Detailed instructions how to use the software

Using the PulseHR - Employee Management System - Detailed Instructions:

Launch the Spring Boot Backend:

- Open your preferred IDE and run the Spring Boot application.
- Confirm successful startup by checking the console logs for any errors.

Initialize the Database:

- Access the H2 database console at http://localhost:3001/
- Use the JDBC URL jdbc:h2: mem:testdb, and login with the credentials specified in the application.properties file.
- Execute SQL scripts or manually insert sample data if needed.

Launch the React Frontend:

- Open a new terminal and navigate to the React app directory: cd employee-management-react.
- Install dependencies: npm install.
- Run the React app: npm start.
- The frontend development server will start, and the app will be accessible at http://localhost:3000.

View Employee List:

- Open your web browser and go to http://localhost:3000
- The Employee List page should display existing employee records fetched from the Spring Boot backend.

Add a New Employee:

- Click the "Add Employee" button to navigate to the Add Employee page.
- Fill in the required details in the form (e.g., first name, last name, email) and click "Save".
- The new employee should be added to the list, and you'll be redirected to the Employee List page.

Edit an Employee:

- On the Employee List page, click the "Edit" button next to an employee.
- Modify the employee details and click "Save".
- The changes should be updated in the employee list.\

Delete an Employee:

- On the Employee List page, click the "Delete" button next to an employee.

- Confirm the deletion.
- The employee should be removed from the list.

Test API Calls:

- Open the browser's developer tools
- Navigate to the Network tab.
- Perform CRUD operations on the UI and observe the corresponding API calls made to the Spring Boot backend.

Additional Features:

- Explore additional features such as searching for employees or implementing pagination.
- Intentionally provide incorrect data to test error handling and validation.

Deployment (Optional):

- If deploying to a server or cloud platform, build the React app for production: npm run build.
- Deploy the Spring Boot backend and the React frontend according to your chosen hosting solution.
- Update API URLs in the React app if the backend is hosted on a different server.