Don't forget your tests!

Note: Keep in mind that you can ask as many questions as you want to us and that you don't have to know everything, if you don't know how to face something technically but have some ideas please feel free to share it with us!