

Full-Stack homework

Overview

This is a coding assignment for a Full-Stack developer position at Aiven.

The exercise should be relatively fast to complete. You can spend as much time as you want to. If all this is very routine stuff for you, this should not take more than a day. If there are many new things, a few days should already be enough time.

Homework evaluation is one of the criteria we use when selecting the candidates for the interview, so pay attention that your solution demonstrates your skills in developing production quality code.

If you run out of time, please return a partial solution, and describe in your reply how you would continue having more time.

Please use React with Typescript on the frontend and Python on the backend for the exercise, otherwise, you have the freedom to select suitable tools and libraries. Make sure the work demonstrates well your own coding skills.

If you use a template (create-react-app or similar) for the homework, please add all the code from the template as initial commit, and your custom own code as separate commits.

Be prepared to explain your solution in the possible interview later.

To return your homework, store the code and related documentation on GitHub for easy access. Please send following information via email:

- link to the GitHub repository
- if you ran out of time and you are returning a partial solution, description of what is missing and how you would continue

Please provide instructions on how to check out and run the project. The project should not need root permission to run: e.g. does not start on port 80.

Your code will only be used for the evaluation.

Exercise

Aiven is a Database As a Service provider. A customer can launch a database in any of the supported clouds (like Google or Amazon cloud) using the Aiven web console.

Your task is to create a prototype web application for improved cloud selection logic. The application should include a frontend and a server implementation. The server should act as an intermediate cache and as a transformation layer on top of the public Aiven REST API.

Aiven's clouds can be listed using the API as described here

<https://api.aiven.io/doc/#tag/Cloud>.

It should be possible to:

- Filter clouds by the cloud provider (e.g. Amazon Web Services or Google Cloud Platform).
- Sort clouds by their distance to the user. The distance comparison should be based on geolocation. Our API returns latitude and longitude values for each cloud.

Criteria for evaluation

- Code formatting and clarity. We value readable code written for other developers, not for a tutorial, or as one-off hack.
- Practicality of testing. 100% test coverage may not be practical, and also having 100% coverage but no validation is not very useful.
- Automation. We like having things work automatically, instead of multi-step instructions to run misc commands to set up things. Similarly, CI is a relevant thing for automation.
- Attribution. If you take code from Google results, examples etc., add attributions. We all know new things are often written based on search results.
- "Open source ready" repository. It's very often a good idea to pretend the homework assignment in Github is used by random people (in practice, if you want to, you can delete/hide the repository as soon as we have seen it).