------------------React -------------

day 6

1)Component is a template used for building application

2)component contains 3 components

1)markups (presentation)--- >html

2)styles---->css

3)logic(javascript) and uses jsx extension ( javascript xtension )---> logic is defined in jsx or tsx(typescript) and is similar to javascript only and added some additional features to it

3)react is a javascript library and mainly used for building user interfaces and react is a componet based .

advantages of components

1) components allows reusability and easy extensibility ( we canot achieve using by javascript )

2) you can build custom components or you can buld existing components providing by various third parties.

------Building a React componet----

1)react component is a function or calss of java script that returns markups , syles and configures functionality .

2)component function or class is defined in simple javascript file.

ex : component.js

function name(){ class name {

} }

note ---)every file in javascript is a module

3)component is a function or class which is defined in a module

4) in order to use function or class from module you have to mark it as a module

ex :

export function name(){ export class name {

} }

or

5)you can configure default export

export default function name(){ export default class name {

} }

6) you have to import modules and components

import {name} from module "Module" (if it is not default ) extension doesnot required

import Name from module "Module" (it it is default )extension doesnot required

Note : Every Module In javascript have one default Function or one default class ( means it allows only one dault member in a module )

7) every component must return Markup

8) function component returns Markup by using Return Key Word

ex export function name(){

return("<markup></markup>");

}

Note: return is a jump statement

9) class Component Returns Markups by using "render()"

export class name{

render(){

<markup></markup>

}

Note : in Mordern javascript we are avoiding to uses classes in react 17 and react 18 especially beacuse earlier we have problems with functions now it is cleared and now classses have problems better to use functions

Dynamic files ----> Src folder

static files ----> public folder

----react architecture ----

react running ---- localhost: 3000 by default

index.html---public folder

index .js----src folder

root.render(

<app/>

)

note : javascript strict mode helps to avoid code voilations and javascript developers need to follow strict mode

note: as per coding standers a variable must be decclared and use

ex : 1 gives error if u remove strict mode it will works but it is not recommended

<script>

"use strict";

x=10;

console .log(x);

</script>

ex : 1 gives error

<script>

"use strict";

var x=10;

console .log(x);

</script>

gives o/p as 10 ;

----setup bootstrap applicaton ----

1) open terminal

2)install following libraries

a) npm install bootstrap --save

b) npm install bootstrap-icons --save

3)go to index.js and import relative css files in bootstrap

import "../node\_modules/bootstrap/dist/css/bootstrap.css"

import "../node\_modules/bootstrap-icons/font/bootstrap-icons.css"

note :if one terminal is project is running and we can install any other dependicies by opening other terminal without affecting project

note : in javascript attributes wil not work only properties will work and similar to the react className and idName

--->jsx elements not allow void elements <----

in react every element must have an end tag

<p/>

or

<p></p>

=-------------jsx rules -----

1) you cannot use attributes only properties can be used

ex : <div class=""> ---------> invalid

<div className=""> ---------> valid

2) every element requires end tag or self ending

<input> -----> invalid

<input> </input> ------>valid

<input/>-----> valid

<img/> ---- > valid

3) component will not allows multiple lines of markup and you need a container and every thing must be rendered inside a container

<h2> hello </h2>

-----> invalid

<p> welcome<p>

<div>

<h2> hello </h2>

-----> valid

<p> welcome<p>

</div>

note -----( to beautify content in html is select html code and right click and format docuement or shift+alt+f)

-----------------------------------------------day 7-------------------------------------------------

Multiple components using in a web page

instead of div we can configure empty container and Fragment

1)<div></div> 2) <> </> 3) <React.Fragment> </React.Fragment>

Vmax ----> view wport max

----------------------- the first name of function will be capital letter in react when u are importing other wise it will give error-----------------

note--------> we can acess coponent inside another component

----> our application starts with index

---> react application can use class component but donot recommended to use

----> call back function is executed according to situation

Ex : function heello(){ var naveen= function (){

}

} it is mainly used in call back functions and it is a ananamous

function without a name and we can change a function and

There is no change in

function and we can print we can print multiple copies using call back function

only one copy

()=>{} we can change inorder to no change we can make it a const

const is means intialisation is mandatory

const a variable canot be declared and assigned it must be assigned

const loginComponent =()=>{

<h1> hello world </h1>

}

export default loginComponent;

to export a function above const variable of a variable assigned a function is we need to use export module

not Important

--------------------- react class component ----------------------------

class is a progromming template in javascript

class have a behaviour of

a)model

b)blue print

c)entity

class {-------------- entity----------------- business requirements

}------------------model ------------------------- data

class is set to be a entity when it is mapping to business requirements

class is set to be a model when it is mapping to data requirements

members of class

a class comprises of

1)constructor

2)properties

3)Acessors

4)Methods

constructor is used for intantioan

properties is used for storing the data

to give a fine control over a property u need acessories

method defines the actions to perform

q) can we decclare a variable inside a class ?

