**what is react js ?**

react js is a library of javascript

react js is also called a framework of javascript.

Framework provides its own structured.

Framework follow some life cycle to completed any web applications.

React js is also follow mvc architectures.

MVC stands for model view and controller structures

React is a library of javascript i.e call through its virtuals DOM.

React is also follow a life cycle to completed its web apps.

Life cycle of react js :

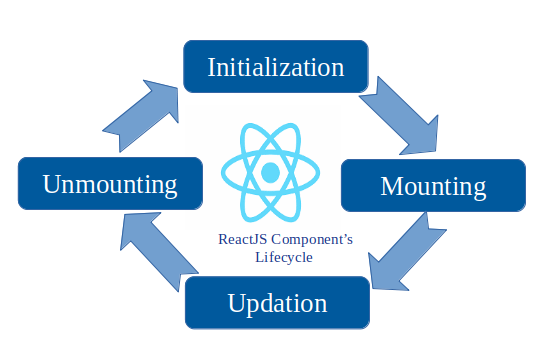
1) initializations

2) mounting

3) updatation

4) unemounting

Structured of react js life cycle



a) initializations : react will be initialized data in its virtuals DOM to called or load view in react js.

b)mounting (add) : mount the data via virtual DOM or called html data in react js via its ID or mount data inside of react js using virtual dom (document.getElementById(“id”));

c) updations: we can used to update one components data with another components and its will be update data or reused data one components to another components.

d) demounting : demounting means removing data from virtuals DOM that concepts is called demounting.

Note : react is a library of JavaScript to create a frontend for any website or web apps.

React js is also called a framework and framework is follow a architectures i.e called MVC architectures.

MVC is stands model view and controller but in react js data will be load or called via virtual DOM in MVC.

**MVC life cycle in react js :**

1) initializations

2) mounting

3) updatation

4) unemounting

**Difference b/w DOM & Virtual DOM**

**DOM (document object model)**

1) load data via of html in js via DOM

2) DOM is low effieciancy

3) DOM is slow and not load fast data

4) DOM local work

**Virtual DOM**

1) load data via of html in js via DOM

2) Virtual DOM is fast load effieciancy

3) Virtual DOM is fast renders data 10x time of faster than DOM

4) Virtual DOM are dynamic data or fast render dynamic data of server(local data server).

**Advantage of react js :**

a) react js is a library of js and developed and manage by facebook engineer.

b) react js is used to create a web apps and website.

c) react js used in frontend but behave just like backed.

d) react is used to make a single page applications

ex: any to do app

e) react is used to load data without page reload or refresh

f) react js help to fast load data on broswers.

g) React js enable to call api in application

h) react js is used seo friendly or provides seo friendly website.

i) react js is used reusable components.

j) react provides a lot of library to help create to call a dynamic data.

k) react js provides some inbuild modules i.e used to help pass data or reused data one page to another page.

Ex: a) import About from ‘about’; // import

b) export default About; // export

c) export {About,Services,Contact} // export

l) react support handy tools or some packages

ex: React routing , React bootstrap , react Saga, React axios , React Swal , React Redux tools.

React external tools or package install by

cmd : npm install react-router-dom

**how to create a react app :**

react used a node\_modules tools to enable to install a default react app or its library so we first download node js to enable node\_modules in our react app.

Node\_modules install via npm and npm is a package manager of node js i.e help to install all library of react js app.

Node js or react js is a open source language.

Free download and used without paying any charge or purchasing any licence.

**step 1: download node js**

[**https://nodejs.org/en/download**](https://nodejs.org/en/download)

**step 2: how to check node js version or npm or npx version**

**NPM :** node package manage its also instal or create react app

**NPX :** node package extentions its also install or create react app but its also create a default extentions of react js and enable strict mode inside of react app.

Cmd : node -v

Cmd : npm -v

Cmd : npx -v

**Node : npm install**

E:\Reactjs930mwf\module3>node -v

v18.17.1

E:\Reactjs930mwf\module3>npm -v

9.6.7

E:\Reactjs930mwf\module3>npx -v

9.6.7

**Step 4: install a react app**

npx create-react-app hello-world

E:\Reactjs930mwf\module3>npx create-react-app Hello-world-app

Cannot create a project named "Hello-world-app" because of npm naming restrictions:

