**React JS Training Contents – 2.5 days**

**Pre-requisites**

Sound experience in the following is mandatory

1. HTML5
2. CSS, CSS3
3. JavaScript
4. Server-side programming (Java,Asp.net, Php, Python or Node JS(Express JS)) (Optional)
5. Participants are assumed to know basics about React.js

**Course Outline**

**Day 1 – 8 hours**

**Introduction to TypeScript [Quick Overveiw]**

* Introduction
* Limitation of JavaScript
* What is TypeScript
* Benefits of TypeScript
* Setup the environment
* First typescript code

**React Components**

* Creating React App with ES5
* Creating React App with ES6 features
* Creating React App with typescript
* React component
* Render function
* Component API
* Component Life cycle
* State
* Props
* Mixins
* JSX

**Components inter communication**

* Component composition
* Pass data from parent to child
* Pass data from child to parent

**React Event**

* Event introduction
* Types of event
* Child event
* Types of event
* Child event

**React JS – keys**

* Introduction to React JS keys
* Creating the component using keys
* Dynamic creation and remove component using keys

**Day 2 – 8 hours**

**React Hook with Functional Style with TypeScript**

**React Hook with Functional Style with ES6**

**React JS – Routers**

* What's in a URL
* React Router's core component
* Building the component of react-router
* The building Match
* The building Link
* The building Router
* The building Redirect
* Using react-router
* More match
* Dynamic routing with React Router
* The completed app
* Using URL params
* Supporting authenticated routes
* Implementing Login
* MatchWhenLoggedIn, a higher-order component

**React JS – With Rest API**

* Introduction to Web Service
* Type of Web Service
* SOAP Vs REST API
* REST API with expose the Resources
* GET, POST, DELETE and UPDATE
* Calling the Rest API using React JS using fetch and axios

Node JS CRUD Operation : get, post, delete, update()

Without database.

JS file which hold data.

**Into to Redux**

* Intro to Redux
* Redux key ideas
* The core of Redux
* Actions
* State
* Actions
* Building the reducer()
* Initialize state
* Handling ADD\_OPERATION
* Handling REMOVE\_OPERATION
* Handling UPDATE\_OPERATION
* Handling RETRIEVE\_OPERATION
* initialState()
* createStore()
* using store getState()
* using store subscribe()
* using store dispatch()

**Day 3 – 4 hours**

**Next JS**

* Intro to Next JS
* Features provided by Next JS
* Next JS vs create-react-app
* How to install the Next.js
* View source to confirm SSR is working
* The app bundles
* Install the React Dev tools
* Linked two pages
* Dynamic content with the router
* Prefetching
* Lazy loading modules