React JS Training

(Intermediate to Advanced)

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Schedule for React JS Training

Day#	Date	Topic
Day-1	16-Apr-2024	State Management : Class Components
Day-2	17-Apr-2024	State Management : Functional Components, Hooks
Day-3	18-Apr-2024	Http Client Programming – AJAX Calls to APIs (Node JS)
Day-4	19-Apr-2024	Working with Forms, Validations, Services
Day-5	22-Apr-2024	Redux – State Management Library
Day-6	23-Apr-2024	Routing – SPA in React JS
Day-7	24-Apr-2024	Unit Testing React Application

Duration : 7 days (2hours per day) ; 2:00 pm to 4:00 pm; 16th Apr – 24th Apr

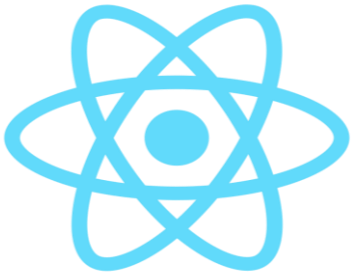
Day-7
Unit Testing React Components

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Index – Day-7



Introduction to Unit Testing

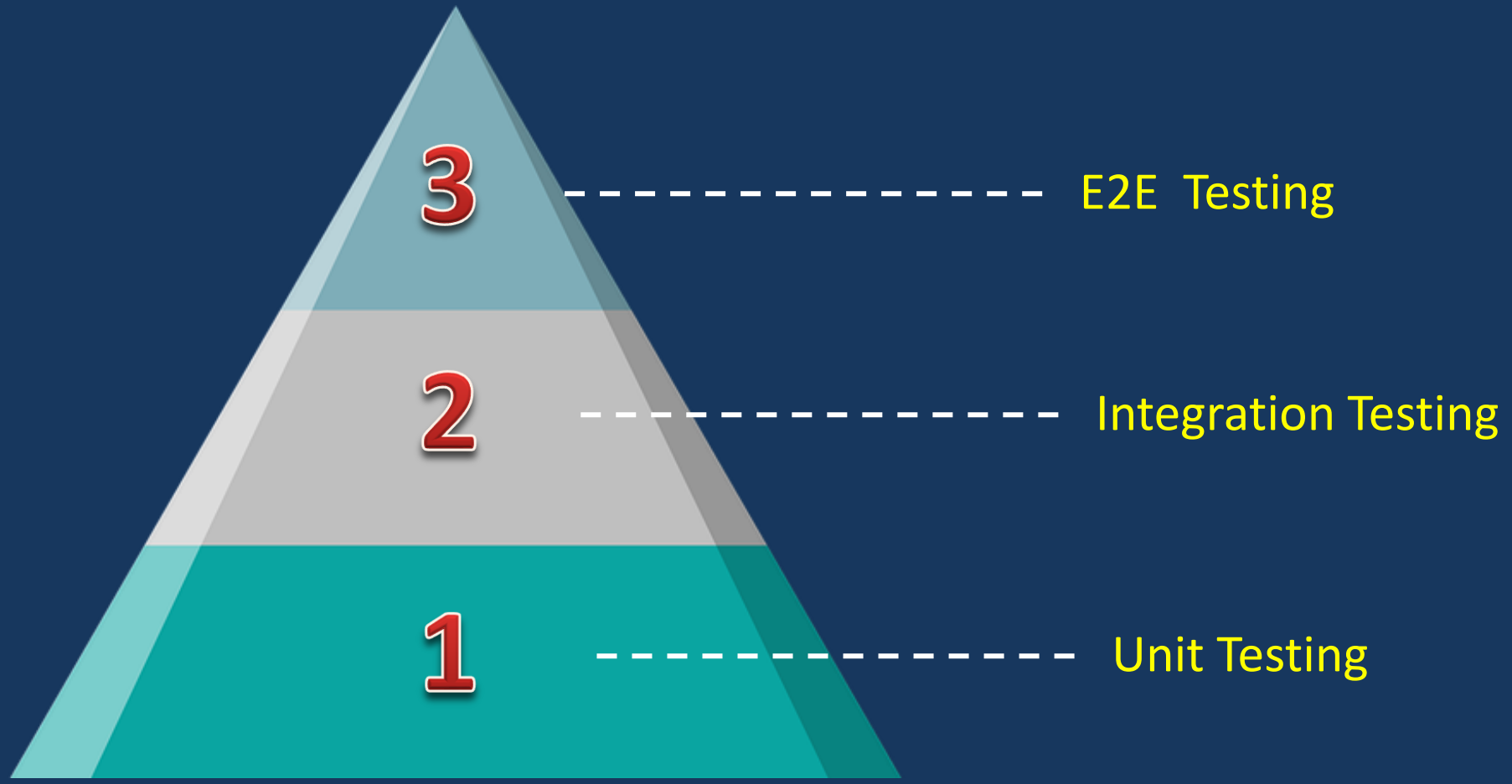
Narasimha

Sr. IT Trainer/Consultant

What is Unit Testing?

