Core | Deadline: Sunday of Week 3 | Difficulty Level: Intermediate | Est. Time: 03:00-04:00

Reactive Forms (Core)

Create a dynamic React form by utilizing every skill learned so far.

Learning Objectives

- Apply state and conditional rendering techniques to display error messages when specific form fields are not meeting the required criteria.
- Design using a reactive approach to store and update the values of form inputs in real-time as users interact with the form.

Welcome to another **Core assignment**! Some students like to explore the assignments before they're finished reading through the lessons, and that's okay! It can be good for your brain to have a preview of what your future challenges might be. However, before you begin this assignment, it's important that you've first:

- Completed the preceding lesson modules
- Taken the knowledge checks to confirm your understanding
- Viewed lecture material related to the assignment topics
- Completed and submitted your practice assignments

Reactive Forms:

This assignment is designed to test your working knowledge of React at this point in the course.

To meet the base requirements, you must fulfill the assignment criteria below.

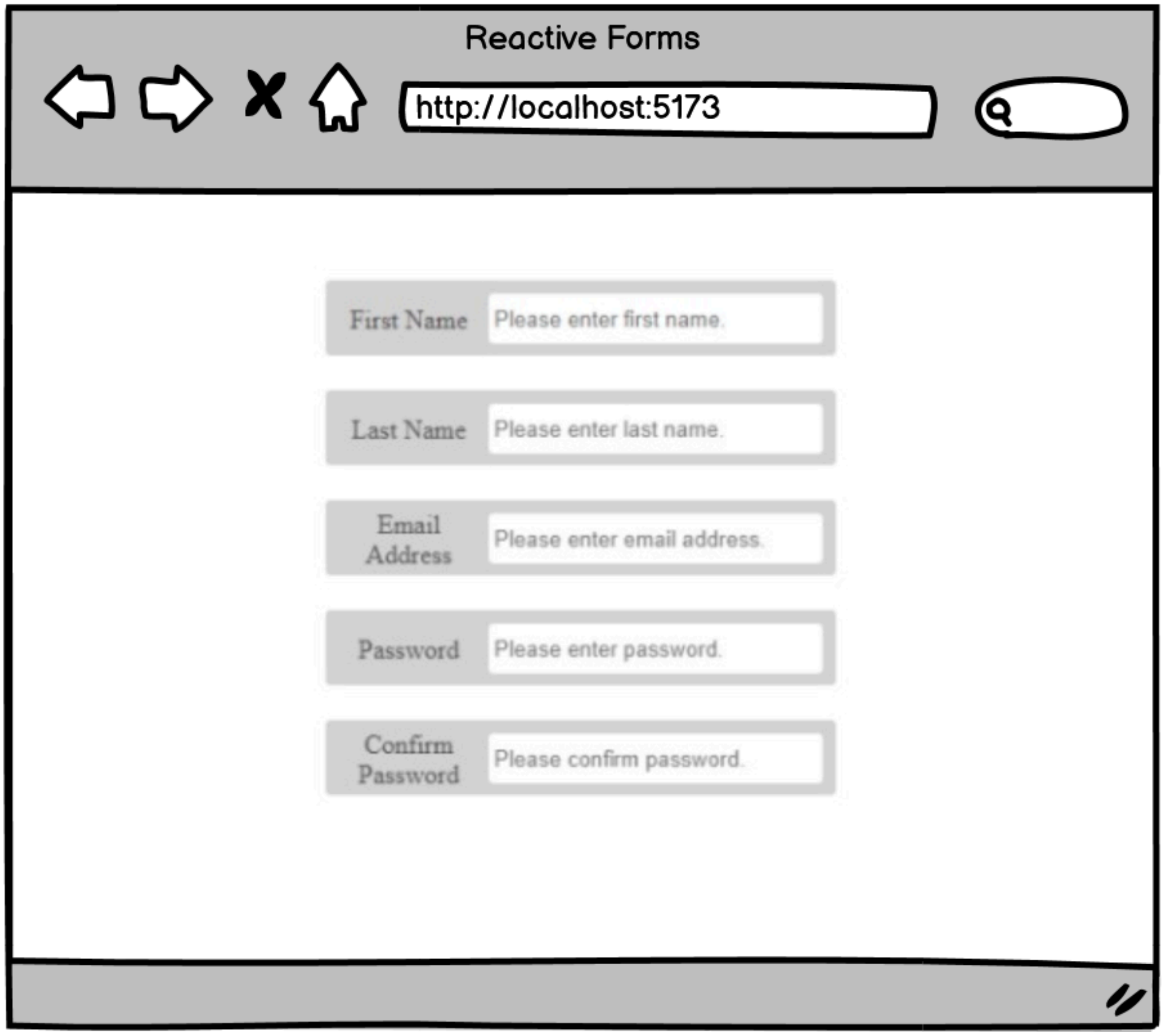
Create a React Form with error messages.

