



Front-end Test

Goal

Welcome to the StarOfService test. If you have this in your hand (or in all possibilities on your desktop) then BRAVO! Your profile is top notch and you have been chosen for our Front-end test.

Let me walk you through the mini-app you are supposed to build. The Goal is to make a signup module which consists of 3 steps as shown in the designs below:

The image shows three mobile app screens for a signup process. The first screen, titled 'Signup', has a blue progress bar at the top and a red error message 'EMAIL IS REQUIRED'. It contains three input fields: 'EMAIL', 'PASSWORD', and 'CONFIRM PASSWORD'. The second screen, also titled 'Signup', has a blue progress bar and contains three sections: 'DATE OF BIRTH' with three input fields for 'DD', 'MM', and 'YYYY'; 'GENDER' with three radio buttons for 'MALE', 'FEMALE', and 'UNSPECIFIED'; and 'WHERE DID YOU HEAR ABOUT IS?' with a dropdown menu. The third screen, titled 'Thank you!', has a blue progress bar and a large green checkmark. It contains a button labeled 'Go to Dashboard →'. Each screen has a 'Next →' button at the bottom right, except for the third screen which has a 'Back' button at the bottom left.

Take time to carefully read through the specs below, as you will be judged by their completeness.

Specs

1. You are expected to use React + Redux to build the forms.
2. User should be allowed to go to next step only if all fields on current step are valid.
3. There is a blue progress meter on top that shows current progress of steps.
4. When moving from step to step, the page should not reload. This is a Single Page Application.
5. Style the app exactly as given in the screens above. Though, you are encouraged to suggest improvements to the form in a note along with your test.
6. You are free to use bleeding edge CSS features like Flexbox to make your work easy.
7. You need to implement following validations:
 - 7.1. Email should be required.
 - 7.2. Email should be a valid email address. Use regex validation.
 - 7.3. Password is required.
 - 7.4. Password should be minimum 6 characters long.
 - 7.5. Password confirmation should match the password.
 - 7.6. All fields in "Date of birth" are required.
 - 7.7. All fields in "Date of birth" should be valid respectively.
 - 7.7.1. DD should be a number and $31 \geq DD \geq 1$.
 - 7.7.2. MM should be a number and $12 \geq MM \geq 1$.
 - 7.7.3. YYYY should be a number.
 - 7.7.4. Dates like 30/2/1991 are invalid.
 - 7.8. The user must be 18 year old or more.
 - 7.9. One gender option must be selected from the 3 given.

7.10. “Where did you hear about us?” is optional.

8. When a field is invalid, it should show a useful message in the label itself, red in colour.
9. The “Go to Dashboard” button at the last step should print all the details entered as a JSON in the Browser console, something like:
 - 9.1. `{ “user_data”: { “email”: ..., “password”: ..., “date_of_birth”: 1485761262, “gender”: “female”, “how_hear_about_us”: null } }`

Now before you begin don’t forget to check out the Extras below, as completing these gives you an extra edge ahead of other candidates.

Extras

Completing the above specs are enough to say that you’ve completed your test. Although, if you want to be adventurous and edge ahead of other candidates you might want to also:

- Write unit tests for React components and reducers.
- Add animations between step transitions and to the progress bar.
- Implement Server side rendering of the form.
- Use redux-form to implement the forms and validation.

Finally

We encourage best practices in our code. And we will do so in our tests too. We expect to see:

- Clean code and conforming best practices.
- Good and easy to follow file structure.
- Giving preference to simpler solution in favour of smart but complicated solution to a problem.

We hope you have a great time building this mini app. If you have any questions about the test, feel free to contact us and we’ll be glad to help.