VieMed assessment project – Web

The goal of this project is to produce a working system that can be efficiently evaluated by our engineers, and that we can discuss during your online interview. Be prepared to:

- Present your solution to a group of smart engineers like yourself.
- Talk about the decisions that went into the creation of your solution.
- Discuss technical design tradeoffs.

Additionally

- Please don't spend on the task more than 3-4 hours. If you don't have enough time to get all functions done, that's fine, but make sure that whatever you deliver is of good quality (see: "Evaluation" below) and please write precisely what is missing.
- Be explicit about all the assumptions you make (you can make any assumption you want).
- Keep in mind that we will read the code, try to run the service and evaluate it before the online session.

Evaluation

Main aspects we will evaluate:

- How much the codebase is production-ready (documentation, tests, etc.).
- Code structure, and organization of responsibility.
- Performance.
- Resilience to failures.

The specification

Write the online service that helps users to manage a simple list of tasks to get done. Users should be able to:

- Add a new task to the list. Each task has a name and a boolean status "isDone".
- Mark the task as "done".
- Delete the task.

Additional requirements

- Use React.
- Use GraphQL API we provide for keeping all users' task at the backend.

GraphQL API

- Mutation generateAccessToken - this mutation generates an access token for a given username and a given apiKey (you should receive apiKey from VieMed). Username is unique within the scope of the apiKey. You can call this mutation many times if needed. Each time you call "generateAccessKey" with the same apiKey and the same username you will get the access to the same list of tasks (but the value of the access token may be different).

All subsequent queries and mutations require access token provided as a value of HTTP header Authorization.

- Query allTasks - returns all tasks for a given user.
- Mutation createTask - creates a new task. Returns an object of type Task.
- Mutation updateTaskStatus - updates status "isDone" for a given task ID. Returns updated Task.
- Mutation deleteTask - deletes a task for a given task ID. Returns true if there was a task that was deleted. False, if there was no task for a given ID (nothing was deleted).