Rapptr Labs React Test

Thank you for taking time to take our coding test! This test will be used to evaluate your candidacy for a web engineering role at Rapptr Labs. Please make sure to follow the instructions closely. If there is any doubt on a requirement, please reach out for clarification. The test is used to measure your coding abilities as well as your attention to detail. The project is not intended to take a significant amount of time. The actual duration can vary based on skill level or experience. In order to accommodate volatile schedules, the test is not timed, but we do ask that you complete the test within seven days of receiving it.

Project Overview

The project is a simple to-do app where a user can login, manage a list of to-do's, and logout. The assessment should be done in React. You are free to use any library to help you for the exception of UI component libraries, such as Bootstrap, material-ui, and semantic-ui. Style libraries that still require the use of CSS are fine so you are free to use styled-components, emotion, sass, less, etc. We want to see your knowledge of CSS

You are free to use a state management library if you wish, but the use of a state management library does not affect your grade. If you use React state or use the React component lifecycle, please use React hooks. Do not use class components.

In the requirement section, there will be mockups of the app to illustrate the look and feel. These are just low-fidelity mockups. Please implement your own styles to make the app look aesthetically pleasing. Just make sure that the designs don't violate any of the requirements below.

The mockups show the app on a tall screen. Assume the app can be viewable on devices as small as an iPhone 12 Pro and as large as a desktop. There is no need to go crazy on mobile optimization. We don't grade heavily on that part. The content's container should be centered horizontally.

Test Submission

When you complete the test, please upload it to GitHub as a public repo and send us a link. For the repo name, do not include the following words: *Rapptr, Labs, or Test*. Please also include a small explanation of how you spent your time on the project and how long each piece took.

Success Criteria

We are looking for a few critical pieces:

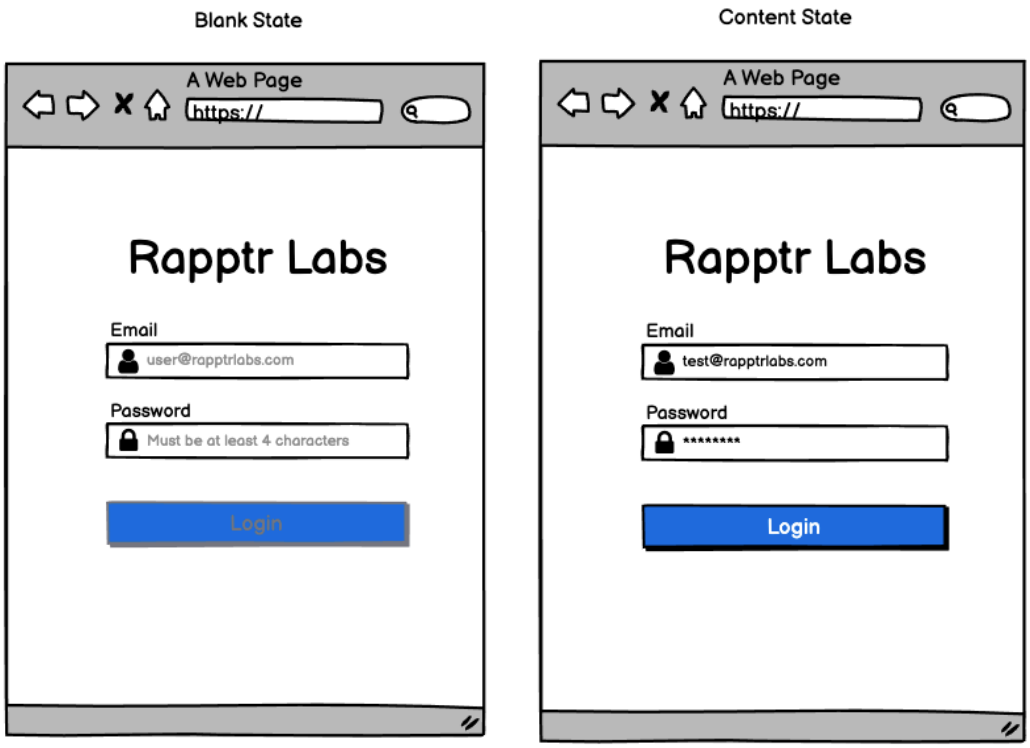
1. A solid understanding of React and web principles.
2. Clean and maintainable code.
3. A well thought-out solution.