no

q) why we canot a a decclare a variable inside a class ?

because variable is immutable and classes only stores only mutable properties

how we can can store a data in class?

by using properties we can store data because properties are mutable

can we define a function inside a class ?

no we have only methods

note : we can define a variables in methods

variable and function is not a direct member of a class and it is a member of method

ex : class product {

properties =“Naveen ”

method (){

var name = “narayana”

function nav (){

}

}

}

\_ property name tells the developer tells another developer is to do further implementation and it is one type of naming convention

Accessors :

<script>

var  username = prompt(" enter username ");

var  designation = prompt("enter designation");

var    productName=prompt("enter product name")

class product {

    get ProductName1() {   //getter

        return this.productName;

    }

    set ProductName1(newproductname) {

        if(designation == "Admin"){

            this.productName = newproductname;

        }else{

            document.write(`hello ${username} you are not able to  acess to give product name `)

        }

    }

}

let obj = new product();

obj.ProductName1= productName;  // setter   and assignining product name to method productname1

if (obj.ProductName1) {  //passsing method name

    document.write("productname" + obj.productName)//getter

}

</script>

Accessors control over the property and we can put restrictions by using acessors

Obj.ProductName1 -------🡪 reference of object and ProductName1 is Method without parenthesis in javascript

Every derived constructor Must have super call

<script>

    class fruit{

       constructor(){

        document.write("hello naveen i am from parent class <br>");  //we can get next  line if we use in br tag  in double invertered commas

       }

    }

    class derrivedapple extends fruit{

       constructor(){

        super();

        document.write("hello naveen i am from derrived class" );

       }

    }

    let obj=new derrivedapple();

</script>

Passing massge in super class constructor

<script>

    class fruit{

       constructor(msg){

        document.write("hello naveen i am from parent class "+msg+"<br>");  //we can get next  line if we use in br tag  in double invertered commas

       }

    }

    class derrivedapple extends fruit{

       constructor(){

        super("naveen");

        document.write("hello naveen i am from derrived class" );

       }

    }

    let obj=new derrivedapple();

</script>

React Class component extends “React.Component” base

Technically we can say React Class component when it extends “React.Component”

Class home extends React.Component{

}

React.Component is a member of a core library you need to import if u have to use import React from “react”;

namespace is a type of related clases

namespace React {

class Component { }

class Service { }

}

Under namespace we have n number of classes

To acess the classes in namespace is as follows

Namespacename.class name ------------------🡪 React.Component

React.Service

React =============namespacename

Component ============= class name

---Note We have to use super class constructor in derived class when the constructor is explicitly is defined in derived class otherwise it wil give error

-------------------------no error ---------------------

<script>

    class fruit{

       constructor(msg){

        document.write("hello naveen i am from parent class "+msg+"<br>");  //we can get next  line if we use in br tag  in double invertered commas

       }

    }

    class derrivedapple extends fruit{

       naveen(){

        // super("naveen");

        document.write("hello naveen i am from derrived class" );

       }

    }

    let obj=new derrivedapple();

obj.naveen();

it is not necessary to call base or parent class constructor using super() when we do not have the constructor in the derived class and it will not give any error

note : a class component must render markup by using render() and return in that method otherwise it will give error .

import React from "react";

export class  ClassComponent extends React.Component{

    constructor(){

        super();

    }

     render(){

        return <h1> Hello naveen i am from class component </h1>

     }

}

what are the draw backs oops

it will not supports low level features

it canot directly interact with hardware services

it uses more memory

it is slow

it is complex in configuration

heavy on Application

------> what are difference between class component and function component <-------

http is a stateless protocol

stateless protocol works on go , get and Forget

it is good for secure operations and it is not good for continue operations means it will clean up a memory when it is loaded and worked and it will clear when next operation failed

every techonology in the world must use statemanagement techniques

types state management techniques

application

session

cookies

query string

hidden

component life cycle helps to track the issues from starting to end

function component is stateless and from react 18 onwardes it is state

you can explicitly use state in function component state

no life cycle methods is c onfigured and you explicitly configure

for which component you will design a class ?