1. If the First Name or Last Name is less than 2 characters, output an error message saying that field must be at least 2 characters.
2. The e-mail is less than 8 characters, say the field must be at least 8 characters.
3. The password must match the contents of the confirm password field and should be at least 8 characters.

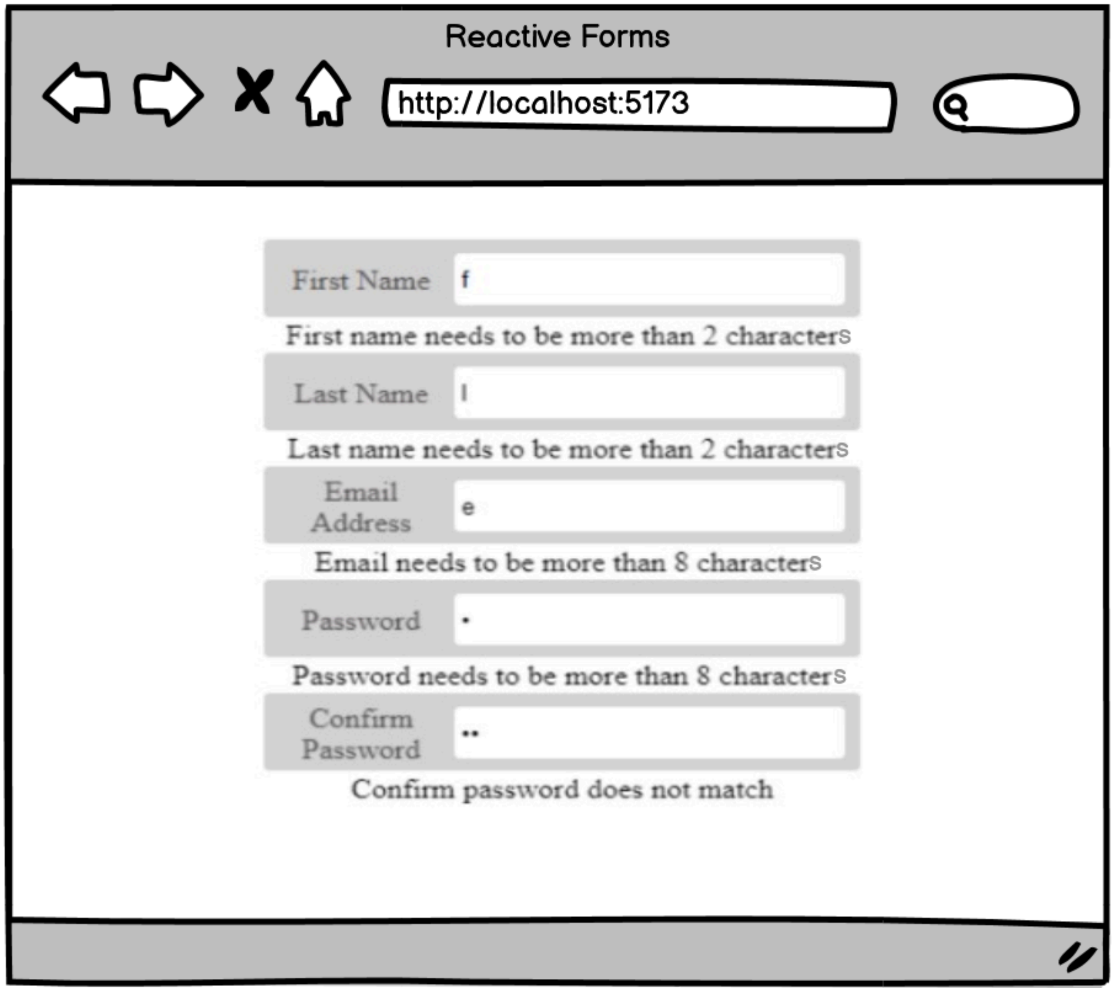
These messages can appear right when you render the page, and will disappear in real time.