Project Requirements

There are two pages: Login and List.

The Login page

The login page contains a simple login form. See requirements below.



Error State 1

Error State 2

1. A title should appear as “Rapptr Labs”. It should be centered horizontally regardless of the page dimensions.
2. Login form
    a. The login form should contain two fields: Email and Password.
    b. The form should validate onChange.
    c. The form should submit when the user clicks on the login button OR presses enter while focused on the email or password field.
    d. While the form is being submitted, the Login button should not be clickable.
    e. If the form has not passed validation, the form cannot be submitted and the button’s background and text should be at half opacity.
    f. Email field:
        i. A title exists labeled “Password” as per the mockup
        ii. Email should have a persistence account icon that should always appear on the left side of the input.
        iii. The text content should always come after the icon.
        iv. The field should use a placeholder for “user@rapptrlabs.com”.
        v. If content exists, the placeholder should not appear (standard for forms).
        vi. The content should be a valid email address and should be no more than 50 characters.
        vii. If the email has a validation error, the error should be displayed below the input field and the border of said field should turn red.
            1. An error should not appear if the field has not yet been touched.
    g. Password field:
        i. A title exists labeled “Password” as per the mockup

ii. Password should have a persistence lock icon that should always appear on the left side of the input.
iii. The text content should always come after the icon.
iv. The field should use a placeholder for “Must be at least 4 characters”.
v. If content exists, the placeholder should not appear (standard for forms).
vi. The content should be at least 4 characters and should be no more than 16 characters.
vii. If the password has a validation error, the error should be displayed below the input field and the border of said field should turn red.
    1. An error should not appear if the field has not yet been touched.

3. Form Submission
    a. When the form has been successfully submitted passing all validation errors, call the following API: http://dev.rapptrlabs.com/Tests/scripts/user-login.php
        i. This API endpoint needs to be called with a POST request.
        ii. It takes two body params: email and password.
        iii. For testing purposes, use the following email and password: test@rapptrlabs.com / Test123.
        iv. A successful response will look like the following:

```
{
   "user_id": 16,
   "user_email": "test@rapptrlabs.com",
   "user_username": "testuser",
   "user_is_active": 1,
   "user_profile_image": "http://dev.rapptrlabs.com/Tests/images/taylor_avatar.png",
   "user_last_active_epoch": 1544680026,
   "user_creation_epoch": 1544713200,
   "user_is_new": 1,
   "user_token": "6dd4737a8b7ec61313ae5e900420d46815e1d13b2902be71b97a8fbf1f421a3e"
}
```

        v. *Note: In a normal environment, the user_token would be used to make authenticated requests, but that is outside the scope of this test.*

The List page

This list page is just a simple to-do list. There will be no API calls with the list page, but the content should be persisted locally (see requirement below).

Normal State

Edit State

1. The list displays current to-do items.
2. Clicking on the new button adds a new to-do form with a save button (see edit state in mockup).
3. The to-do text needs to have at least one character and at most 25 characters.
4. Clicking save adds the item to the list and changes the item to the normal state.
5. Clicking on the pencil icon changes the item into an edit state.
6. Clicking on the trash icon removes the item from the list.
7. Typing something in the search bar filters the list, case insensitive.
8. The list can have an unlimited number of entries.
9. There is no special empty state. An empty list is fine.
10. Persist the data between sessions (local storage is fine).
11. A logout button exists on the top right. Clicking the logout button will log the user out and return him or her to the login screen.