

Cross Functional Test

This is the specification for the technical test we are doing before we move forward with the process to hire you long-term hourly on an ongoing basis.

DISCLAIMER: We are going to pay you 8 hours for the test JUST if you are hired. We will consider the test a success if you get at least 80 points but your attitude and english communication skills are being evaluated too. You could still get 80 points and not be hired.

Test summary

Create a form with some validations and save the information to the database using the required tech stack.

The form has Name, Last Name, Country Flag + Phone number, email, password, and main form button.

There is a second view with a success message.

Figma prototype for the test. It is a working prototype. Please click on the fields

[Prototype](#)

Mandatory tech for the test

FrontEnd

- ReactJS <https://es.reactjs.org/>
- Typescript
- React Router <https://reactrouter.com/>
- Material-ui <https://mui.com/>

Backend

- Nodejs or Java or Go
- Spring (If Java is used)

BD

- Mongo or Mysql

Test fails if

1. If the mandatory tech is not used the test will be considered FAILED
2. If the project won't run with the README instructions the test will be considered FAILED
3. If you get less than 80 points, the test will be considered FAILED

Test points

1. The name and last name are required, have the Capital letter (e.g William), parse the input so the capital letter is set automatically **(5pts)**
2. The phone gets formatted as US format and has the US flag (Required)**(5pts)** - United States format: (436) 463-4364.
3. Email always starts with lowercase and format is correct and validated (Required)**(5pts)**
4. Use of a password field (Required) with the following conditions **(5pts)**: - 1+ uppercase character - 1+ lowercase character - 1+ number - 8 characters long
5. The button is only activated once the form is correct **(5pts)**
6. On button click turn on the loader on the button and information is saved into the database. **(15pts)**
7. Loading the success view looking as designed with the form information **(5pts)**
8. Backend validates the data **(5pts)**
9. Backend has unit tests **(5pts)**
10. Backend project is well structure using patterns and quality code (ESLint or Sonar will be use to evaluate code quality) **(10pts)**
11. React best practices and folder structure. (ESLint will be use to evaluate code quality) **(5pts)**
12. Use of Github **(5pts)**
13. Use of mongo or mysql **(5pts)**
14. Use docker to run the project **(10pts)**
15. The solution has 2 projects (Frontend client, Backend Project) **(10pts)**

Bonus (Only if you have time)

1. Deploy the to a cloud provider **(20 pts)**
2. Use reactive programming on backend **(10pts)**
3. Create a pipeline on Github actions to build the project on every push **(5 pts)**

You will pass the test if

The test will be considered a success if you got at least 80 pts

