4.1

App.js

import React from 'react';

import ClickCounter from './ClickCounter'; // or LoginForm from './LoginForm'

function App() {

  return (

    <div className="App">

      <ClickCounter />

    </div>

  );

}

export default App;

// ClickCounter.js

// ClickCounter.js

import React, { useState } from 'react';

export default function ClickCounter() {

  const [count, setCount] = useState(0);

  return (

    <div>

      <p>Count: {count}</p>

      <button onClick={() => setCount(count + 1)}>Increment</button>

    </div>

  );

}

\src\ClickCounter.test.js

// ClickCounter.test.js

import React from 'react';

import { render, screen, fireEvent } from '@testing-library/react';

import ClickCounter from './ClickCounter';

test('renders initial count as 0', () => {

  render(<ClickCounter />);

  expect(screen.getByText(/Count: 0/i)).toBeInTheDocument();

});

test('increments count by 1 on button click', () => {

  render(<ClickCounter />);

  fireEvent.click(screen.getByText(/Increment/i));

  expect(screen.getByText(/Count: 1/i)).toBeInTheDocument();

});

test('increments correctly on multiple clicks', () => {

  render(<ClickCounter />);

  const button = screen.getByText(/Increment/i);

  fireEvent.click(button);

  fireEvent.click(button);

  fireEvent.click(button);

  expect(screen.getByText(/Count: 3/i)).toBeInTheDocument();

});

4.2

App.js

import React from 'react';

import LoginForm from './LoginForm';

function App() {

  return (

    <div className="App">

      <LoginForm />

    </div>

  );

}

export default App;

// LoginForm.js

import React, { useState } from 'react';

export default function LoginForm() {

  const [form, setForm] = useState({ email: '', password: '' });

  const [errors, setErrors] = useState({});

  const [success, setSuccess] = useState('');

  const handleChange = (e) => {

    setForm({ ...form, [e.target.name]: e.target.value });

    setSuccess('');

  };

  const handleSubmit = (e) => {

    e.preventDefault();

    const errs = {};

    if (!form.email.trim()) errs.email = 'Email is required';

    if (!form.password.trim()) errs.password = 'Password is required';

    setErrors(errs);

    if (Object.keys(errs).length === 0) {

      setSuccess('Login successful!');

    }

  };

  return (

    <form onSubmit={handleSubmit}>

      <input

        name="email"

        placeholder="Email"

        value={form.email}

        onChange={handleChange}

      />

      <div style={{ color: 'red' }}>{errors.email}</div>

      <input

        name="password"

        type="password"

        placeholder="Password"

        value={form.password}

        onChange={handleChange}

      />

      <div style={{ color: 'red' }}>{errors.password}</div>

      <button type="submit">Login</button>

      {success && <p style={{ color: 'green' }}>{success}</p>}

    </form>

  );

}

// LoginForm.test.js

import React from 'react';

import { render, screen, fireEvent } from '@testing-library/react';

import LoginForm from './LoginForm';

test('renders email and password input fields', () => {

  render(<LoginForm />);

  expect(screen.getByPlaceholderText(/Email/i)).toBeInTheDocument();

  expect(screen.getByPlaceholderText(/Password/i)).toBeInTheDocument();

});

test('displays validation errors on empty submit', () => {

  render(<LoginForm />);

  fireEvent.click(screen.getByText(/Login/i));

  expect(screen.getByText(/Email is required/i)).toBeInTheDocument();

  expect(screen.getByText(/Password is required/i)).toBeInTheDocument();

});

test('submits form with valid data and shows success message', () => {

  render(<LoginForm />);

  fireEvent.change(screen.getByPlaceholderText(/Email/i), {

    target: { value: 'user@example.com' },

  });

  fireEvent.change(screen.getByPlaceholderText(/Password/i), {

    target: { value: 'password123' },

  });

  fireEvent.click(screen.getByText(/Login/i));

  expect(screen.queryByText(/Email is required/i)).not.toBeInTheDocument();

  expect(screen.queryByText(/Password is required/i)).not.toBeInTheDocument();

  expect(screen.getByText(/Login successful/i)).toBeInTheDocument();

});