Frontend Developer Challenge

Context

Our user is an industrial engineer (IE) at an electronic manufacturing service (EMS). She wants to view an overview of all of her open projects. Since there are 100 available she wants a way to sort them so the most relevant ones are displayed first.

One Project consists of:

- Project Name
- Creation Date
- Status: This is either inProgress, won or lost.
- id

User Story Acceptance Criteria:

1. As a User Given that I have 100 Projects then I can see 100 Project cards with data on the name, creation date and status.
2. Each project must be displayed as a card (e.g. https://material-ui.com/components/cards/) and not in a table.
3. As a User, Given that I am on the Page, Then I can see 2 buttons to sort by creation date.
4. As a User, Given that I have selected sort by earliest then I can see the earliest first
5. As a User, Given that I have selected sort by latest then I can see the latest first

Task

1. Write a small application fulfilling the acceptance criteria above.
2. The application should be in React, ideally using Typescript and the Material UI library.
3. We have created a basic setup (Create React App with a server installed) in the zip file you received with with the email. You can use this as a starting point.
    1. run `yarn` to install the packages.
    2. run `yarn run-server` to run the mock backend. The API route for projects is a GET request: `http://localhost:3004/projects`
    3. run `yarn start` to run the frontend
4. Data should be retrieved from the api endpoint.
5. Write some unit tests to ensure that the sorting function works as expected.

API Design

- The call `/projects` returns an object with a \`data\` property which contains an array of objects of the following type:

```
{
'projectName': string // name of the project
'creationDate': string // date in the form of ISO8601 https://opensource.zalando.com/restful-api-guidelines/#126
'status': string // can either be inProgress, won, lost
'id': string //uuid of the project
}
```

Hints

- Feel free to use js and/or libraries for the sorting (i.e. do not implement your own sorting algorithm but use a sorting library or js. sort).

Submission Details

We do not expect you to spend more than a few hours on this challenge, that means that you should cut corners (in terms of the functionality and the design of the website, not in terms of code quality) wherever you feel like it. In general, the focus should be more on the functionality detailed above than an exceptional design.

When you are done, please email us with the an attached zip file (including the .git folder but without the node_modules folder).

Good Luck and let us know if you have any questions! 😌