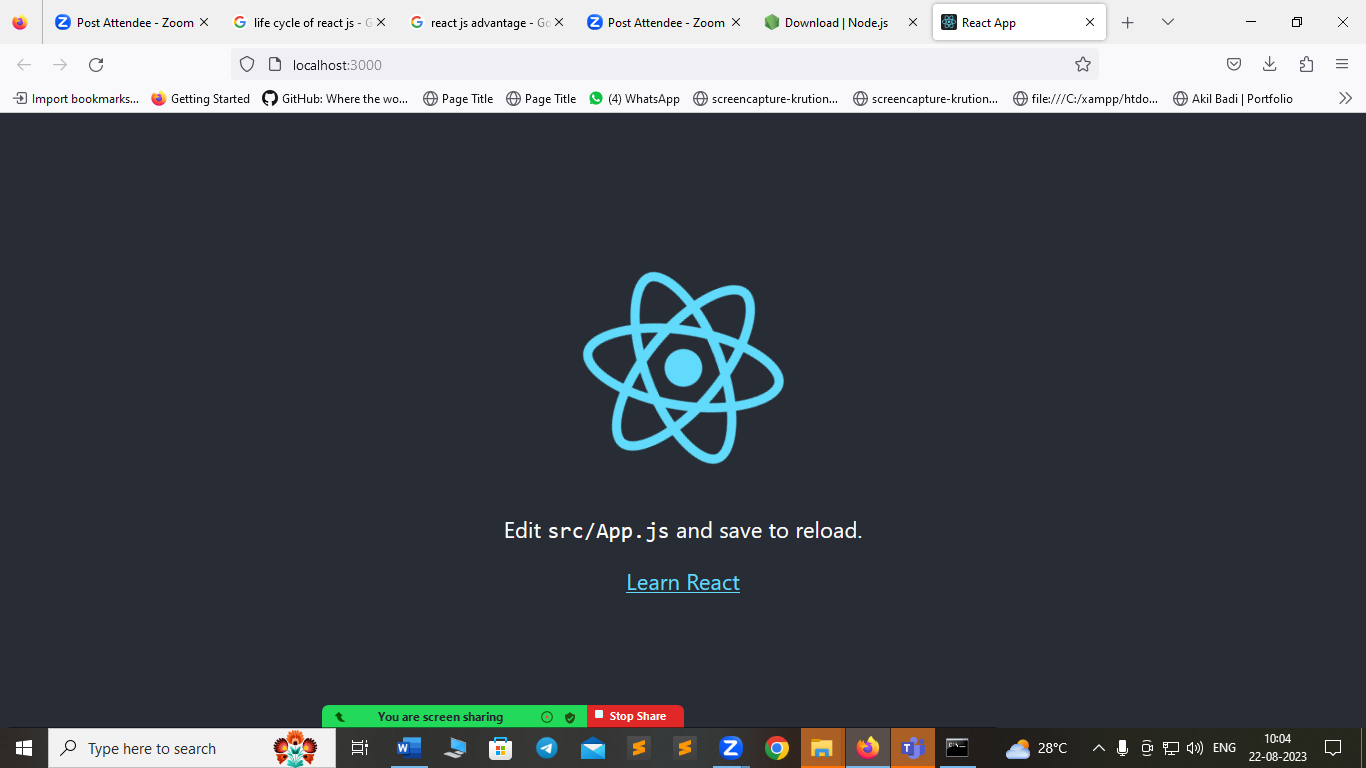
\* name can no longer contain capital letters

Please choose a different project name.

