Homework assignment

WEB DEVELOPER

Thank you for taking the time to have a chat with us, it was nice getting to know you a bit better. As the next step of the recruitment process we want to share an assignment with you.

This is a coding assignment for the Web Developer position at Aiven.

Overview:

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 few hours. If there are many new things, a few evenings 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 for the exercise, otherwise, you have the freedom to select suitable tools and libraries, but 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 defend your solution in the possible interview later.

To return your homework, we will send you an invitation to a private Github repository where you can commit your code. Please confirm you`ve read this information and ask for the link.

Your code will only be used for the evaluation.

WEB DEVELOPER

Task:

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. Aiven's clouds can be listed using the API as described here https://api.aiven.io/doc/#tag/Cloud.

It should be possible to select and filter regions by the cloud provider (e.g. Amazon Web Services or Google Cloud Platform) and by distance (shortest distance to the user). The distance comparison should be based on Geolocation. Our API returns latitude and longitude values for each region.

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).