Problem Description

Build a javascript-based web or react-native application that does the following:

1. Authenticates a user using Auth0 (auth0.com). It's free to create an auth0 account and use it for authentication.
2. After a successful login, the application asks the user to fill out a form. In the real-world, this form specification would be supplied by the backend (and customized per-user), but for purposes of this exercise use the attached JSON object as the form specification (filename *form.json*). Each form has a number of questions/fields and the page each field needs to be on. The order of the fields is as it is in the JSON object (so the page number of the last field is the total number of pages in the form). Please support the following features:
    a. Allow the user to move back (and forward) between the form pages. The user can only move to the next page after all questions on the current page have been filled-in (including explanation boxes if relevant).
    b. Allow the user to hand-draw their signature and save it into the filled out form as a base64 encoded PNG image (feel free to use a canvas library that will help you do this)
    c. For information on field names to use for user-input, look at the attached sample filled-out form (filename *filled_form.json*). Write the filled out JSON form to the console and pop-up a modal saying "Thank you for filling out this form". Logout the user (and redirect them back to the login screen) once the modal is dismissed.

Extra Credit

1. When a user fills out a form, store the filled out form locally on-device so if the same user logs back in, they're able to view/edit their answers.

Other Considerations

1. Think about how you'd make the frontend testable. Also add a few unit-tests.
2. Use your best judgment for styling / layout of the form and usability features (e.g. adding a way to clear the signature field would be helpful to the user)
3. Using Typescript would be preferable over plain Javascript, but if you're unfamiliar with Typescript, it is okay to use plain Javascript.
4. Please state any assumptions you're making, and feel free to reach out if you have any questions
5. If you're looking to prioritize, please note that displaying, filling the form and printing it out to the console is higher-priority.

JSON Form Notes

1. `ffField` is a question/description followed by a plain text field input
2. `slField` is a question/description followed by a few radio options
3. `nestedSlField` is a question/description followed by a bunch of slFields
4. `signField` is used to collect the user signature