# React Js

- front end technology
- used to create UI
- UI - User interface
  - it refers to the visual and interactive part of a software or a web application that allows
    user to interact with
  - UI encomposses all the elements like buttons, drop downs, carousels, cards, forms and any
    other visual element that a user interact with to use the web application/software
- React is a javascript libray
- It make use of multiple javascript libraries to create front-end
- Libraries: predefined code to do a particular task
  - set of codes are bundled into  a library
- Framework: is a pre-built, reusable tools, libraries and conventions that provide a structured way to develop a
  software application

## Features of React

- used to create front-end/UI
- SPA: single page application

  SPA is a web application, that provide a seamless, dynamic user experience without need to reload the entire
  page from the server

  In react we have only one html file, called index.html inside public folder. Browser only loads this
  html page, while user request for other pages, only the content inside this index,html page reloads

  - In traditional web-application, we have multiple html pages, for each iteartion, the application typically
    loads a new page from the server resulting in full page refresh/reload

- React uses Virtual Dom

  Dom : Document object model

  Real Dom: The actual representation of the webpage that browser renders

  Virtual Dom: Light weight representation of the real dom

How react uses virtual dom?

While a react application loads/starts, initially it create a Real Dom along with Virtual Dom

if any events happened and changes the html element, it create a new virtual dom, then compares the
newly created virtual dom with previous one, and find the difference.

Then reacts updates only the changed html elelment based on the difference identified

Diffing: The process of comparing new virtual Dom with the previous virtual dom to find the difference

