**ReactJs Documentation**

**What is ReactJs?**

ReactJs is an open-source frontend Javascript Library.

Used to build ‘User Interfaces’ using ‘UI Components’ and to make it an ‘Single Page Applications’.

It is maintained by Meta and a community of individual developers and companies.

React can be used as a base in the development of SPA.

**MVC:** Model ‘View’ Controller

It is used for handling the ‘view’ layer for ‘Web’ and ‘Mobile’ (React Native)

How React works?

React is a Declarative (<User />), Efficient and Flexible JS Library for building UI’s.

It’s ‘V’ a view layer in MVC.

Comparing to JS, how React is differing?

Js: DOM (Document Object Model)

React: VDOM (Virtual DOM)

Comparing the Library and Framework?

Inversion of Control

Packages – Modules – Functions

Setup an Env., and Creating a new project in React:

Node –version

Npm –version

Npm install -g create-react-app

Npm create-react-app <yourappName> (or)

Npx create-react-app <yourappName> (or)

Create-react-app <yourappName>

We even added the Htmls, Css (External & Inline), Declared the Variables and set the Properties into that, worked with JS’s expr.,

Created the sample reactjs webpage(s) using html & css

= Created the sample reactjs webpage(s) using html & css & bootstrap

Segregating the Page & its corresponding Components

Implemented the Routing Concept

Added the Bootstrap

Started with Dynamic Page: ex., Products.,

Listing out of all products,

Added the Properties at Individual Component Level and accesed into the Def., of the Component

Later, Created the ‘state’ object, and maintained all the required properties and accessed into the return statement with its indexing

Event Handling:

Created the Toggling Effect,

Updated the Records through Button & ListItem

Deleted the Records

Mapping

One Way: Displaying the Value Directly from the Given Property

Two Way Property Binding: User Interaction, and that value can be manage

Forms & Validation using Regular Expression

Ref’s

RestApi

TodoApp

Cont., with TodoApp

Class Based Life Cycle Hooks – RestApi – TodoApp – Fetch, Display, Delete, Update, Add

Functional Life Cycle Hooks – RestApi – Photos

Cont., Functional Life Cycle Hooks – RestApi – Photos

Styled Components

ContextApi