# KOJI - Technical test for front-end developers

## General information

- The objective of the test is to understand your technical level, your ability to explain concepts and to resonate.
- This test is to be done independently and should not include copy pasted elements from any source. (aka Github or StackOverflow)
- You can make assumptions in order to complete the test. Don't forget to include them in your report.
- You can ask questions if you feel you are missing information.
- The test should take no longer than 1 hour. Send your progress once this time is up.
- Your code must be available on a public access Git
- You must send the answer to the questions and the repository link by email.

## A. Statement:

You are a developer working on a new project for a print content editing business application.
You are responsible for the front end of the application. It was decided that the application would be in Typescript with React.

During the scoping meeting you got the following information:

- The application must have an authentication system with a simple role management (user, admin)
- The administrator must be able to add a certain number of documents with the necessary data for the generation. These documents are considered as "templates".
- A user can only access his generated documents
- A user must be able to generate a document from a "template" created by an administrator by providing the required fields.

### Reflection questions:

You must answer the following questions (each question requires an answer followed by an explanation):

- Which library will you use to manage the state of the application?
- What pages are needed?
- How will you arrange your navigation?
- What are the minimum api requests you will need?
- How will you manage the execution of api requests and their asynchronous return?
- How will you manage authentication?
- Bonus: You need to implement an optional "dark" version in the application. How will you proceed ?

# B. Practical case:

> ⚠ This part is independent, you may choose to take a different approach than what you described in the first part.

You need to implement a React application using the selected methods. This application should have a page with a list of articles and one per article. The articles must come from the following api: https://jsonplaceholder.typicode.com/.