**HTML, CSS, JavaScript ES6**

* ReactJS with JavaScript
* ReactJS with TypeScript
* React18 and update
* MERN Project
* M – MongoDB
* E – Express
* R- React
* N- Node JS

Redux  
All 3rd party libraries required for real-world  
axios  
formic  
yup  
mui  
react native mobile  
ionic  
Deploy on cloud and local  
webpack

**What is React or React JS?**  
- It is a JavaScript library for building UI.  
  
**Why React?**  
  
**FAQ: What are the challenges in modern web development?**  
Ans:  
- Web Started in early 1990's - Tim Berners Lee  
- Current Generation: 90% users are using web with smart devices.  
- Traditional web applications have issues with modern devices  
  
    1. Unified UX  
        - Application must have same experience across all devices.  
        - Mobile users must get access to everything.  
  
     2. Fluid UX  
        - User stays on one page and can get access to everything on to the  
          page.  
  
    3. Loosely coupled and extensible  
        - You can build features remotely  
        - Push the new features into existing app without having  
          catastrophic failures.  
  
     4. Simplified deployment

**FAQ: What is solution?**  
Ans :  Better build  
            SPA [Single Page Applications] or  
            PWA [Progressive Web Applications]  
  
**FAQ: How to build SPA & PWA?**  
Ans :  
        HTML , CSS, JavaScript, jQuery => SPA => Yes  
        Issues with JS & JQ  
        - Uses lot of DOM manipulations  
        - Lot of coding  
        - Heavy  
        - More memory  
        - Legacy  
         
        Modern => Angular, React, Vue, KnockoutJS, Backbone, Ember etc..  
     
                 
**FAQ: What is difference between Angular and React?**  
**Ans:** Angular is a Framework  
        React is a Library  
  
  
        jquery UI  
        React  
        RxJS  
        Redux

1. What are the challenges in modern web development?

a) Unified UX

b) Fluid UX

c) Loosely Coupled and Extensible

d) Simplified Deployment

2. What is the solution?

SPA, PWA

3. What are the issues with JavaScript and jQuery?

4. What is solution?

React, Angular, Vue, BackBone, Knockout etc..

5. What is difference between Angular and React?

**Features of React**

1. Virtual DOM

What is DOM?

- DOM is an hierarchy of presenting elements in browser.

- Document Object Model.

Markup => Bytes => Chars => Token => Node => DOM => Layout => Render => Paint

What is Shadow DOM?

- It is an hierarchy configured for every component in HTML.

- Shadow DOM comprises of

a) Markup <html>

b) Styles CSS

c) Logic JavaScript

What is Virtual DOM?

- It is a copy of actual DOM in memory.

- Update are first appended into Virtual DOM and then into actual DOM.

- Faster in redering output.

2. Component Based

Reusability , Maintainability, Testability, Loosely Coupled

3. Modular

Legacy

Modular

4. AOT Technique

**What is React?**

What are the features of React?

a) Virtual DOM

[DOM, Shadow DOM, Virtual DOM]

b) Component Based

c) Modular

d) AOT

**What are the issues with React?**

- React is not designed for your requirements hence lot of GAP's.

- To fill the GAP's you have to depend on lot of 3rd parties.

- SEO issues

- Pace of development.

- Poor documentation.

**Setup Environment for Web Development**

1. Download and Install "Node JS" on your PC.

- We are downloading Node JS for a package manager called NPM.

- There are various package managers

NPM

Bower

NuGet

Yarn

RubyGems

- Install Node JS: https://nodejs.org/en/

- Test Node and NPM Version

Node > 14

NPM > 6

C:\>node -v

C:\>npm -v

2. Download and Install "Visual Studio Code" editor

https://code.visualstudio.com/

3. Download and Install following extentions for VS-Code

- Live Server

- vscode-icons

- Intellisense for css class names

**Create a new Project - [Web Application]**

1. Create a new folder on your PC for project

E:\react-web-app

2. Open the folder in your Editor VS-Code

3. Open Terminal

Terminal Menu => New Terminal [Ctrl + ` ]

4. Run the following command

>npm init -y

5. Add following folder

- public : It comprises of static resources: html, image, docs ...

- src : It comprises of dynamic resources: .js, .css, .scss, .ts etc.

6. Add a new file index.html

**index.html**  
  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<h1>Home Page</h1>

<a href="home.html">Home</a>

</body>

</html>

**7. create home.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Home</title>

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>

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

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

<script type="text/babel">

**const root = ReactDOM.createRoot(document.querySelector('p'))**

**root.render("Welcome to ReactJS",);**

</script>

</head>

<body>

<a href="index.html">Back</a>

<p></p>

</body>

</html>

**OLD way**ReactDOM.render("Welcome to old way reactjs",document.querySelector("p"));

------------------------------------------------------------

Install libraries using NPM and create react app  
------------------------------------------------------------

1. Go to visual studio and open terminal using ctrl + ~ and change PowerShell to command prompt.  
   E:\react-web-app>
2. Run the following commands

E:\react-web-app> npm install react

E:\react-web-app> npm install react-dom  
E:\react-web-app> npm install @bael/standalone

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src="../node\_modules/react/umd/react.development.js"></script>

<script src="../node\_modules/react-dom/umd/react-dom.development.js"></script>

<script src="../node\_modules/@babel/standalone/babel.js"></script>

<script type="text/babel">

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render('Welcome to ReactJS');

</script>

</head>

<body>

<h1 id="root"></h1>

</body>

</html>

--------------------------

**Javascript Module**

--------------------------

DemoModule.js

var username = "Suryakant";

export function welcome(){

return username;

}

Index.js

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src="./DemoModule.js"></script>

<script type="module">

import {welcome} from './DemoModule.js';

document.getElementById('root').innerHTML = welcome();

</script>

</head>

<body>

<h1 id="root"></h1>

</body>

</html>

Javascript module is a set of variables, functions, classes.

There are so many modules systems JavaScript uses:

**CommonJS** – by default web browser working on CommonJS

**UMD** - Universal Module Distribution

**AMD -** Asynchronous Module Distribution

**RequireJS** –

**Note:** React uses UMD module system

Create a new react application with webpack implicit

E:> npx create-react-app appName

7. For JavaScript you can add "EsLint".

JavaScript Versions

- ES5 [ECMA Script 2015]

- ES6

- ES7

....

- ESNext

To setup ESLint

> npm init @eslint/config