Assignment Title: Creating a Registration Form with Validation using React JS Functional Components

Objective: The objective of this assignment is to help you understand how to create a registration form with validation using React JS functional components and Bootstrap CSS. You will learn how to create a form that takes in user input, validates it, and displays error messages when necessary.

Instructions:

1. Create a new React JS project using your preferred development environment and setup.
2. Create a new functional component named "RegistrationForm" that displays a form with several input fields for the user to fill out.
3. Implement the form logic using React state with the "useState" hook. Store the user input as state values in the component.
4. Validate the form inputs using the "useEffect" hook. Display error messages for invalid inputs.
5. Add styling to the form and error messages using Bootstrap CSS to make it look presentable.

Requirements:

1. Use React functional components to implement the RegistrationForm with the "useState" and "useEffect" hooks for state and validation.
2. Use Bootstrap CSS to style the form and error messages.
3. Add validation to the form inputs to ensure that the user enters valid data.
4. Submit the completed project as a ZIP file, including all project files.

Bonus:

1. Add custom validation functions to validate specific input fields.
2. Add additional input fields to the form and validate them accordingly.

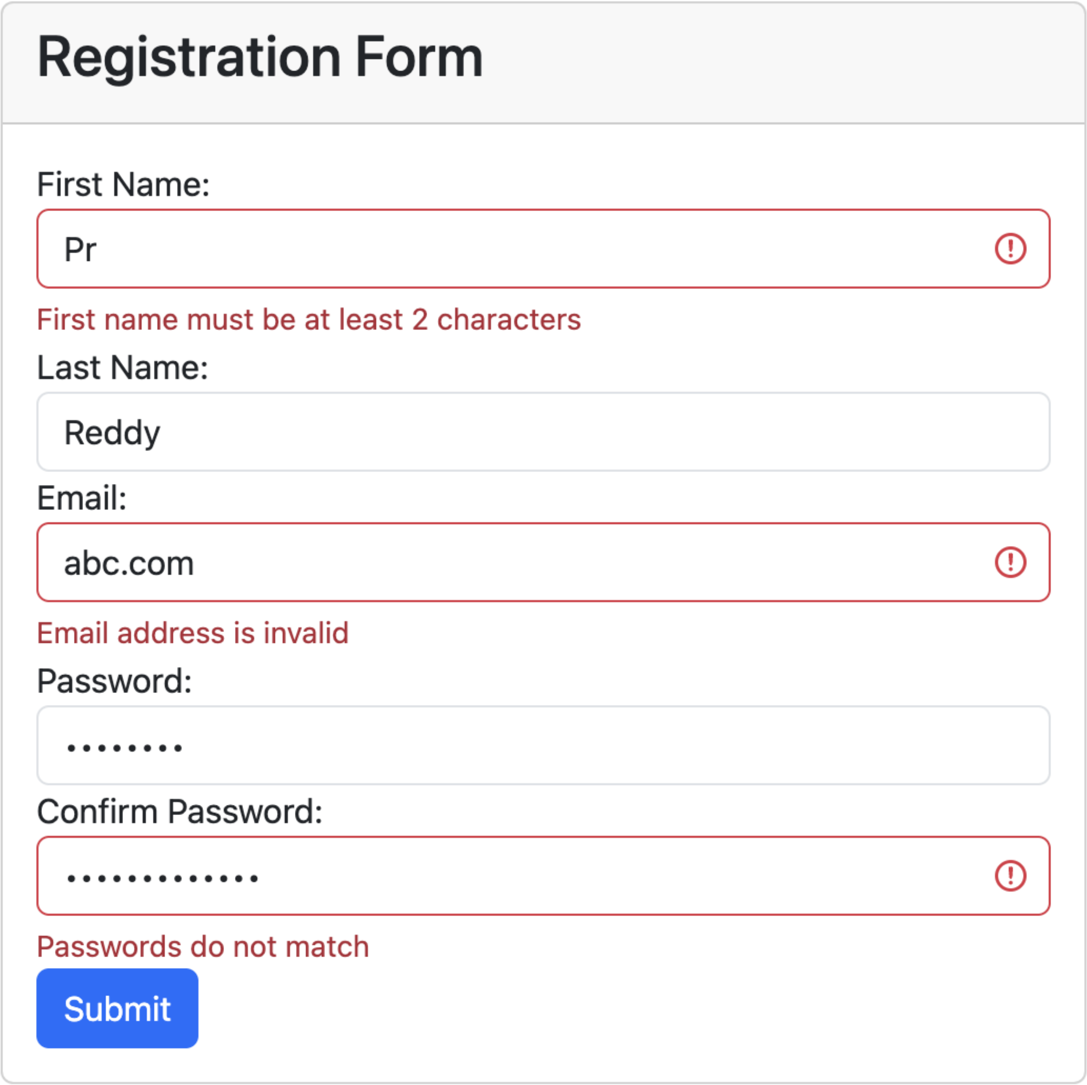
Tips:

