Senior Full-Stack Engineer
ICE Take Home Task

Introduction

We're excited to have you as part of our interview process! This take-home task is designed to evaluate your technical skills, system design approach, and overall software craftsmanship. We encourage you to treat it as if you were implementing a production-ready system.

Context

Imagine you are building a responsive web application that would allow a user to easily keep track of their favourite songs of all time. Keep in mind that the web application needs to provide a user-friendly and snappy user interface.

Task Requirements

Your task is to design and implement a system that provides the following features and user experiences:

1. **Add a new song:** As a user, you should be able to add a song to your catalog, capturing information like song name, artist, album, year, song length and genre.
2. **Fetch the user's songs catalog:** As a user, you should be able to see all of your songs inside a table. The table needs to show information about each song.
3. **Filter the user's songs catalog:** As a user, you should be able to filter the displayed songs by release year and artist name.
4. **Sort the user's songs catalog:** As a user, you should be able to sort your songs by release year and song name.
5. **See more details about a song:** As a user, you should be able to click on any song and see more details about it in a different page.

These requirements brief provide an outline for the service. However, how you approach these requirements, how you design and implement the system, and how you anticipate and plan for potential issues is entirely up to you. We encourage you to make assumptions where necessary, but please ensure that you document and justify those assumptions.

Extra Information

- **Beautiful UI:** The UI needs to be responsive and follow UX best practices
- **Speed:** The table needs to be performant even with large numbers of rows, it should not make a difference if it shows 10 or 1000 songs.
- **Languages/Libraries:** Your application should be implemented using JavaScript/TypeScript and React on the FE, while the BE should use a JVM language and framework.
- **Unit Testing:** Write at least 1 unit test for the front-end web application.
- **Production-Ready:** Despite being a relatively straightforward application, we want your implementation to be ready for production. We understand that "production-ready" may have different interpretations, and we look forward to discussing what it means to you during the review.
- **Freedom of Choice**: Besides the requirements above, you are free to choose the build tools, libraries, frameworks, etc., that you want to use.

Deliverables

Please submit your solution either as a link to a public repository or as a zip file with all the necessary components for a production-ready system.

Be prepared to present and explain your design and implementation. If there are additional artefacts (e.g., slides, Miro-board, etc.) that you'll use during the presentation, please include them in your submission.

If you have any questions or require further clarification, feel free to reach out to us.

We're looking forward to receiving your solution!