Introduction to React

- React JS is a Javascript Library.
- React JS is used to create User Interface

Challenges to Develop User Interface using HTML and JS

- Using HTML, We can create a User Interface in the easiest way but we cannot reuse the User Interface.
- Using HTML, We cannot create a Dynamic User Interface.
- Using Javascript, We can create a Dynamic User Interface and we can reuse the User Interface but it is very complex to create a User Interface.
- It is very time consuming process to create a User Interface
- Explicitly Developer need to take care to optimise the application to enhance the Performance

We can overcome the above Challenges using Library and Framework.

What is Library and its Advantages

Library is a Collection of Predefined Function or Classes or Properties.

What is Framework and its Advantages

Framework is a Collection of Libraries and a set of rules to develop the Application.

Difference between Library and Framework

One of the Library that helps us to Overcome the above Problem is React.

Why React JS is popular to create UI

- React is a simple and easy to learn
- React Learning curve is very small
- React Application are very fast in the performance
- React uses Virtual DOM concept
- React is supported by Facebook
- React is easy to Integrate With Different Libraries
- React has very huge Developer Community

How to add a React to an Html page

1. Create Html File
2. Add **<div id="root">** tag in the body section of html and assign id attribute
3. Add Following <script> tags to the HTML page right before the closing </body> tag

   <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.develop
ment.js"></script>
```

How to create a React Element

- Using **createElement()**, lets you create a React element.

Syntax:

createElement(type, props, ...children)

- **type:** The type argument must be a valid React component type. For example, it could be a tag name string (such as 'div' or 'span'), or a React component (a function, a class, or a special component like Fragment).
- **props:** The props argument must either be an object or null. If you pass null, it will be treated the same as an empty object. React will create an element with props matching the props you have passed. Note that ref and key from your props object are special and will *not* be available as element.props.ref and element.props.key on the returned element. They will be available as element.ref and element.key.
- **optional ...children:** Zero or more child nodes. They can be any React nodes, including React elements, strings, numbers, portals, empty nodes (null, undefined, true, and false), and arrays of React nodes.

Example:

let h1=createElement('h1',{ className: 'greeting' }, 'Hello’);

How to Add React Element to DOM

Using ReactDOM Package or Library or Module
ReactDOM, renders React elements to the DOM
ReactDOM.render(element, container, callback)

Parameters: This method can take a maximum of three parameters as described below.

- **element:** This parameter expects a JSX expression or a React Element to be rendered.
- **container:** This parameter expects the container in which the element has to be rendered.
- **callback:** This is an optional parameter that expects a function that is to be executed once the render is complete.

Example :

ReactDOM.render(h1,document.getElementById(‘root’))

ReactDOM is a package that provides DOM specific methods that can be used at the top level of a web app to enable an efficient way of managing DOM elements of the web page. ReactDOM provides the developers with an API containing the following methods and a few more.

- render()
- findDOMNode()
- unmountComponentAtNode()
- hydrate()
- createPortal()