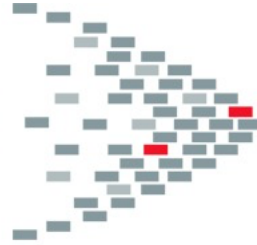


SA SOFTWARE



GET/technical\_assessment/

```
{  
  "project"      : Full Stack Consultant  
  "author"       : Sa Software  
  "version"      : v1.0  
}
```

# Technical Assessment

## / Objective

Your challenge is to build out the frontend and backend components of an employee application.

## / Brief

Your task is to build out the project to the design files provided in the `/assets` folder. The functionality outlined in **\*\*Expected Behaviour\*\*** is more important than implementing the designs pixel perfect. You are only supposed to build out the web version of the assignment and it does not need to be responsive.

## / Tasks

Your users should be able to:

- > Create, read, update, and delete employees
- > Create corresponding API endpoints
- > No authentication/session management is required. Imagine you're building this application for a single user (yourself)
- > Search employee by first name, last name, email

BONUS:

- > Receive form validations when trying to create/edit an employee
- > Filter employees by: Year (date of birth), and skills
- > Keep track of any changes, even after refreshing the browser (`localStorage` could be used for this)
- > State management
- > Unit tests

## / Expected behaviour

- > Creating an employee
  - When creating a new employee, an ID needs to be created. Each ID should be 2 random uppercased letters followed by 4 random numbers.
- > Editing an employee
  - When saving changes to an employee, all fields are required when the "Save Changes" button is clicked. If the user clicks "Cancel", any unsaved changes should be reset.
- > Adding skills
  - When adding skills, you should be able to add multiple.
- > Feel free not to add custom styling for the date and dropdown form fields. The designs for those fields are optional extras and are mostly for illustration purposes.

## / Evaluation criteria

- > Show us your work through your commit history
- > We're looking for you to produce working code, with enough room to demonstrate how to structure components in a small program
- > Completeness: did you complete the features?
- > Correctness: does the functionality act in sensible, thought-out ways?
- > Maintainability: is it written in a clean, maintainable way?
- > Testing: is the system adequately tested?

## / Code submit

Please organize, design, test, and document your code as if it were going into production - then push your changes to the master branch. After you have pushed your code, you may submit the assignment by sending us the link to the repo on Github.