**REACT JS**

* REACT IS A FREE AND **OPEN-SOURCE FRONT-END JAVASCRIPT LIBRARY** FOR BUILDING USER INTERFACE BASED ON COMPONENTS.
* REACT JS IS USED FOR **CREATE A SINGLE PAGE APPLICATION**.

**REACT JS COMPONENTS**

* IN REACT JS COMPONENT IS A FUCTION WHICH RETURN **REACT ELEMENT**(ONLY ONE).
* TWO TYPES OF REACT COMPONENTS:-
  1. CLASS BASED COMPONENT
  2. FUNCTION BASED COMPONENT
* TWO TYPES OF EXPORT COMPONENTS:-
  1. EXPORT DEFAULT COMPONENT
  2. NAMED EXPORT COMPONENT

1. **EXPORT DEFAULT:-**WHEN WE IMPORT THIS COMPONENT THEN WE HAVE WRITE WITHOUT CURLY BRACKETS.
   * EXAMPLE:- import React from ‘react’.
2. **NAMED EXPORT:-** WHEN WE IMPORT THIS COMPONENT THEN WE HAVE WRITE WITH CURLY BRACKETS.
   * EXAMPLE:- import { useState } from ‘react’.

* **COMPONENT EXAPMLE:-**

const App=()=>{

return(

<>

<h1>COMPONENTS</h1>

</>

)

}

**REACT JS STATE**

* STATE IS AN OBJECT IT’S LIKE VARIABLE, BUT IT HAS SOME ADVANCE FEATURES. LIKE WHEN STATE IS CHANGE THEN AUTOMATIC RE-RENDER THE COMPONENTS WHERE STATE IS USED.
* **SYNTEX:-** const [ stateName , chnangeStateMethod ] = useState();
* IF USE THE STATE THEN COMPULSORY IMPORT useState FROM React.
  + Import { useState } from ‘react’;
* **EXAMPLE:-** import { useState } form ‘react’;

const [ name, setName ]=useState(“ ”);

* WITH setName FUNCTION WE CAN CHANGE THE NAME VALUE. LIKE **setName(“Ahmad”);**
* WHEN WE CHANGE THE STATE THEN SYSTEM CREATE NEW OBJECT AND LINKED WITH STATE NAME

LIKE:-

**111**

**Replace to**

**112**

name

**&112**

**&111**

**New value of state**

**Value of state**

(Address of the memory)

**REACT JS PROPS**

* IN REACT JS PROPS IS AN OBJECT IT USE **TO PASS DATA** FROM PARENT COMPONENT TO CHILD COMPONENT.
* PASS THE PROPS DATA WITH **KEY AND VALUE PAIRS**.
  + SYNTEX:- <Component key=value key=value />
  + EXAMPLE:- <Contact name=”Ahmad” contact=”7383294925” />
* GET THE PROPS DATA WITH **DIFFERENT METHODS.**

1. **WITH (PROPS) KEYWORD:-**
   1. EXAPMLE:- const Contact (props) => {

return(

<>

<h1>{props.name}</h1>

<h1>{props.contact}</h1>

</>

);

}

1. **WITH PROPS DESTRUCTURE:-**
   1. EXAMPLE:- const Contact (props) =>{

const { name, contact }=props;

return(

<>

<h1>{name}</h1>

<h1>{contact}</h1>

</>

);

}

**NOTE:-** WE CAN ALSO GIVE NICKNAME OF PROPS LIKE,

EXAMPLE:- const Contact (props) =>{

const { name:myName, contact:mobile }=props;

return(

<>

<h1>{ myName }</h1>

<h1>{ mobile }</h1>

</>

);

}

1. **WITH PROPS DESTRUCTURE INTO FUNCTION ARGUMENT:-**

EXAMPLE:- const Contact ( { name, contact } ) =>{ };

**NOTE:-**WE CAN ALSO GIVE DEFAULT VALUE OF PROPS WHEN DO NOT GIVE PROPS DATA THEN PRINT THAT VALUE.

EXAPMLE:- const Contact ( { name=”Ahmad”, contact=”7383294925” } );

**NOTES:-**

* **JSX:- JSX IS A JAVASCRIPT EXTENSION. JSX PROVIDE US TO WRITE JAVASCRIPT LIKE HTML STRUCTURE. BUT JSX IS NOT A HTML. BECAUSE OF CLASSNAME, SELF CLOSING TAGS AND MANY OTHERS.**
* **BABEL:- BABEL IS A COMPILER OR LIBRARY OR CONVERTER WHICH CONVERT JSX TO JAVASCRIPT.**
* **ELEMENT:**- **REACT ELEMENT IS AN OBJECT .**
  + **EXAMPLE:- jsx:- <h1>Ahmad</h1>**

**Js:- React.createElement(“h1”,{},”Ahmad”);**

* **NPM: - NPM IS A PACKAGE MODULE FOR JAVASCRIPT. NO SPECIFIC FULL FORM OF NPM.**