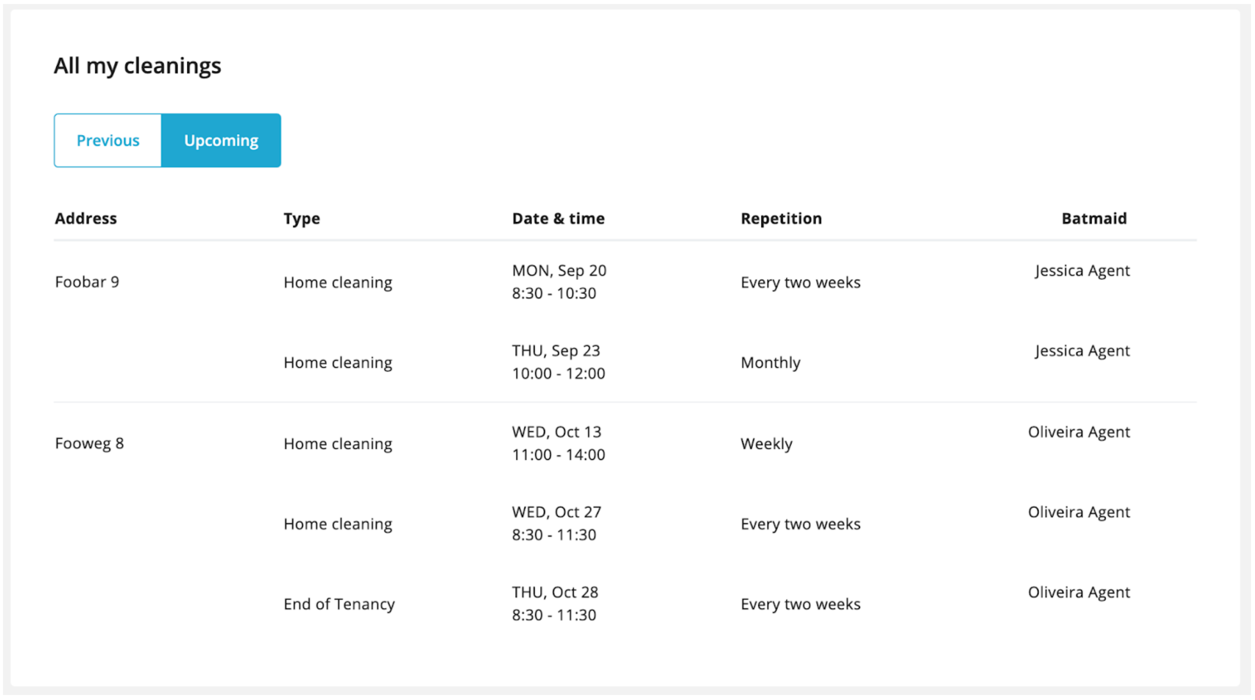
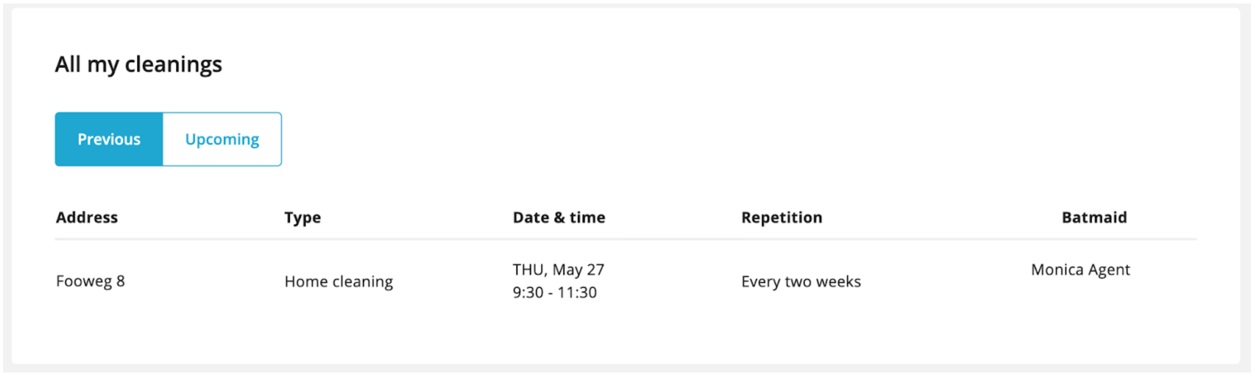
RECRUITMENT TASK

Your job is to build a simple dashboard that renders a list of cleanings sorted by location.
The dashboard consists of two main views - Upcoming cleanings and Previous Cleanings.
Users can transition between these views using the toggle buttons in the upper left corner. The rest of the dashboard doesn't have any additional interactions.
You should build this component using React, Redux & TypeScript.



Address	Type	Date & time	Repetition	Batmaid
Foobar 9	Home cleaning	MON, Sep 20 8:30 - 10:30	Every two weeks	Jessica Agent
	Home cleaning	THU, Sep 23 10:00 - 12:00	Monthly	Jessica Agent
Fooweg 8	Home cleaning	WED, Oct 13 11:00 - 14:00	Weekly	Oliveira Agent
	Home cleaning	WED, Oct 27 8:30 - 11:30	Every two weeks	Oliveira Agent
	End of Tenancy	THU, Oct 28 8:30 - 11:30	Every two weeks	Oliveira Agent



Address	Type	Date & time	Repetition	Batmaid
Fooweg 8	Home cleaning	THU, May 27 9:30 - 11:30	Every two weeks	Monica Agent

Rules:

1. You can create the project repo from scratch or use create-react-app
2. Please use the provided mocked data (mock.json)
3. Mock a fetch request and store that data in Redux
4. You can transform the data however you like, the goal is to have it in a shape that makes it easy to render it

5. The mocked data has some optional fields that you don’t have to display, we only want to see the details visible in the above designs/graphics
6. The cleanings should be divided into Previous and Upcoming based on current date (the mock data is prepared in such a way to ensure there is at least one previous cleaning)
7. You can style the components anyway you want - by using pure (S)CSS, a CSS library (like Bootstrap), StyledComponents or other libs.
8. Styling doesn’t have to be pixel-perfect, we only want to see the table rendered correctly.
9. We didn’t include any designs for the mobile view on purpose, you can ignore it (or just stack the table data vertically)
10. Please write some tests that ensure the component works correctly (integration, unit, UI, you name it). We don’t want 100% test coverage, just a few tests to see how you approach them :)
11. (Nice to have) Please include a fallback screen which will be shown when data fetching fails
12. (Nice to have) Create skeleton loaders to present a fetching process in the table instead of using spinners etc.
13. You can send this assignment by email or push it to a github repository
14. If you have any questions feel free to contact us
15. Good luck!