State and conditional rendering will come in handy during this assignment.

Your finished product should look like the following wireframe before any text is entered:



When the validations are triggered, your work should resemble the following:



First Name f

First name needs to be more than 2 characters

Last Name l

Last name needs to be more than 2 characters

Email Address e

Email needs to be more than 8 characters

Password •

Password needs to be more than 8 characters

Confirm Password ••

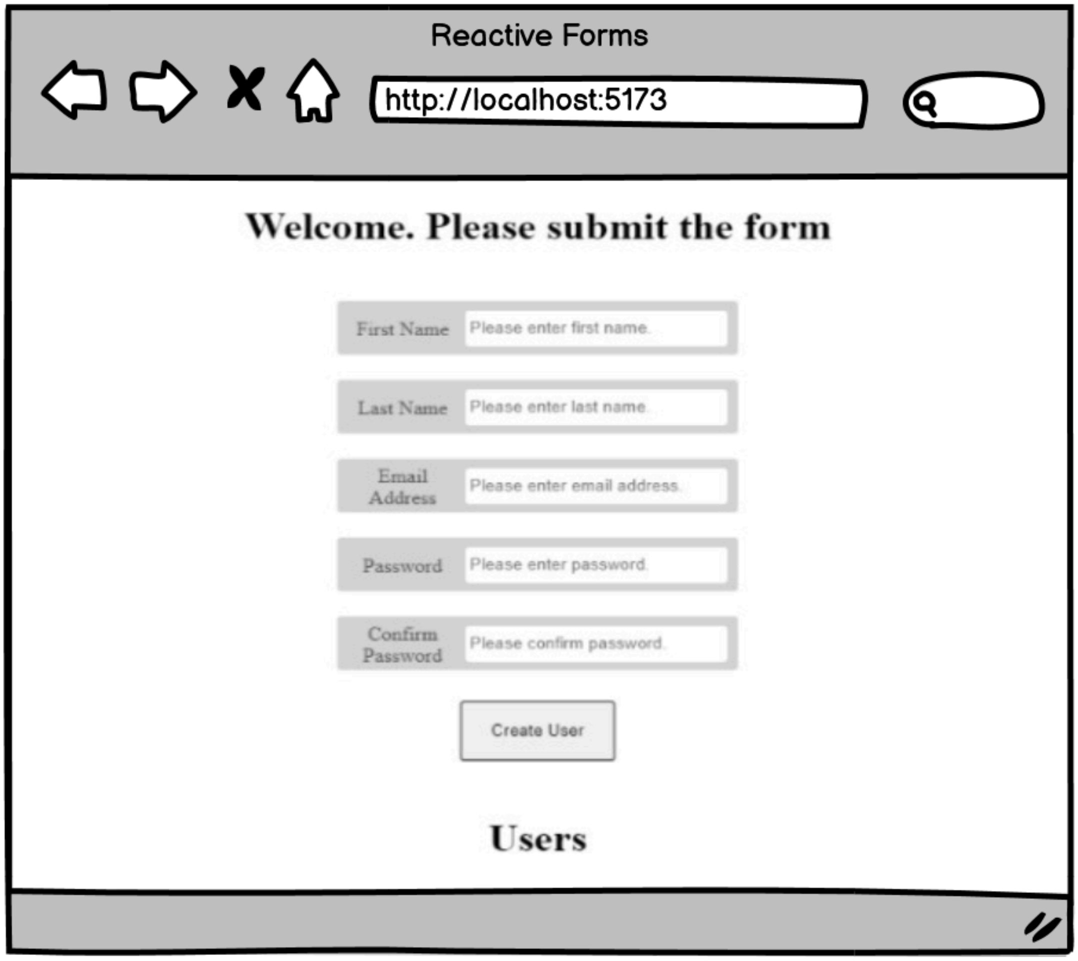
Confirm password does not match

Ninja Forms

To truly put your skills to the test, you can attempt to replicate the following wireframe.

