

Exercise: Adding form validation

Overview

Now that all the features are implemented, it is important to provide a good user experience. Part of a good user experience is preventing the user from submitting invalid data. This has the added benefit of helping to ensure data integrity when it is submitted to the API.

Scenario

In this exercise, you will update the booking form to use client-side validation. You will achieve this using both HTML5 validation and validation through React. You can recap concepts related to client-side form validation by revisiting the following lesson items:

- [HTML and CSS in-depth: Forms and validation](#)
- [HTML and CSS in-depth: Making the most out of client-side validation](#)
- [React basics: Common event handling](#)
- [Advanced React: Creating a form component in React](#)

Instructions

Step 1: Implement HTML5 validation

- For each form input field, consider what HTML5 validation can be applied. For example, the user should not be able to reserve a table for less than one person.
- Apply the validation to each form input field in `BookingForm`.

Step 2: Implement client-side validation using React

After applying the HTML5 validation, consider which input fields could use further validation via JavaScript and React. For example, the submit button should be disabled if the form input fields are invalid.

Tip: Use events to detect field changes and submit button clicks.

Tip: If you built your form using `Formik`, take advantage of Formik's built-in validation functions.

Step 3: Run the web app

Run the web app and verify that the form validation behaves as expected. When all of the fields are valid, it should be possible to submit the form.

Conclusion

In this exercise, you added client-side form validation to improve the user experience and help ensure valid data is submitted to the API. Customers reserving a table using the website will now have a more satisfactory experience when

using the Little Lemon website.