Git commands:-

1. git status
2. git add React.docx
3. git commit -m "first\_react\_app creation with component creation"
4. git push

**React**

* React is a front-end library.
* It is developed by Facebook.
* Unlike Angular JS, which is MVC(Model View Controller), it focus on View part only but not on model and controller. For model and controller, we need Node JS , Spring Boot,etc.
* It creates re-useable components.
* React is all about component. You need to think everything as a component.
* The virtual DOM makes react fast.

**Pre-requisites:-**

* **Solid understandings of HTML, CSS, Java Script is required.**

**Features:-**

1. JSX:- JavaScript syntax extension.

**Limitations:-**

It covers only View layer. So, we need to choose another technology for complete development.

**NOTE: It runs on Node, So Node JS is needed to be installed.**

**To install React App globally:**

npm install –g create-react-app –save

Where npm = node package manager

Create-react-app will create your app with all the necessary files needed.

To start React app:

We use npm start

When we create React App, By default it is created with required packages under node\_modules

And following directories to start off with.

These directories namly:

1. node\_modules
2. public: It contains index.html
3. src: It contains components

When we do **npm start**-> it will search ‘start’ in package.json .

start comes under react-scripts package.

"dependencies": {

"react": "^16.8.6",

"react-dom": "^16.8.6",

"react-scripts": "3.0.1"

},

"scripts": {

"start": "react-scripts start",

"build": "react-scripts build",

"test": "react-scripts test",

"eject": "react-scripts eject"

},

As we can see start comes under react-scripts.

**How to make class/Component**

Components look like HTML but it is actually **JSX.**

import React, { Component } from 'react';

class App extends Component {

render() {

return (

<div>

Hello World, Hi suryasnata

</div>

)

}

}

export default App;

**Container**:

It is a Component that can contain one or multiple Components.

When you start the app, internally it will go to index.js

**index.js**

import React from 'react';

import ReactDOM from 'react-dom';

import App from './container/App';

import \* as serviceWorker from './serviceWorker';

ReactDOM.render(<App />, document.getElementById('root'));

serviceWorker.unregister();

index.js will get ‘root’ from index.html and will then put <App /> Component which is our container in it.

**index.html**

<div id="root">

**App.js**

import React, { Component } from 'react';

import WelcomeMessage from '../components/WelcomeMessage';

class App extends Component {

render() {

return (

<div>

Hello World, Hi suryasnata

<WelcomeMessage />

</div>

)

}

}

export default App;

Here, App Component(In our case it is container) is using WelcomeMessage Component.

**WelcomeMessage.js**

import React, {Component} from 'react';

class WelcomeMessage extends Component {

render(){

return(

<div>

Hello

</div>

)

}

}

export default WelcomeMessage

**Starting the new Application from the beginning:-**

import React, { Component } from 'react';

class WelcomeMessage extends Component {

state = {

name: "surya",

age: 30

}

render() {

return (

<div>

Hello

<p>

My name is {this.state.name} and I am {this.state.age}

</p>

</div>

)

}

}

export default WelcomeMessage