This form is not just for show as the following additional features have been added on top of the previous wireframe:

- A button to add the user input data to a list
- Inputs that clear when the Create User button is clicked
- A display section for each user added
- A page header that dynamically changes once the form is submitted



Reactive Forms

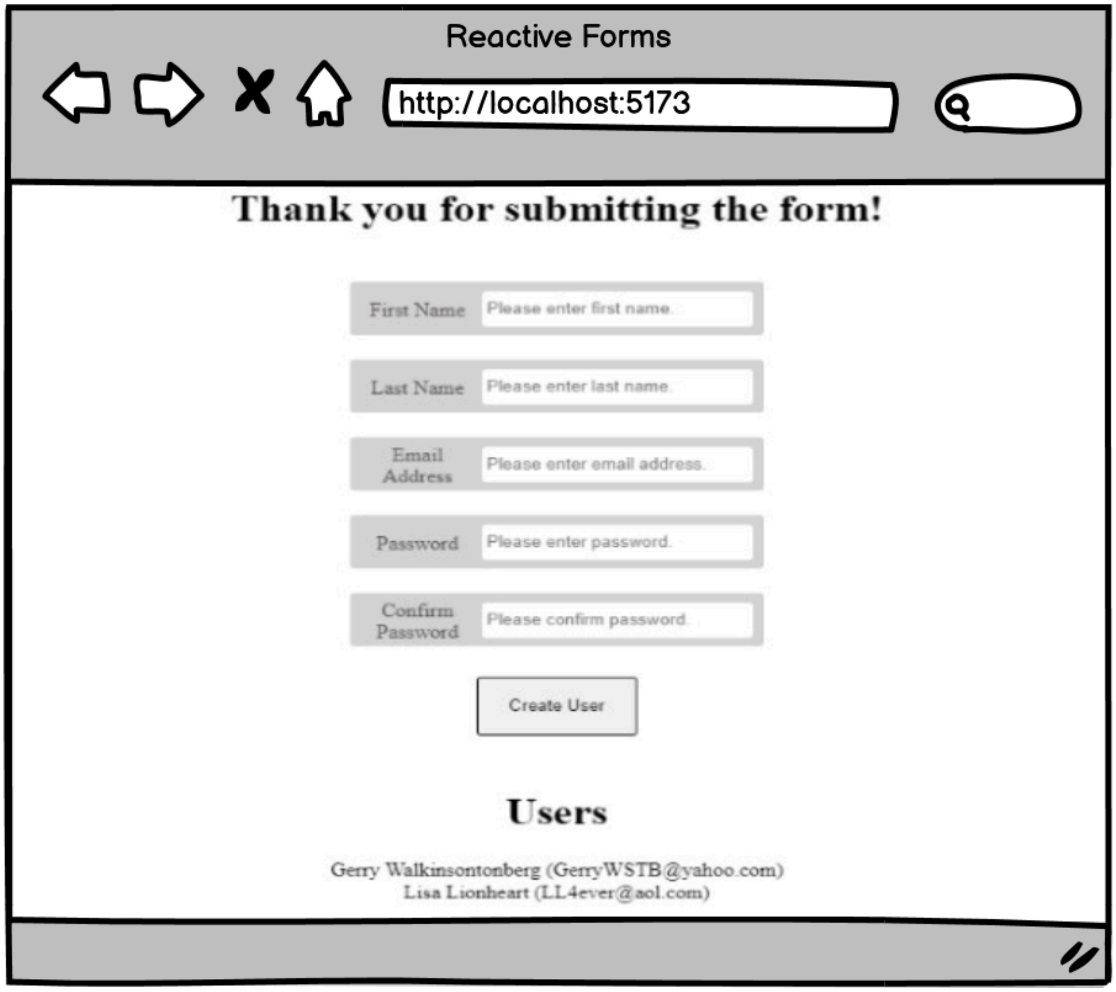
http://localhost:5173

Welcome. Please submit the form

First Name	Please enter first name.
Last Name	Please enter last name.
Email Address	Please enter email address.
Password	Please enter password.
Confirm Password	Please confirm password.

Users

Your assignment should resemble the following after adding a couple of users:



Thank you for submitting the form!

Users

Gerry Walkinsontonberg (GerryWSTB@yahoo.com)
Lisa Lionheart (LL4ever@aol.com)

- □ Create validations for First Name
- □ Create validations for Last Name
- □ Create validations for Email
- □ Create validations for password and confirm password
- □ Only show the validations if the input is not blank (You should not see the error messages if you have not typed anything in)
- □ Ninja Bonus Section:
- □ Create a button that adds the user input data to a list
- □ Clear the inputs after submission

- □ Display each user added using the map method

- □ Changes the app's header after the form has been submitted