- Unit can be the smallest part of an application that is testable.
- Eg: individual function, method, etc.
- Software developers are the ones who write the unit test cases.
- The aim here is to match the requirements and the unit’s expected behavior.
- Performed before Integration testing
- Finding issues/bugs at an early stage

Levels of Testing



AAA Pattern in Unit Testing

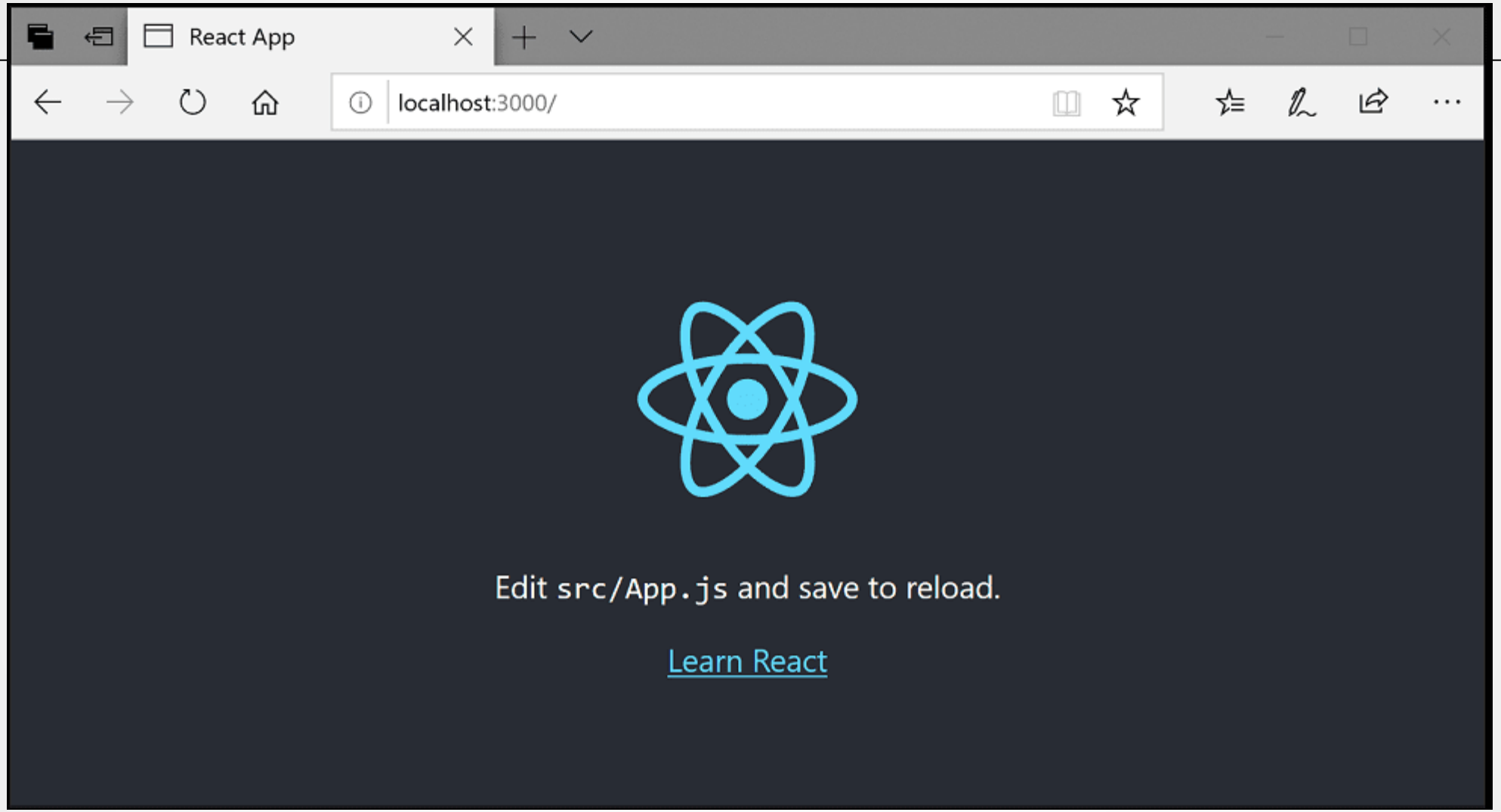
AAA Pattern in Unit Testing

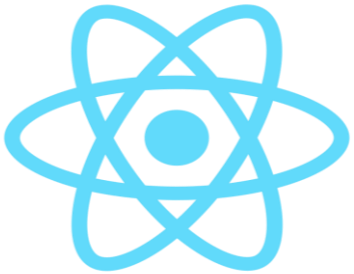
1. **Arrange**: This is the first step of a unit test application. Here we will arrange the test, in other words we will do the necessary setup of the test.
2. **Act**: This is the middle step of a unit test. In this step we will execute the test i.e. we will do the actual unit testing and the result will be obtained.
3. **Assert**: This is the last step of a unit test application. In this step we will check and verify the returned result with expected results.

