Java Backend Assignments

Writing

1. What’s your proudest achievement? It can be a personal project or something you’ve worked on professionally. Just a short paragraph is fine, but we’d love to know why you’re proud of it.
2. Tell us about a technical book or article you read recently, why you liked it, and why we should read it as well.
3. Tell me about one feature of upday you really like, and why.

Coding

This is the situation: We are a publishing company that created an app for reading news articles.

To serve data to the app we need a backend to provide RESTful APIs for the following use cases:

- allow an editor to create/update/delete an article
- display one article
- list all articles for a given author
- list all articles for a given period
- find all articles for a specific keyword

Each API should only return the data that is really needed to fulfill the use case.

An article usually consists of the following information:

- header
- short description
- text
- publish date
- author(s)
- keyword(s)

Hints:

- Use the Java technology you are most comfortable with (e.g. spring-boot).
- The data does not need to be persisted after the application is shut down.
- Using Kotlin would be a plus.

How we review

Your application/program will be reviewed by at least two of our engineers. We do take into consideration your experience level.

We value quality over feature-completeness. It is fine to leave things aside provided you call them out in your project's README. The goal of this code sample is to help us identify what you consider production-ready code. You should consider this code ready for final review with your colleague, i.e. this would be the last step before deploying to production.

The aspects of your code we will assess include:

- Architecture: how clean is you interface?
- Clarity: does the README clearly and concisely explain the problem and solution? Are technical trade-offs explained?
- Correctness: does the application/program do what was asked? If there is anything missing, does the README explain why it is missing?
- Code quality: is the code simple, easy to understand, and maintainable? Are there any code smells or other red flags? Does object-oriented code follow principles such as the single responsibility principle? Is the coding style consistent with the language's guidelines? Is it consistent throughout the codebase?
- Security: are there any obvious vulnerabilities?
- Testing: how thorough are the automated tests? Will they be difficult to change if the requirements of the application were to change? Are there some unit and some integration tests?
- We're not looking for full coverage (given time constraint) but just trying to get a feel for your testing skills.
- Technical choices: do choices of libraries, databases, architecture etc. seem appropriate for the chosen application?