# Frontend Technical Test

## Objective

Use **React JS** as the JavaScript framework.

Use **React-router** for views management

Use **Redux** for state-management

Use **Aggrid** for table

Use [Material-UI](https://material-ui.com/) or [Antd](https://ant.design/" \t "_blank) as the CSS framework

Make use of **Create React App** cli to create a simple “Employee” manager

## Instructions

Create a Homepage. The home page url will be (<http://localhost:3000/employee/list>)

The homepage will display a list of employees with following mentioned attributes in a table

1. First name
2. Last name mdz
3. Email address
4. Phone number
5. Gender
6. Edit/Delete button on each row at the end of the row.

You will be able to get an initial list of employees from <https://6164f6e709a29d0017c88ed9.mockapi.io/fetest/employees> with a GET request.

In the same page, Above Table, in the top right corner, should have an option to “Add” Employee. By clicking on “Add” Employee has to take you to next page with the URL (<http://localhost/employee/add>) with following mentioned inputs:

1. First name (Reusable Textbox) *Minimum 6 character and max 10 characters validation.*
2. Last name (Reusable Textbox) *Minimum 6 character and max 10 characters validation.*
3. Email address (Reusable Textbox) *email validation*.
4. Phone number: *SG phone number validation*.
5. Gender *(Radio Button Group)*
6. “Submit” button at the bottom right corner.

By clicking on Submit button, Form must be validated and should be highlighted with red border color with validation message at the down of each incorrect input fields in the form.

By clicking on Edit Button on the row, Application has to take to next page with the url (<http://localhost/employee/edit>.......) with router params and display clicked employee information in edit form same as “Add” Employee form. The same validation rule should applied here as well.

*Bonus: If User made some changes and moving away from this page, need a warning stating that “Form has been modified. You will lose your unsaved changes. Are you sure you want to close this form?” By clicking on “OK”, has to take to destination page where he was intended to go. Otherwise stay in the same page.*

Delete: By clicking on Delete button, Confirmation popup should appear. Summary page should be refreshed.

For CRUD operation, you can use localStorage.

*Bonus: Make use of redux saga library to handle side effects*

What we are looking for

1. Applicant’s ability to code, code readability, folder structure and reusability of the components.
2. CSS coding style. Try to reduce and use your own custom style even when using the css framework
3. Ability to use different react libraries such as Redux, Router, Redux Form
4. Redux State management
5. Applicants are encouraged to use best practices

You may upload the technical test to Github and provide us the link, or you may zip it up and send it over via email. Please ensure that your code is able to compile and able to run.

All the best!