1. Use console.log statements to debug and troubleshoot issues.
2. Refer to the Bootstrap documentation for styling and class names.

Sample template code

```
import React, { useState, useEffect } from 'react';
import './RegistrationForm.css';
import 'bootstrap/dist/css/bootstrap.min.css';

function RegistrationForm() {
  const [firstName, setFirstName] = useState('');
  const [lastName, setLastName] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [confirmPassword, setConfirmPassword] = useState('');
  const [firstNameError, setFirstNameError] = useState('');
  const [lastNameError, setLastNameError] = useState('');
  const [emailError, setEmailError] = useState('');
  const [passwordError, setPasswordError] = useState('');
  const [confirmPasswordError, setConfirmPasswordError] = useState('');

  useEffect(() => {
    // TODO: Implement form validation logic
  }, [firstName, lastName, email, password, confirmPassword]);

  const handleSubmit = (event) => {
    event.preventDefault();
    // TODO: Implement form submission logic
  };

  return (
    <div className="RegistrationForm container mt-5">
      <div className="card">
        <div className="card-header">
          <h2>Registration Form</h2>
        </div>
        <div className="card-body">
          <form onSubmit={handleSubmit}>
            <div className="form-group">
              <label htmlFor="firstName">First Name:</label>
              <input type="text" className="form-control" id="firstName" value={firstName}
onChange={(event) => setFirstName(event.target.value)} />
              {firstNameError && <div className="invalid-feedback">{firstNameError}</div>}
            </div>
            <div className="form-group">
              <label htmlFor="lastName">Last Name:</label>
              <input type="text" className="form-control" id="lastName" value={lastName}
onChange={(event) => setLastName(event.target.value)} />
              {lastNameError && <div className="invalid-feedback">{lastNameError}</div>}
            </div>
            <div className="form-group">
              <label htmlFor="email">Email:</label>
              <input type="email" className="form-control" id="email" value={email}
onChange={(event) => setEmail(event.target.value)} />
              {emailError && <div className="invalid-feedback">{emailError}</div>}
            </div>
            <div className="form-group">
              <label htmlFor="password">Password:</label>
              <input type="password" className="form-control" id="password"
value={password} onChange={(event) => setPassword(event.target.value)} />
              {passwordError && <div className="invalid-feedback">{passwordError}</div>}
            </div>
            <div className="form-group">
              <label htmlFor="confirmPassword">Confirm Password:</label>
              <input type="password" className="form-control" id="confirmPassword"
value={confirmPassword} onChange={(event) => setConfirmPassword(event.target.value)}
/>
              {confirmPasswordError && <div className="invalid-
feedback">{confirmPasswordError}</div>}
            </div>
            <button type="submit" className="btn btn-primary">Submit</button>
          </form>
        </div>
      </div>
    </div>
  );
}

export default RegistrationForm;
```

Sample output



Registration Form

First Name:

Pr

First name must be at least 2 characters

Last Name:

Reddy

Email:

abc.com

Email address is invalid

Password:

•••••••••

Confirm Password:

••••••••••••••

Passwords do not match