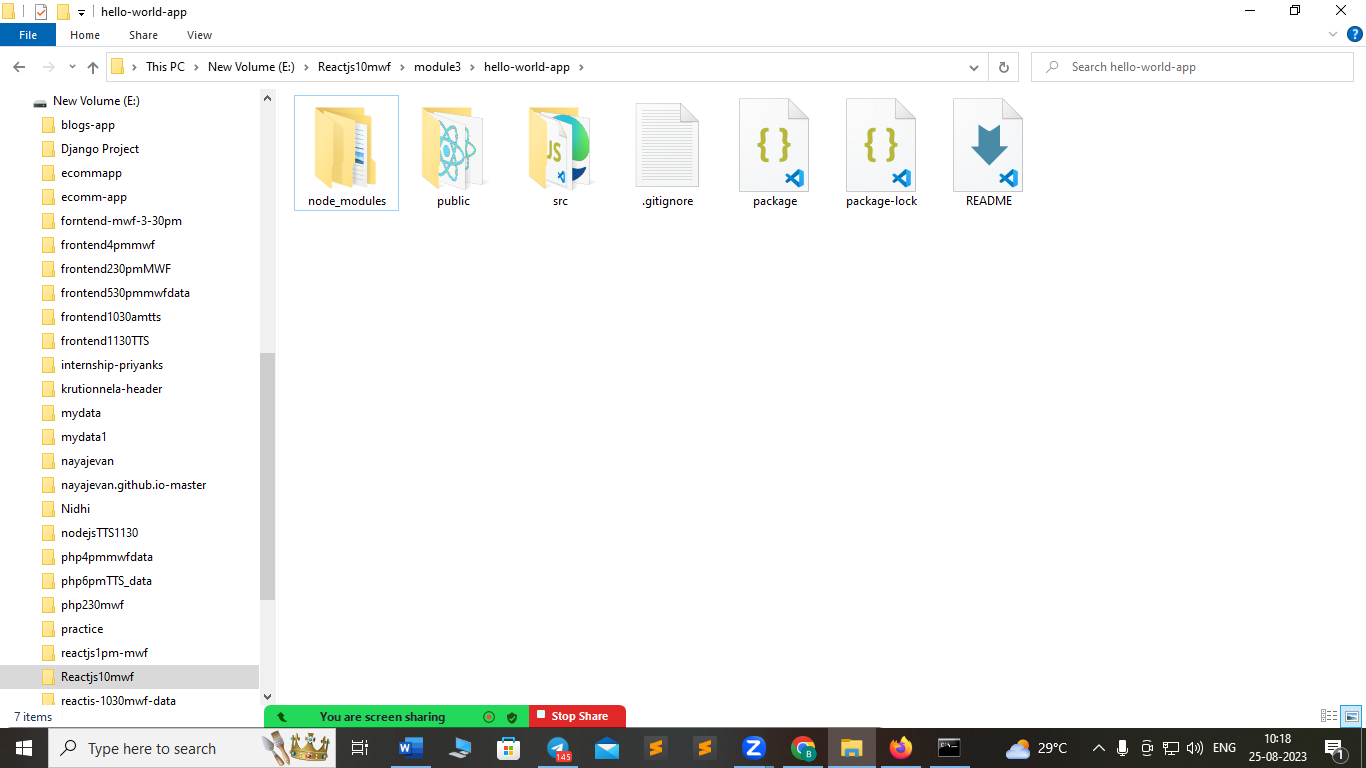
**Step 5: how to run react app**

**npm start**

[**http://localhost:3000/**](http://localhost:3000/)



**React js directory systems :**



**Node\_modules :** node\_modules is default modules comes through node js install or npm install through node module package.

All react modules, library and package are installed here

Modules : module is import and used in react js peace of files

Library : Library is inbuild bundle of file that is used to provides a complete functionality

Ex: npm install Swal;

Package : package is also a bundle of file we will install and used packages and library in react js.

Ex: npm install Email;

Npm install react-router-dom;

**Src :** src is default our app files of react js where wecan create a components | modules and import all libraryies is src file.

src is main directory of our application where we create all components of our apps.

Examples : if we want to create SPA (single page application hello world in react js )

a) components create => src

b) style is also create => src

c) images of apps is also created => src

d) logo create in src => src

**Public :** public is accessible anywhere and we can load our index.html inside of public

Webpages load from public and called via virtual DOM in src =>components

document.getElementById(“id”);

public is also stored all assets file of your react js apps

a) assets => css => style.css

js => load.js

fonts => timesnew roman

images => logo.png

**Components =>** components file always create is src => Apps.js

Aboutus.js

Note : components is reusables one file to another file and it is peace of small files saved in with .js exetention.

**package.json =>** all information’s about apps of react js **stored in .json formate in package**

**React js version :** "react": "^18.2.0",

**package-lock.json** => cache created package-lock.json formate it is same as package.json but we can not do anything in this file.

**Readme :** all information about steps of apps how to run , how to build up apps via command it is mention in readme.

Run react js app

cmd : npm start

how to checked react js version

a) checked via command

PS E:\Reactjs10mwf\module3\hello-world-app> npm view react version

b) checked via script

import library in react inbuild components or create your own components also

import React from 'react';

let a=React.version;

function App() {

  return (

    <div className="App">

      <header className="App-header">

       <h1>Version of react js is : {a}</h1>

      </header>

    </div>

  );

}

export default App;

c) checked via package.json

**React js version :** "react": "^18.2.0",

How to print Hello-world :

Index.html

<!DOCTYPE html>

<html>

<head>

    <meta charset='utf-8'>

    <meta http-equiv='X-UA-Compatible' content='IE=edge'>

    <title>Our first app in react js</title>

    <meta name='viewport' content='width=device-width, initial-scale=1'>

    <link rel='stylesheet' type='text/css' media='screen' href='main.css'>

    <script src='main.js'></script>

</head>

<body>

    <div id="demo"></div>

</body>

</html>

Index.js

import React from 'react';

import  ReactDOM  from 'react-dom/client';

import reportWebVitals from './reportWebVitals';

import Hello from './components/HelloWorld';

const root=ReactDOM.createRoot(document.getElementById("demo"));

root.render(

<React.StrictMode>

<Hello />

</React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

components/HelloWorld.js

import React from 'react';

let name="Hello world";

function Hello()

{

    return <h1 align='center'>{name}</h1>

}

export default Hello;

Hello world print from scratch :