PROJECT SPECIFICATION

Employee Polls

Application Setup

CRITERIA	MEETS SPECIFICATIONS
Is the application easy to install and start?	The application requires only `npm install` and `npm start` to install and launch.
Does the application include README with clear installation and launch instructions?	A README is included with the project. The README includes a description and clear instructions for installing and launching the project.

Login Flow

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Does the application have a way to log in and log out? Does the application work correctly regardless of which person the user impersonates?	1. There should be a way for the user to impersonate/ log in as an existing user. (This could be as simple as having a login box that appears at the root of the application. The user could then select a name from the list of existing users.) 2. The application works correctly regardless of which user is selected. 3. The application allows the user to log out and log back in. The user should be logged in to submit new polling questions, vote, and view the leaderboard. 4. Once the user logs in, the home page is shown. 5. Whenever the user types something in the address bar, the user is asked to log in before the requested page is shown.

Application Functionality

CRITERIA	MEETS SPECIFICATIONS
Does the home page have the desired functionality?	1. The answered and unanswered polls are both available at the root. 2. The user can alternate between viewing answered and unanswered polls. 3. The unanswered questions are shown by default. 4. The name of the logged in user is visible on the page. 5. The user can navigate to the leaderboard. 6. The user can navigate to the form that allows the user to create a new poll.

CRITERIA	MEETS SPECIFICATIONS
Are the polling questions listed in the correct category (Unanswered vs Answered), and do they have the desired functionality on the home page?	1. Each polling question resides in the correct category. For example, if a question hasn't been answered by the current user, it should be in the "Unanswered" category. 2. A polling question links to details of that poll. 3. The polls in both categories are arranged from the most recently created (top) to the least recently created (bottom).

CRITERIA	MEETS SPECIFICATIONS

Are the details of each poll displayed with all of the relevant information?	1. The details of the poll are available at `questions/:question_id`.
	2. When a poll is clicked on the home page, the following is shown:
	  ◦ the text “Would You Rather”;
	  ◦ the avatar of the user who posted the polling question; and
	  ◦ the two options.
	3. For answered polls, each of the two options contains the following:
	  ◦ the text of the option;
	  ◦ the number of people who voted for that option;
	  ◦ the percentage of people who voted for that option.
	4. The option selected by the logged in user should be clearly marked.
	5. When the user is logged in, the details of the poll are shown. If the user is logged out, he/she is asked to log in before before being able to access the poll.
	6. The application asks the user to sign in and shows a 404 page if that poll does not exist. (In other words, if a user creates a poll and then the same or another user tries to access that poll by its url, the user should be asked to sign in and then be shown a 404 page. Please keep in mind that new polls will not be accessible at their url because of the way the backend is set up in this application.)

Does the voting mechanism work correctly?	1. Upon voting in a poll, all of the information of the answered poll is displayed. 2. The user’s response is recorded and is clearly visible on the poll details page. 3. When the user comes back to the home page, the polling question appears in the “Answered” column. 4. The voting mechanism works correctly, and the data on

CRITERIA	MEETS SPECIFICATIONS
	the leaderboard changes appropriately.
Can users add new polls?	1. The form is available at `/add`. 2. The application shows the text “Would You Rather” and has a form for creating two options. 3. Upon submitting the form, a new poll is created and the user is taken to the home page. 4. The new polling question appears in the correct category on the home page.

Does the leaderboard work correctly and have the desired functionality?	1. The Leaderboard is available at `/leaderboard`.
	2. Each entry on the leaderboard contains the following:
	  ◦ the user’s name;
	  ◦ the user’s avatar;
	  ◦ the number of questions the user asked; and
	  ◦ the number of questions the user answered.
	3. Users are ordered in descending order based on the sum of the number of questions they’ve answered and the number of questions they’ve asked.

Is the application navigable?	1. The app contains a navigation bar that is visible on all of the pages. 2. The user can navigate between the page for creating new polls, and the leaderboard page, and the home page without typing the address into the address bar.
Does the application interact with the backend correctly?	1. The data that’s initially displayed is populated correctly from the backend. 2. Each user’s answer and each new poll is correctly recorded on the backend.

CRITERIA	MEETS SPECIFICATIONS
Does the code run without errors? Is the code free of warnings that resulted from not following the best practices listed in the documentation, such as using `key` for list items? Is the code formatted properly?	The code runs without errors. There are no warnings that resulted from not following the best practices listed in the documentation, such as using `key` for list items. All code is functional and formatted properly.

Architecture

CRITERIA	MEETS SPECIFICATIONS
Does the store serve as the application's single source of truth?	The store is the application's source of truth. Components read the necessary state from the store; they do not have their own versions of the same state. There are no direct API calls in the components' lifecycle methods.

CRITERIA	MEETS SPECIFICATIONS
Is application state managed by Redux?	Most application state is managed by the Redux store. State-based props are mapped from the store rather than stored as component state. Form inputs and controlled components may have some state handled by the component.
Does application state update correctly?	Updates are triggered by dispatching action creators to reducers. Reducers and actions are written properly and correctly return updated state to the store.
Does the architecture of the application make sense?	The code is structured and organized in a logical way. Components are modular and reusable.

Unit Testing

CRITERIA	MEETS SPECIFICATIONS
Are jest, @testing-library/react, and @testing-library/jest-dom present in the project?	The package.json file should include the following under devDependencies: 1. jest 2. @testing-library/react

CRITERIA	3. @testing-library/jest-dom MEETS SPECIFICATIONS
Can all the unit test be run by entering the `npm start test` command?	The application requires only `npm start test` in order to run all the unit tests in the project.
Do all the unit tests pass?	After running `npm start test`, all unit tests should pass. There should be no failing tests.
Are there at least 10 unit tests?	The project requires a minimum of 10 passing unit tests.
Are There two unit tests written which test the _saveQuestion() function in _DATA.js?	The following two unit tests must be present for _saveQuestion(): 1. An async unit test to verify that the saved question is returned and all expected fields are populated when correctly formatted data is passed to the function. 2. An async unit test to verify that an error is returned if incorrect data is passed to the function.
Are there two unit tests written which test the _saveQuestionAnswer() function in _DATA.js?	The following two unit tests must be present for _saveQuestionAnswer(): 1. An async unit test to verify that true is returned when correctly formatted data is passed to the function. 2. An async unit test to verify that an error is returned if incorrect data is passed to the function.

CRITERIA	MEETS SPECIFICATIONS
Is there at least one snapshot test present?	At least one test must call the `toMatchSnapshot()` function from jest. Doing this will generate a folder called **snapshots** where the snapshot will be stored.
Is there is at least one DOM test which uses the fireEvent function?	At least one unit test must use the render method from @testing-library/react to render one of your React components. The unit test should then perform some actions on the component using fireEvent such as `fireEvent.click()` or `fireEvent.change()`. After calling fireEvent, call the `expect()` method from jest to verify that a change occurred in the UI after the event was fired.

Suggestions to Make Your Project Stand Out!

1. The person using your application should have a way of impersonating/logging in as an existing user. The minimum we suggest is simply having a dropdown where you can select a user from the list of existing users in order to be authenticated. However, if you really want your submission to stand out and behave more realistice to real-world use cases, try creating your own username/password login authentication flow. You could even allow users to create their own accounts!
2. The project requires a minium of ten unit tests; six of which apply to verify specific use cases. However, if you want to stand out, you could write even more! Unit testing is crucial in software development and can expose bugs in your code you may not have been aware of. Try to apply everything you learned during the **Testing with Jest** lesson and write meaningful unit tests that verify your code is working as you intended.