

Powering healthcare outside of the hospital.

hiring@gethealthie.com | <https://www.gethealthie.com>

Important note: Please do not use chat.openai.com or other AI text generators for any of your responses. You may use search engines or other research tools, if you do please indicate this next to each response. Utilizing these tools will result in disqualification.

Please do NOT include the `node_modules` directory in your submission.

Written answers

Pick one of the questions below.

- Describe why React is useful, and any thoughts on pros or cons you might have.
- If you wanted to improve the performance of a web app you were working on, what are some things you would look at / try?
- In your experience, what do you consider to be the difference between a mid-level and senior-level engineer?

Backend Coding

Please provide code in a runnable file, with some instructions in comments at the top for how to run it. In some questions, there may be instructions that are unclear or ambiguous. In those cases, choose your best interpretation and explain your thinking. Also feel free to reach out to us with questions. **We use Rails at Healthie so we'd love to see you showcase your Rails skills for these questions.** If you feel another language/framework is better suited, or if you're more comfortable with something else, please just explain why in your response.

1. Class design

Create classes Cat, Dog, Chicken, CatFood, DogFood, HumanFood, Lemon, Milk

- a cat likes CatFood, Chicken, and Milk
- a dog likes DogFood, CatFood, Chicken, HumanFood
- a Cat can meow and a Dog can bark (print "meow" or "bark" to STDOUT)
- when an animal eats food it likes, it makes a noise three times
- when an animal eats food it dislikes, it makes a noise one time

You can implement this any way you like, with whatever api you think makes sense.

Include a driver file which requires the files, instantiates the classes, and invokes the methods to demonstrate the functionality.

2. Database and model design with queries

We want to model providers (e.g. dietitians), their clients, and journal entries.

- Both providers and clients have a name and an email address.
- Providers have many clients
- Clients usually have one Provider but can have more than one.
- Clients have a plan that they paid for, signified by the string "basic" or "premium". For each provider that a client is signed up with, they have 1 plan.
- Clients post journal entries. These consist of freeform text.

We want to be able to do these queries:

- Find all clients for a particular provider
- Find all providers for a particular client
- Find all of a particular client's journal entries, sorted by date posted
- Find all of the journal entries of all of the clients of a particular provider, sorted by date posted

Using ActiveRecord, create the classes and schema migrations for this, including indexes. If possible, please use sqlite for your database.

Show how to do each of the queries listed above.

If you are torn between multiple approaches, just choose one to implement and explain the other approach and what the pros and cons of each might be.

3. JS/React/CSS

If you are a React coder, use React (*preferably Typescript*). Otherwise, code in the JS framework of your choice.

Make a frontend-only todo list app.

- It has three vertical lists. Todo, Doing, and Done
- Items are created via a form (they can be stored in browser state, no need to have a backend)
- Items are draggable from one list to another
- Items can be reordered within a list
- When you drag an item into the Done list, confetti explodes on the screen! Or something else interesting. You can use a package for this.
- Overall the app has a usable design with a unique color palette. You are free to make whatever UI and UX decisions you want to make it usable. Don't worry, we aren't looking for a groundbreaking innovative design. Just lay things out decently and add some colors.
- Using Typescript will get you bonus points.