Reconcillation: After diffing, react update the Real dom with only the

```
changes
    - React js is fully component based
    - Data Sharing - is uni-directional , ie from parent to child component
    - Language: jsx
                combination of javascript, html and css
    - file extensions for component, generally we are using .jsx
    - transpiling:  process of converting jsx into JS, HTML and css
            Babel is a compiler, that perform the transpiling

    - How to create a React application
    ------------------------------------
    step1: we need a npm packge create-react-app
            npm install create-react-app
    step2: by using the above npm packge create a react application
            npx create-react-app application_name
    step3: after the application is created, navigate to apllication folder and run
it
            npm start
    By default the React application runs in
    http://localhost:3000/

    npm : Node package manager
    npx : Node package execute

    File and Folders inside a React application
    --------------------------------------------
    README.md: Details of the project
                How to start application
                default running domain and port
                Scripts used
    Package.json: Heart of the React application
                  it holds the npm configuration
                  it includes all the installed libraries/node modules
                  if we install a new module, it will automatically updated in
package.json
                  also it includes scripts, to run the application, build the
application, test the application, execute
    package-lock.json: More detailed version of package.json file with with all
details of the libraries/modules
                    with exact version number
    .gitignore: it lists files and folders that need to be ingored while pushing
the code into github
                like node modules, exact
    Node Modules: All the libraries used in the application
                While pushing the code into github, we are not pusing the node
modules,
                when user clone that project, node modules are not there, For
getting the node modules
                below commpand user need to run
                npm install
                So what will happen is that, it automatically installs all the
packages/libraries mentioned
                in the package.json file
    Public folder: Files that can be accessed throughout the project
       - index.html: this is the only file, that browser loads while running the
react application
       in index.html file, there is one div with id =root, in that div, we are
binding the page/data
    src:
        index.js: entry point to the application
        here we imports multiple depencecies/libraries
        inside this index.js, we access the div that mentioned in the index.html
file buy using
        const root = document.getElementById('root')
        then  to root, we bind the App component
    App.js: parent component, this the parent component taht loads first in the
index.js file

    components:
        Basic unit of the react application. We can split the entire react
application into small, small components
        like header, footer, homepage
        - while creating component, it should start with Capital letter
        - common extension we are using is .jsx
    in react class we used is className
    JSX expressions must have one parent element
    - we can use empty fragment also as parent tag <>....</>

    return(): what the user need to view on the screen

    Basic structure of Functional component

    We can use VS code extension to automatically create the structure of a React
application
    ES7+ React/Redux/React-Native snippets
    command:  rfce or rfc

    function ComponentName(){
        //  javascript code
        return({
            // jsx code
            <>
               // User view items
            </>
        })
    }

    Parent component: The componet where child components are binded
    eg: App component
    Child component: the components that are binded in parent component
    {}:  is used to bind dynamic or values in js code section of the react
componnet to JSX part

    Difference between App.css and index.css
    ----------------------------------------------
    index.css:  This is the global style sheet, whatever the styles mentioned in
index.css file
         will affect the throughout apllication. like browser style setting, body
styles
    App.css:  This style sheet is only belongs to App component and its immediate
children
        The styles mentioned in App.css will only affect the HTML elements belongs
to App component
        and its immediate children

 Data Transfer between components:
 - In react by default data passing is uni directional, ie from Parent component to
child

 props:
 ----------
 - it is a concept in react
  is used to send data from parent component to child component
  here how we are passing data is that
  keyName=values
  while we are accessing this value in child component, we got this as object
  Then we need to use dot operartor to get the specific value
  or we can destructure with respect to key name also

  class component
  -----------------------
  - here we are using class to create component
  - By default class doens't have properties of react, do we need to import React
library from 'react' and then
   - the created class inherits properties and methods from React.Component class
   - render() method is used to render the UI, return is given inside render()
   - Class component name must be start with Capital letter

Styling
-------------
1) inline css
2) External
3) modules

Inline: syntax
=============
style={{property:'value', property2:'value'}}
inside curly brace, pass it as an object
```

```
External:
------------
create a .css file
import that css file into the corresponding component
- inside css file, there is no need to export

Modules
-----------
 file extension : file_name.module.css
 - created inside src folder
 - we can store the imported module into a variable and style using that variable
 - can be accessed throughout the project

 events
 -----------
 Action performed by the user, which leads to some update in the Dom
 eg: button click
    key downs
    mouse hover

    ButtonClick
    --------------
    event: onClick={}
    two types:
    1) not passing any argument from jsx code to js code
            onClick={function_name}

    2) passing argument from jsx code to js code
        here we call function as callback function
        onClick={()=>function_name(arguments)}

Getting value from input box
-----------------------------
onChange() method is used to get value from input box
syntax
 <input type="text" placeholder='Enter the Topic Name'
    onChange={(e)=>getTopic(e)}
 />
  const getTopic=(e)=>{
        console.log(e.target.value)
    }

Conditional rendering
------------------
loading the content in browser based on some Conditional
1) if
    operator used is called truthy operator &&
2) if else
 opeartor used is ternary opeartor ?:
```

```
 State
 ------------
 - State is an Object which is used to store property of a react class component
 - whenever the value of state changes, the entire component re-renders

 setState():  is used to update the value of state

 Life cycle of a React Class component
 ------------------------------------
 life cylce of a component means,
 different life cyle stages from the creation of component to its destruction
 destruction:  removing from the view/dom
  1) Mounting phase
   -  adding or binding the component into Dom
    1) constructor() - first executed method
    2) getDerivedStateFromProps(): this method will check any data is coming in
props
    3) render(): loading jsx code into Dom
    4) componentDidMount()
  2) updating phase
    - updating the Dom
    1) getDerivedStateFromProps(): is there any update in props value
    2) shoulComponentUpdate(): return true or false
    3) getSnapShotBeforeUpdate()
