

# Case C - Fullstack

## Technical Test - Case C - Fullstack

Welcome to the [KOJI](#) recruitment process, the objective of this test is to understand your technical level and your ability to resonate with a minimal level of information.

**The estimated time for this test is around 2-3 hours**

### General information

- This test is to be done independently and should not include copy pasted elements from any source. (aka Github or StackOverflow)
- Using code you don't own apart from an external library will result in direct disqualification.
- You can make assumptions to complete the test. Don't forget to include them in your report.
- Your code must be provided by creating a pull request to this repository.
- Your code must be runnable only by following your [README.md](#) (the project will be run under the latest version of OSX)
- If you have any questions regarding this test, you should email them to the person in charge of your recruitment.
- You can use any external libraries you want.
- The design is up to you, but keep in mind that this app must be easy to use.
- Any feature not described below will be considered as a bonus, you can customize anything if the requested features are present

### Statement:

- You must create a React application allowing the user to follow the weather of a city.
- You must create an API to track the weather of the followed city.
- You can use Create React App to set up your application.
- You can use the Nest CLI to set up your API.
- It is strongly recommended to use NestJS to handle the API because of its simplicity but it's not required.
- The user must be able to select a city using a typeahead input, you can use any library/database to achieve this goal. (We will only search for French cities, you can skip other countries cities if needed)

- Only one city can be selected at a time.
- The weather must be provided by the [OpenWeatherMap](#) API, you can create an account if needed.
- You must handle possible errors with external API (invalid API token, network connectivity issue) by showing a message to the user in any way you want.
- You must provide a Dockerfile or a Docker-Compose that permits us to run the project.
- Using a SQL database is not required, you can use JSON file storage if you want.

## User Stories

- As a user, I want to see a text input so that I can search for a city by its name.
- As a user after performing a city search, I want to select a city, so I can follow its weather
- As a user with a followed city, I want to delete a city to stop following it
- As a user with a followed city, I want to see the weather history for a city since I have been following it
- As a user with a followed city, I want to see weather the city to be updated every 10 minutes even if I don't have the application open (The server is still up)
- As a user with a followed city, I want the history of the city to be deleted if I select another city

## Dictionary

- **Weather:** In this test, the weather refers to the meteorological conditions and the temperature.