**Session1**

**What is ReactJs?**

ReactJs is a free and open – source frontend – end javascript library.

**Library vs Framework?**

Inversion of Control:

* In framework, your appl., is under control
* In library, its both., your appl., is under control of library., at the same time the appl., code can also control the builtin methods.,

A framework is a foundation for your application

The framework itself, internally connected or having the library

The library is n no of builtin methods

**What is usage of ReactJs?**

1. It is used to build the Single Page Applications
2. SPA: loading the data into a particular selector without refreshing the page
3. The ReactJs is used for building ‘User Interfaces’ specifically for SPA

**How ReactJs Works?**

1. Reactjs is a declarative, efficient and flexible js library
2. Architecture., MVC: Model, View, Controller – Reactjs especially given preference to ‘V’

**Diff., between Javascript and ReactJs:**

1. Javascript preferably – DOM – Document Object Model
2. ReactJs – VDOM

**Diff., between DOM and VDOM**

**Real DOM**

1. It updates slow
2. Can directly update the HTML
3. Creates a new DOM if element updates
4. DOM manipulation takes time & expensive
5. Too much of memory wastage

**Virtual DOM**

1. It updates faster
2. Can’t directly update the HTML
3. Updates the JSX if element updates
4. DOM manipulation is very easier
5. No memory wastage

**Advantages of ReactJs:**

1. Speed in development of an appl.,
2. Performance levels increasing of your appl.,
3. Reusability components
4. Mobile app., dev., - React Native.,
5. Code of Reactjs works at Client & Server

**Env., Setup & Creating the Project:**

1. Install NodeJs & NPM: Node Package Manager
2. Check: > node –version && npm –version
3. Create a new ReactJs Project: > npx create-react-app react-app
4. Cd react-app && npm start
5. Localhost:3000

Component: Components are **independent and reusable bits of code**. They serve the same purpose as JavaScript functions, but work in isolation and return HTML. Components come in two types, Class components and Function components, in this tutorial we will concentrate on Function components.

**Working with Html’s, Css – [external, embedded & inline], Javascripts, Images – Assets**

**Session2**

**Adding Bootstrap to the Pages:**

1. CDN – Url
2. Installing the Package

**Routing:** > npm install react-router-dom

BrowserRouter, Routes, Route, NavLink

Creating the Group of Items & Displaying

Props & State

To update the data: setState()

Map() – mapping all the rcords & displaying

Call to methods & passing parameters to methods

Two way property binding

**Session3**

Form Validation with Regular Expressions

FileUpload and Preview

RestApi

**Session4**

RestAPI

Life Cycle Hooks: common methods which can help your appl., to run / execute automatically when the situations

Class Based Life Cycle Hooks:

componentDidMount(), componentDidUpdate(), componentWillUnmount()

fetching the data from the given resource url (restapi)

accessing / displaying

adding the data

updating the data

deleting the data

package: node-uuid, axios

**Session5**

Functional Based Life Cycle Hooks

Conditional Rendering

Simple Todo Appl.,

**Session6**

Redux - ex., Counter

Redux is an open source javascript library for managing and centralizing appl., state.

React Redux is the React binding for Redux

It allows the React Components to read the data from a Redux Store, and dispatch Actions to the Store to update data.

1. UI Components
2. Actions
3. Reducers
4. Store

Npm install react-redux redux

React – Material UI – Crud Operations – RestAPI (axios)