TECHNICAL TEST

The purpose of this technical test is to create an end-to-end web application that allows the user to read, create and update a to-do list of duties of any kind.

Required: A backend written in Node JS with TypeScript in strict mode, using PostgreSQL as a database engine. It must be able to perform read, create and update operations (Using SQL. Not allowed use of ORM).

Required: A frontend developed in React written in TypeScript in strict mode, using hooks that render the list of duties retrieved from the backend and allows the user to create new ones and/or modify existing ones. Must include form validations.

Example of the data structure for each duty as follows:

```
interface Duty {
    id: string;
    name: string;
}
```

Required: The code is published in a public git repository from which anybody could clone the solution.

Required: A README.md file is written with clear instructions on how to run the solution.

Note: Build and serve commands must run on all operating systems (Windows, macOS and Linux).

Required: Production ready software, meaning readability and observability

Required: correct error communication and error handling

Required: rest services

Required: Unit testing using jest (frontend & backend)

Optional: The solution is also able to delete items from the to-do list.

Optional: GraphQL is used instead of REST or any other.

Optional: Ant Design as the component library for the frontend