You work for the technology department of a large financial company, WealthHealth.

This company uses an internal web application, called HRnet that manages employee records. The application is old and uses jQuery on the front end, which leads to considerable bugs and increased internal complaints. Your development team has been working on upgrading the application for some time now.

One morning you go to work and notice the following messages from Jade, a fellow developer on the team:

Jade: Hey, you! Hope you had a good weekend! Did you hear the news? Management is FINALLY going to let us upgrade HRnet and convert it to React! No more maintaining old jQuery code! Youhouuu!!! 🎉😁

You: Oh, really? So good! The day is finally here! 😱💃

Jade: Yes, it's so exciting! And since you've been working your butt off and doing amazing work lately, you'll be one of the first developers to start converting the code! Congratulations!

You: That's awesome! I can't wait to get in there and (finally) solve this problem! HRnet alone must account for about 90% of our total tech debt, lol 😂

Jade: Tell me about it! I figured I'd send you some information to help you get started. As I'm sure you know, the biggest issues for HRnet users right now are date pickers, modal windows, drop down menus, and tables. We have received several complaints that these jQuery plugins are very slow.

So we want to create our own React components instead of these third-party jQuery plugins that are used in the user interface. And we hope that converting these jQuery plugins into React components will improve the performance and stability.

You: Okay, got it. All those network calls and all that DOM manipulation required for jQuery plugins add up and slow things down. And it will be nice to start having our own React components to have more control over the application.

Jade: Exactly! Management doesn't want this first part of the conversion process to take too long, so please pick ONE of the four jQuery plugins and convert it to a React component. (I'll send them to you later today). You'll need to create a separate GitHub repo for the converted code.

You: Of course! It's a good way to get an initial version of the new app faster, and we can always iterate on it later.

Jade: Exactly! To manually test your converted React components, you'll also need to convert the main HRnet app into a React app. Since you know that jQuery and React don't go well together, so we want the new version of the app to be 100% React and 0% jQuery. We don't want to create some kind of jQuery/React monster!

You: Got it. Thanks for pointing that out, I was already starting to tremble at the thought of creating a React/jQuery hybrid 😱 Jade: No 🙅🏽♂️🙅🏽♂️We really don't want this ahah

I'll send you the current HRnet repo so you can get started. We'll also need you to do some performance testing and provide us with reports.

You: Good call! I'll be sure to include those performance reports in my deliverables.

Jade: Also, keep in mind that we want to follow a functional programming paradigm when writing these libraries in React, so avoid using classes when you convert it. And try to write smaller, modular pieces of code and standalone functions for optimal modularity and maintainability.

Also, make sure to document your converted React component with a general description of what the component does and comments explaining what each accessory is for and how it is used. The rest of the team will thank you 😉

You: Alright, I'll take care of that!

Jade: Perfecto! That should be all you need to get started - I'll also send you a recap email later. If you have any questions or need any other details or resources, feel free to let me know. Let's get rid of that jQuery code! 🎉

You: I'm on the case! You can count on me, Jade! 😁

As promised, a little later in the day, Jade sends you the following email :

Subject: Details on the redesign project

From: Jade

To: Me

Hi there!!!

I can't wait for you to get started on this project! Here's a recap of the main tasks :

Convert the entire HRNet project to React.

Convert one of the four current jQuery plugins to React. Replace the 3 remaining jQuery plugins with React components.

Perform Lighthouse performance tests comparing the old and new application.

You can find all the key information for each task below:

**HRNet project conversion**

Here is the current HRnet repo. Don't forget that the whole HRNet application has to be converted to React :

You'll have to make a new version of the "Create Employee" and "Employee List" pages with React.

You will have to add a state management system (the current version uses local storage).

You also have to make sure that everything is consistent in style. You don't have to redesign the application, but if you want to change the style to something more modern, you are welcome to do so.

If you have time, you can test the React code with a unit test suite. Otherwise, only manual tests are needed.

**Plugin conversion**

For plugins, feel free to consult the Issues tab to have more context on the problems users encounter with existing jQuery plugins: issue on date selectors, issue on modal windows, issue on dropdown menus, and issue on tables.

Here is the list of currently used jQuery plugins that need to be converted:

Date picker plugin

Modal window plugin - jQuery.modal.js

Drop down menus

Plugin for data tables

Also, when converting a jQuery plugin to a React component, keep in mind to convert only the code that deals with the actual functionality of the plugin's user interface. For example, if a jQuery plugin includes AJAX code, you don't need to convert it. If you convert a jQuery plugin for a modal window, focus on creating a React component that works as a modal window, and nothing else.

**Performance testing**

We also want to measure quantifiable data (e.g. page load times, network calls) to ensure that converting the app to React actually improves performance. To do this, make sure you do Lighthouse performance audits. To compare, you'll need to do one for the current jQuery HRnet application, and then another once the application and the chosen jQuery plugin are converted to React.

That's all! Once the HRnet app in React is working, you can publish the React component to npm as a package and share the link so we can use it if needed. If you have trouble with npm, you can use GitHub packages as an alternative.

Also, let me know when everything is done and we'll do a code review to address any remaining comments.

Can't wait to see what you come up with!

Jade

That's it, you have all the information you need. Now the hard part will be choosing which jQuery plugin to convert!

**Deliverables**

A GitHub repo containing:

The source code of the selected jQuery plugin converted to ReactJS (only one plugin should be converted).

A link to the converted React library published on npm (or GitHub Packages) in the README file.

A second GitHub repo containing the source code of the HRnet application converted to React.

A LightHouse performance report of HRnet running before and after the library conversion, in JSON format.

The GitHub repo must be named with the following convention: FullName\_#\_DatedStart. The # corresponds to the project number on the path and the date must be in ddmmyyy format. For example, FrancoisLenotre\_12\_05032020.

**Presentation**

The presentation will be structured as follows, with the evaluator playing the role of Jade:

Presentation and explanation of deliverables (10 min)

Discussion (15 min)

The evaluator will ask you questions about your methodology and results.

Be prepared to defend your work, as the evaluator will question your decisions.

The evaluator will stop playing the role of Jade five minutes before the end of the session so that you can debrief together.

**Skills**

Redesign an application to reduce technical debt

Analyze the performance of a web application

Deploy a front-end application

Programming in JavaScript with functional programming