the component that need regular extensions or changes we go for classs

----------------------------Data Binding ----------------------------------------

Data binding is a technique that defines how data is binded to ui (means how we can present ui or view )

Javascript and jquery uses lot of dom methods for data binding

React can handle data binding using binding expression and simple collection methods

React Data binding expression is represented by ------------------🡪 { }

You bind to a literal or You bind to a Property

<div>literal </div >-------------🡪 anything which is in btw we place div tag or any tag or without tag is literal

You can bind only html properties not javascript properties or otherwise you have to bind to a literals

You canot bind any dynamic value to attribute .

In html whatever you write is attributes and html don’t have the properties

Properties are used by the languages and the libraries to handle this elements

In react we can bind only properties and we do not bind to the attributes

<Input type =”text” size=”4” required>

Class ----------🡪 attribute

Size is both attribute and property

Classname is a property

You canot bind any dynamic value to the attribute

For table we donot have the height property in javascript , html and even in react we get error when u assign a height property in react js

In react js we bind table height properties by using style binding techniques

<div class={}> // invalid

<div className={}> // valid

<table height={}> // invalid because table does not have the height as property \

React does not require dom methods and all these

How to read a collections without explicitly using for loops and iterators etc ………

Var collections = [a, 1,3,8,10]

In react iterations every iterating item must have unique key

---------------------displaying table data in react --------------------------

export  function TableComponent() {

    var products = [

        { Name: 'Samsung TV', Price: 56000.44 },

        { Name: 'Nike Casuals', Price: 4200.44 }

    ];

    return (

        <div className="container12">

            <h2>Products Table</h2>

            <table className="table table-hover">

                <thead>

                    <tr>

                        <th>Name1</th>

                        <th>Price1</th>

                    </tr>

                </thead>

                <tbody>

                    {

                        products.map(product=>

                            <tr>

                                <td>{product.Name}</td>

                                <td>{product.Price}</td>

                            </tr>

                        )

                    }

                </tbody>

            </table>

        </div>

    )

}

We have good cooding and bad coding and as a developer we need to follow the good coding rules

And to follow we need to folollow the coding rules and you neeed to follow certain rules in good coding websites ------------PMD---------------

You need to go to this website

https://docs.pmd-code.org/latest/

nested iteratios programme

export function DataBindingNestedComponent() {

    var menu = [

        { Category: "Electronics", Products: ["Samsung TV", "Mobile"] },

        {

            Category: "Footwear", Products: ["Nike Causals", "Lee Boot"]

        }

    ]

    return (

        <div className="container">

            <h2>Categories</h2>p Screen Recorder

            <ol>

                {

                    menu.map(item =>

                        <li>{item.Category}

                            <ul>

                                {

                                    item.Products.map(product =>

                                        <li>{product}</li>

                                    )

                                }

                             </ul>

                        </li>

                    )

                }

         </ol>

</div>

    )

     }

Neted options group in data binding

 </ol>

            <h2>OptionalCategories</h2>

            <select>

                {

                    menu.map(item =>

                        < optgroup label={item.Category}>

                            {

                                item.Products.map(product =>

                                    <option>{product}</option>

                                )

                            }

                        </optgroup >

                    )

                }

            </select>

React supports one way binding and it is the draw back of react

Actually data binding are of two types

1. One way binding --🡪 it will read and present the data and any modifications it will not update back
2. Two way binding ---🡪 directly react does not supports the two way binding
3. In React two way binding is not implicitly supported and you have to implement explicitly by using event binding
4. Two way binding is the process of updating the changes in ui and updates back to data
5. You have to explicitly manage two way binding by using event binding like
6. On change b)on blur c)on Key up etc

Never uses a variable to handle data component

Variable are immutable

Componets must have dat in only mutable properties

You have to use state for storing data

Class component in react have default state where as function component we do not have

And we need to use explicit method

You can configure state for function component by using use state hook

With the use of variables for storing and accessing the data is not good

You have to use state otherwise http will be a stateless protocol

If you do not use state http will be a state less protocoal by default http was a state less protocoal to configure the state we use use state and use state is a method for storing any type of data either array or string or Boolean etc ……….. and by default use state is array type , use state contains multiple values and how to acess those multiple values and use it

To individually handled array is called de structuring of an array . and destructuring is allowed in a function and use state contains a getter and setter