Environment Setup

1. **Jest**
2. **React Testing Library**
3. **Enzyme**

- npm i --save enzyme
- npm i --save enzyme-adapter-react-17-updated

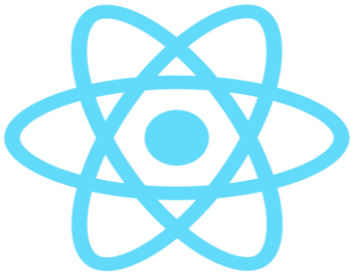

React App
localhost:3000/
Edit src/App.js and save to reload.
Learn React



Testing with Jest and Enzyme

Narasimha

Sr. IT Trainer/Consultant



Creating Test Cases, Test Suites

Narasimha

Sr. IT Trainer/Consultant

Testing React Components

1. Test --- Load the components without crashing
2. Test --- the DOM elements, input
3. Test --- the getting and setting the values to input --- state
4. Test --- the event handling

Sample Test Cases

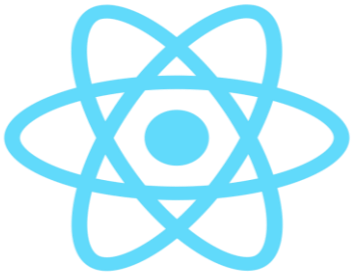
```
import Enzyme, {shallow} from 'enzyme’;

Enzyme.configure({ adapter : new Adapter() });

it('should render component without crashing', () => {
    const wrapper = shallow(<Login />);
    expect(wrapper.find("input").length).toBe(3);
});
```

Important statements for Unit Testing

```
1. let str  = wrapper.find("#t1").props().value;
2. wrapper.find("#b1").simulate("click");
3. wrapper.find("#t1").simulate("change", { target : { value :
   "Stephen"} });
4. let str  = wrapper.find("#p1").text();
```

Practice Hands-Ons

Narasimha
Sr. Corporate Trainer, Mentor
tnrao.trainer@gmail.com



Application Testing using Playwright

Narasimha

Sr. IT Trainer/Consultant

What is Playwright?

- Playwright is an open-source test automation library initially developed by **Microsoft** contributors.
- It supports programming languages like Java, Python, C#, and **JavaScript**.
- Playwright comes with Apache 2.0 License and is most popular with NodeJS with Javascript/Typescript.
- Playwright's first release was in **January 2020**, and it has gained much popularity ever since.

Playwright Component Testing

- Playwright Test was created specifically to accommodate the needs of **end-to-end testing**.
- Playwright supports all modern rendering engines including Chromium, WebKit, and Firefox.
- Test on Windows, Linux, and macOS, native mobile emulation of Google Chrome for Android and Mobile Safari.

Playwright Component Testing

- Playwright has features that support component testing with some popular web frameworks like React.
- Playwright is built to enable cross-browser web automation that is evergreen, capable, reliable, and fast.

```
npm init playwright@latest
```

Advantages of Playwright

1. Easy Setup and Configuration
2. Multi-Browser Support
3. Multi-Language Support
4. Built-in Reporters
5. Debugging Tools Support

Running Tests

- npx playwright test
- npx playwright test --project=chromium
- npx playwright test --debug
- npx playwright test one.spec.js
- npx playwright show-report

Sample Test Cases

```
const { test, expect } = require('@playwright/test’);

test.beforeEach(async ({ page }) => {
  await page.goto('http://localhost:3000/');
});


test('has title "React App"', async ({ page }) => {
   // Expect a title "to contain" a substring.
  await expect(page).toHaveTitle("React App");
});
```

Important Statements

1. Get the elements by placeholder

```
const txtUserId = page.getByPlaceholder('User Id');
```

2. Fill text fileds with values

```
await txtUserId.fill("admin");
```

3. Access the input field value

```
let str  = await page.locator("input#t1").inputValue() ;
```

4. Perform button click event

```
await page.locator("input#b1").click();
```

5. Get innerText of elements

```
let str = await page.locator("p#p1").innerText();
```

Accessing Input Fields and Fill the values

```
  // Fill user id textbox with "admin" value
test('fill user id textbox', async ({ page }) => {
  const txtUserId = page.getByPlaceholder('User Id');
  await txtUserId.fill("scott");
  let str  = await page.locator("input#t1").inputValue() ;
  expect(str).toBe(“scott”);
});
```

Practice Hands-Ons

Narasimha
Sr. Corporate Trainer, Mentor
tnrao.trainer@gmail.com

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com