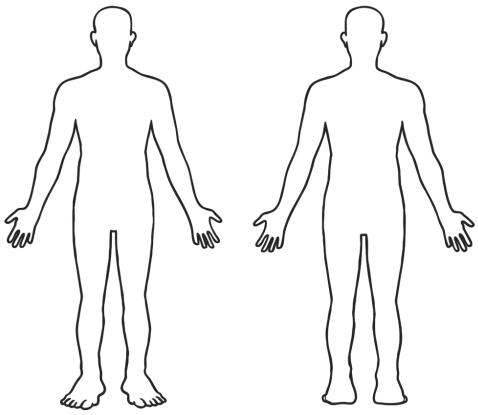
Lief - WebDev Engineer Task

What you need to build: An Injury tracking system

It is a web application that can be used by an organization (such as the police) to easily record and track the injuries reported by a person

Required Features:

1. Report: Users should be able to create, view, update, and delete an injury report. Each injury report should have the following
    a. Name of reporter
    b. Date & time of injury
    c. Body Map: Use should be able to encircle different areas of injury on a body map image such as below. The system should automatically number label them e.g. if the user encircles left hand and left foot, each circle should automatically be labelled 1 & 2



Front Back

    d. List of injuries: For each area encircled the user should be able to give details of the injury e.g. based on previous example the user should get two boxes labelled 1 & 2 to give details of the injuries on the left hand and left foot respectively

2. List of Reports: The user should be able to see a list / table of all the injuries reported
    a. the user should see the name, date time of injury and date of report
    b. The user should be able to sort by any of the fields mentioned above
    c. The user should be able to search by name
    d. The user should be able to filter by start and end date of datetime of injury and date time of report

3. User Authentication: Users should be able to register for an account on the app using a username and password, with options for Google login and email login. Authentication should be implemented using Auth0. Once registered, users should be able to log in and log out of their account, and view a history of their tasks.

4. UI/UX: The application should have a clean and user-friendly interface, built using Grommet or Ant as the design library, that is responsive and works well on desktop and mobile devices. It should be visually appealing and easy to use.

We understand that there may be a lot of possibilities for the UX beyond described, but we encourage you to go with an MVP version for each feature. For any clarifications, feel free to reach out to us at career@lief.care.

Bonus features:

If you're done with the above and would like to take this app a step further, here's some bonus stuff you can try for brownie points.

5. Progressive Web App (PWA): The entire application could be built as a fully responsive Progressive Web App (PWA) that can be installed on a user's home screen, works offline, and provides an app-like experience across different devices and platforms.

6. Automatic location detection: When the user encircles an area of injury the label could be automatically detected instead of just a number e.g. in the previously mentioned example instead of 1 & 2 the system could automatically label them 'left hand' and 'left foot'. These labels need not be anatomically precise or accurate for the sake of this task.

7. Analytics Dashboard: analytics dashboard that provides visualizations of relevant metrics.

Recommended Tech-stack:

- Front-end: Next.js with Grommet or Ant as the design library. Please use React Context for state management, and not Redux or any other state management

library.

- Back-end: GraphQL with Prisma as the ORM for database connectivity. You can use any database.
- Authentication: Auth0 for user authentication, with options for Google login and email login.
- Analytics: Use any relevant libraries or frameworks for data visualization, such as Chart.js or D3.js.
- Progressive Web App (PWA): Implement service workers and other necessary PWA features to build the application as a fully responsive PWA.

Please note that these are just recommendations based on the tech stack we use at Lief. Feel free to use any other technology if you wish.

Submission Guidelines:

- There is no hard deadline for the submission, however, the sooner the better. We encourage submissions within 10 days of receiving the task.
- Host your app on a relevant hosting site (Netlify, Vercel, Heroku etc), and use this form to submit the URL.
- We expect code and resources for the task to have optimal quality and performance, and ***not*** be sent to us as a zip file/google drive link/pigeon transported flash drive (Although bonus points **if** you manage to do the last one).
- Provide thorough documentation for the codebase, including a Readme file with an overview of the application and structure of your codebase. Please note that this documentation is meant for us as reviewers of your task, and not for general users of the app. Make sure to include what all you attempted from the list of features above.