You have to use destructuring technique to acesss the acessors from use state

There are many types of destructuring techniques was there

1. Object destructuring
2. Array destructuring

Note : in class component the actions to be performed on component initialization are performed by using life cycle hook method

a)componentDidMount()

b)componentMount()

in function component it is managed by Hook Method “useEffects()”

function donot have constructors and function have use effects method and it acts a constructor

syntax ----useEffects(()=>{

},[dependicies])

The parameters are used to modify the function and dependicies in useEffects is acts as a parameter and if you want to keep empty parameter [] dependicies is to keep as empty

In javascript promise of Fetch method we have problem that fetch method returns the data in the form of a binary data and we need to convert binary data in to the json object

Two way binding

React does not support the two way binding

You have to explicitly implement by using event binding

It is the process of identifying the changes in ui and updating the changes in data source

What is Event?

Event is a message sent by sender to its subscriber in order to notify the change. Recorded with Top Screen Recorder

Event follows a software design pattern called "Observer".

It is a communication pattern.

function Insert(){ (Subscriber)

}

Sender

<button onclick="Insert()"> Insert </button>

JavaScript events are managed by "browser" window object.

JavaScript Browser Objects

window

location

navigator

history

document

React can't directly use the browser events [Actual DOM]

React uses "SyntheticEvents" object for Virtual DOM.

Synthetic Events will map to browser events.

Onclick=”insert()”--------🡪 event handler

Onclick is -🡪an event

Every event handler in javascript contains two arguments

1)arguments ----------🡪 client, keyCode

2)this ---🡪 it will send object related properties like id, class , name

React Synthetic Events will support only one "event" argument.

[ 'this' is not allowed ]

event.EventProperties --------------------> event related properties accessing in react js

event.target.objectProperty-----------------🡪object related properties accessing in react js

Recorded with Top Screen Recorder

Syntax:

function Insert()

{

}

Button

Example

Syntax:

function Insert(e)

{

e.clientX, e.keyCode;

e.target.id, e.target.name, e.target.value;

}

<button onClick={Insert}>

React uses all JavaScript Events

1. Mouse Events

onMouseOver,

onMouseOut

onMouseMove

onMouseDown

onMouseUp

2. Keyboard Events

onKeyUp

onKeyDown

onKeyPress

3. Element State Events

onChange

onSelect

onFocus

onBlur

onChecked

4. Button Events

onClick

onDbIClick

onContextMenu

5. Timer Events

setInterval()

clearInterval()

setTimeout()

clearTimeout()

6.Clipboard Events

onCut()

onCopy()

onPaste()

7. Touch Events on TouchStart()

onTouchEnd()

on TouchMove()

8. Form Events

onSubmit()

on Reset()

Note: If you are binding any value to HTML form element using "value" property, then it is configured as "read-only".

It allow read/write you have to bind "onChange" event for HTML form element.

If value binding is not defined, then you can configure without "onChange".

Syntax:

const [ userName] = useState('John');

<input type="text" value={userName}> // invalid

--------------------w3 schhols----------------------------

-------------------react ----------------------

React is a JavaScript library for building user interfaces.

React is used to build single-page applications.

React allows us to create reusable UI components.

------To use react app in existing application or by using urls of cdn of react 18 ------

<script src="[https://unpkg.com/react@18/umd/react.development.js](https://unpkg.com/react@17/umd/react.development.js)" crossorigin></script>

<script src="[https://unpkg.com/react-dom@18/umd/react-dom.development.js](https://unpkg.com/react-dom@17/umd/react-dom.development.js)" crossorigin></script>

<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

Complete locally React app creation

Open command prompt of any location and run this command

npx create-react-app my-react-app -----🡪 this command will create complete react application

diff b/w normal and arrow functions

normal function

hello = function() {

return "Hello World!";

}

Arrow function

hello = () => {

return "Hello World!";

}

The arrow function expects a return value, and returns the value by default, without the **return** keyword.

hello = () => "Hello World!";

React's goal is in many ways to render HTML in a web page.

React renders HTML to the web page by using a function called createRoot() and its method render().

What is JSX?

JSX stands for JavaScript XML.

JSX allows us to write HTML in React.

JSX makes it easier to write and add HTML in React.

JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement()  and/or appendChild() methods.

JSX converts HTML tags into react elements.

You are not required to use JSX, but JSX makes it easier to write React applications.

With JSX you can write expressions inside curly braces { }.

The expression can be a React variable, or property, or any other valid JavaScript expression. JSX will execute the expression and return the result: