Assignment - Employee Management System UI

Objective:

Build an Employee Management System Dashboard using React and tailwindcss or tailwindcss based template. The dashboard will display employee data, allow the user to add new employees, and filter or search through existing employees.

Requirements:

Employee Data Structure: you can expect following response for each user from server:

Unset

```
{
   name:””,email:””,position:””,department:””,salary:””,start_date:””
}
```

UI Components

- The UI should be responsive.
- Create the following pages:
    - A page to show all employees with option to filter and sort results
    - A page to create and edit employees

State Management

- Use React's state and props effectively to manage the list of employees and the input from the Add Employee Form.
- Optionally, use React Context API or Redux for state management, especially if you plan to scale the application.

Validation and Error Handling

- Validation of fields on Add Employee Form is important. Make sure to implement good validation rules.
- Display appropriate error messages for invalid inputs.

Interactivity and Dynamic Data

- Implement add, delete, and search/filter functionality. The employee list should update in real-time based on user actions.
- (Optional) Use local storage to persist employee data or mock an API using JSON-server for a more realistic data management experience.

Documentation

- Include a README file with instructions on how to set up and run your application, as well as an overview of its features. You can use any package manager of your choice.

Submission Guidelines:

- Push your code to a Private GitHub repository and share it with Github users @**devbro1 and @maulmota**.
- Ensure your application is well-organized and code is clean and commented.
- Deadline: Within 10 days of starting the assignment.

Evaluation Criteria:

- Functionality: All features work as described.
- Code Quality: Clean, modular, and well-commented code.
- Design: Use of Bootstrap for a visually appealing and responsive design.
- Best Practices: Usage of React and Bootstrap best practices.