    4) render()
    5) componentDidUpdate()
  3) unmounting phase
    1) componetWillUnmount()

React Hooks
----------------
    - pre-defined functions
    - it is used to provide specific features to functional component
    eg: useState()
       useEffect()

Usage
-----------
1) import hooks from React Library to the functional component
2) call react hooks at the top level of the js code of functional component
3) Hooks cannot be used base on certain conditions

Diffrent Types of Hooks
----------------------
1) predefined: in built hooks, we can use it directly by importing it into the
component
    useState()
    useEffect()
2) Custom: user created based on particular need
```

```
statefull component and stateless component
-------------------------------------------
- class component is called stateful component, because state is inbuilt on class
component,
  we can directly use it by calling this.state
  and setState() method can be used to change the value of state
- functional component is called stateless component, by default fnal component
doesn't have state
- for implementing state in fnal component, we have to import state from React
library
- useState() is the hook used to implement state in functional component

useState() hook
--------------
 this hook is used to implement state in functional component
 synatx:
 const [variablename, method] = useState("Initial value")
 variablename: it hold the value of state
 method: it is used to update the value of state
 initialValue : not mandatory

 if the value of a state is JS object, and when we try to change the value of any
one
 of the element in that object, the entire object is replaced by only the updating
value'
 so, to resolve this issue, we have to use spread operator(...)
 EG:
   const [colors, setColor] = useState({
        first: "Red",
        second: "Blue",
        third: "Yellow",
        fourth: "Green"
    })

    <button type='button'
onClick={()=>setColor({...colors,second:'Orange'})}>Change Color</button>

React forms
----------------------------



Using bootsrap in React
----------------------------
1) CDN:  we can copy the cdn url from bootsrap website and paste it in index.html
file (head tag)
2) my installing bootsrap module in our application

Regular expressions
-----------------------
-it is used to check whether a given input/string have a particulr patter
```

```
Rules to create a Regular Expression (RE)
--------------------------------------------
- it should have forward slash at the beginning and end
- starting of the expression is indicated by  ^ (raised symbol)
- ending of the expression is indicated by $
eg: /^[0-9]*$/
the above RE checks whether the input contains only Numbers
Usage:
var x='123h';
console.log(!!x.match(/^[0-9]*$/))
!! is used to conver the result of RE into a boolean value

var x='Hg';
console.log(!!x.match(/^[A-Za-z]*$/))

Pure function
-------------------
1) Pure function always produces the same output
 - that means, no matter how many times we call the function with same arguments,
 it always return the same result

 function add(a,b){
    return a+b;
 }
 add(3,4)

 - content inside the pure function does not change based on any
 external data that is coming from API or state changes
 - it has no side effects
 Eg of side effects: Api call. read or write file, IO opeartions
 Pure function cannot be affected by any side Effects

useEffect()
==============
 this is one of the most commonly used Hook in React js
 - it is used to handle side effects
 - some examples of side effects are fetching data, directly updating the dom and
timers
 - useEffect hooks accept 2 arguments
 - useEffect(<function>, <dependency>)
 - second argument optional
 - useEffect hook is used to fetch data after the component is mounted/rendered

 In 3 situation it can be used
 1) No dependency is passed: useEffect runs on every render
 syntax
 --------------
 useEffect(() => {
  //Runs on every render
 });
```

```
2) An empty array is passed as dependency: useEffect run in first render
syntax
-----------
useEffect(() => {
  //Runs only on the first render
}, []);

3) props or state is passed as dependency:- use effect runs in the first render and
also
        when the value of props or state changes
syntax
-----------
useEffect(() => {
  //Runs on the first render
  //And any time any dependency value changes
}, [prop, state]);

vite
=============
 - Vite is a modern front-end build tool that offers a fast development
 experience for web projects
 - it can be used with react
 - it make building of the project very quick
 - it replaces trditional build tool 'webpack'

 How to create React application using vite
 ----------------------------------------------
 npm create vite@latest application_name -- --template react
 - navigate to application_name folder and execute npm install
 - npm run dev : used to run the react application

 Routing
 ---------------
 Routing means navigate from one page to another
 or loading some other components
 Package used: react-router-dom
 npm i react-router-dom
 steps :
 1) Enclose <App/> inside <BrowserRouter></BrowserRouter> (in main.jsx)
 2)  Go to App.js and place components that need routing inside <Routes</Routes>
 3) inside Routes, add each path along with component need to be loaded inside
<Route/>
 syntax
 --------------
 main.jsx
 ------------
   <BrowserRouter>
      <App />
   </BrowserRouter>
 App.js
 ------
    <Routes>
        <Route path='/' element={<Landingpage/>}/>
        <Route path ='/home' element={<Home/>} />
        <Route path = '/watch' element={<Watchhistory/>}/>
    </Routes>

Navigate to particular page/ load corresponding component
-------------------------------------------------------
Link tag is used for that

syntax
------
    <Link to='/home'>
            Home page
    </Link>

    Data communication in React
    -----------------------------
    1) Redux
    2) Context API
        Above 2 are used to comminicate between un related components
    3) State lifting - for comminication between siblings component
            sibling components: componet with same parent

    State lifting
    ---------------
     -  state lifting is used to pass data between sibling components'
     - Here what we are doing is that, create a common state in parent component
      and pass it to child components

    Props drilling
    ------------------------
    Props Drilling is the process of passing data from a parent component
    to deeply nested child component through multiple layers of intermediate
    components


Disadvantages of Props Drilling
--------------------------------
- Props drilling tightly couples components together (component become more
dependent
    on another component)
- There is a chance of losing data in between any components, if any corresponding
    components fails
- intermediate components doesn't need data. they are only passing it down
- It makes code more complex
```

Solution to this props drilling
--------------------------------
1) Redux
2) Context API

Redux
-----------------
Redux is a state management library that allows us to store
the entire application's state in a central Store making it
accessible to any component in the tree
- It is javascript library
- It is not specific to any specific FE technology, can be used with
Angular, react, Vue,..

Main Components of Redux
-------------------------
1) Store: we are creating a state inside store, and will make that
   globally accessible
   - component has no ability to update the
   value in state
   - But component can read the state inside Store
2) Action
 - It holds the logic to update the state
 - Dispatch - is a method used to call update function
   inside action
- Action send the response (payload) to reducer
 - payload: the response after logical operation
3) Reducer: It update the state inside store

The above is the Exact Redux concept

Redux toolkit
---------------
- Advanced version of Redux
- Here action and Reducer is written in single file
- combination of Action and Reducer is called Slice