

## MERN Stack:

(MongoDB + Express JS + React JS + NodeJS)

### Objective and Manifest

- *MERN* is a latest technology stack which uses **React JS** for UI, **Node.js** for Server Side, **Express.js** for RestFul Web Services and **Mongo DB** as a NO-SQL Database. At the end of this course, a participant is expected to integrate all the technologies and create an end to end application.

### Understanding & Using ReactJS

- ✓ What is ReactJS and why to use it?
- ✓ React Js Application Development
  - Project Configuration using WebPack.
  - Module Loader
  - Using Web Pack for Images, CSS
    - Understanding WebPack
    - Installing required modules for application development
- ✓ Understanding ES6 Features for coding (HOL)
  - Let vs Var
  - Rest & Spread
  - Arrays
  - Arrow Functions
  - Classes
  - Modules
  - Promises with Asyn
- ✓ React Js Features (HOL)
  - Component
    - The component class and its properties
    - Creating Component Using ReactJS APIs
    - CreateClass, CreateElement, etc.

- Define HTML Element
- Render Method
- Defining Component Properties
  - State and Props
  - Applying Properties for the component from Root Render method
- ✓ Understanding JSX (HOL)
  - Creating Elements in the Component Class
  - Defining Properties for the Element.
- ✓ React Events (HOL)
  - DOM Events
  - Event and Property Binding using Bind function
  - Synthetic Events

## Using React Forms

- ✓ Using React.js Forms (HOL)
  - Using Form Element
  - Managing Submit
  - Controlled Components
  - Component Lifecycle
    - Methods invoked while loading a component
    - Methods invoked on re-rendering a component
    - Methods invoked while unloading a component
  - Un-Controlled Components
    - Using Refs
  - Validations
- ✓ Using Parent Child Components
  - Passing data across Components
  - Parent/Child Events.

## React & Redux

- ✓ Developing Single Page Application using React (HOL)
  - React Router
  - Path Matching
  - Routing with Multiple Components
  - Managing Ajax Calls
- ✓ Understanding Redux
  - Store for managing Application State and Event Broadcasting
  - Action for Creating Action, Dispatching Action, Action Binders
  - Reducer for Creating and combining reducers
  - Injecting Redux Store using Providers

## Mongo DB

- ✓ Introduction to the NO-SQL Databases
- ✓ Data Modeling
- ✓ Create & Drop Database
- ✓ Create & Drop Collection
- ✓ Data Types
- ✓ Insert Document
- ✓ Query Document
- ✓ Update Document
- ✓ Delete Document
- ✓ Projection
- ✓ Limiting Records
- ✓ Sorting Records
- ✓ MongoDB – Indexing
- ✓ Object Id
- ✓ Map Reduce

## Express.js

- ✓ Introduction to the Express.js Framework
- ