Test Writing Process:

1. Pick one out Component to test all by itself.
2. Make a test file for the component if one does not exist
3. Decide what the important parts of the Component are
4. Write a test to make sure each parts works as expected
5. Run tests at the command line.

Test Code Format:

* First Import the Packages of jest
* Second import the component which we want to test
* Use a test function method to test the case

Sample:

Import {render,screen} from ‘@testing-library/react; package import

Import User from ‘@testing-library/user-event’; package import

Import userForm from ‘./useForm’; component path

test(‘statement’,method call);

test(‘it shows two input and button’,()=>{

//render the component

//Manipulate the component or find an element

//Assertion – Make sure the component is doing what we expect to do..

});  
  
ARIA Role:

1. ARIA Roles clarify the purpose of an HTML element
2. Traditionally used by screen readers-softwares to help people understanfd the content on the screen
3. Many HTML elements have an ‘implicit ’ ,or automatically assigned, role.
4. Elements can be assigned manually assigned a role.Even trained engineer do this incorrectly.

Heading =>h1 to h6 tags

List => ul and li

Button =>button

Link => **a** tag

Textbox => input , type=”text”

Matchers from React Testing Library:

1. Expect(element).toBeInTheDocument() => Make sure element is present on the page
2. Expect(element).toBeEnabled() =>Makes sure an element (like an input) is not disabled
3. Expect(element).toHaveClass() =>Makes sure an element has a class name
4. Expect(element).toHaveTextContent() =>Make sure an element has some particular text
5. Expect(element).toHaveClass() => Make sure an input,select or textarea has a value.