

# Frontend Technical Test

## Introduction

Build an Employee Management System that allows users to view, add, edit, and delete employee information. The application should provide a user-friendly interface for managing employee details, ensuring a smooth experience while adhering to the specified requirements. You may utilize Ant Design or Material-UI as your CSS framework. For CRUD operations, you can choose to use either local storage or any available free MOCK API.

## User Stories

### User Story 1: Employee List Management

As a user, I want to view a list of employees, so that I can see all employee details at a glance.

#### Acceptance Criteria:

1. The homepage displays a table of employees with the following attributes:
    - First name
    - Last name
    - Email address
    - Phone number
    - Gender
    - Date of Birth
    - Joined Date
    - Edit/Delete buttons on each row
  2. Clicking the 'Add Employee' button navigates to the add employee page.
- 

### User Story 2: Adding an Employee

As a user, I want to add a new employee, so that I can keep the employee list updated.

#### Acceptance Criteria:

1. The add employee page contains fields for:
    - First name (Minimum 6 characters, maximum 10 characters validation)
    - Last name (Minimum 6 characters, maximum 10 characters validation)
    - Email address (Must be a valid email format)
    - Phone number (Must validate as a Singapore phone number)
    - Gender (Radio Button Group)
    - Date of Birth (Date Picker)
    - Joined Date (Date Picker) (Joined date must be after Date of Birth. Highlight the field with an error message if it does not.)
  2. Clicking the "Submit" button validates the form and highlights incorrect fields with a red border, displaying validation messages below each field.
- 

### User Story 3: Editing an Employee

As a user, I want to edit an existing employee's details, so that I can update their information as needed.

#### Acceptance Criteria:

1. By clicking on the Edit button on the row, the application navigates to the next page with the URL (<http://localhost/employee/edit>.....) with router params and displays the clicked employee's information in the edit form, the same as the "Add" Employee form.
  2. The same validation rules apply as in the add employee form.
  3. If the user attempts to navigate away with unsaved changes, a warning is displayed: "Form has been modified. You will lose your unsaved changes. Are you sure you want to close this form?" By clicking on "OK," the user is taken to the destination page they intended to go. Otherwise, they stay on the same page.
-

### User Story 4: Deleting an Employee

As a user, I want to delete an employee, so that I can remove outdated or incorrect entries.

#### Acceptance Criteria:

1. By clicking the Delete button, a confirmation popup is shown.
  2. Upon confirmation, the employee is deleted, and the summary page is refreshed.
- 

### Error Handling

#### Acceptance Criteria:

1. If any API call fails (e.g., during add/edit/delete), an appropriate error message is displayed to the user.

### Testing Requirements

- **Unit Tests:** Candidates should write unit tests for key components and functions.
- **E2E Tests:** Bonus points for candidates who include E2E tests.

### Documentation

Candidates must provide documentation on how to run the project, including setup instructions and any assumptions made during development.

### Evaluation Criteria

We are looking for candidates who demonstrate the following skills and qualities:

- **Technical Proficiency:**
  - **React Proficiency:** Experience with React libraries, including Redux Toolkit for state management, React Router for navigation, and React Hook Form for handling forms.
  - **CSS Expertise:** A solid understanding of CSS is essential. While using a CSS framework is acceptable, we encourage candidates to prioritize custom styles.
  - **Code Quality:** Proficiency in coding with a strong emphasis on readability, maintainable folder structure, and reusable components.
  - **Scalable Solutions:** Ability to design and implement solutions that can grow and adapt over time, ensuring long-term viability and performance.
- **Design and User Experience:**
  - **User-Centric Design:** Consideration of user needs to deliver the best UI/UX solutions, focusing on intuitive and effective design.
- **Problem-Solving Skills:** Ability to think critically and creatively when approaching challenges.
- **Best Practices:** Adherence to industry best practices, including code organization, documentation, and testing.

You can upload the technical test to GitHub and provide us with the link, or you can zip it and email it to [erichai@gic.com.sg](mailto:erichai@gic.com.sg) and [sekarchelliah@gic.com.sg](mailto:sekarchelliah@gic.com.sg).

All the best!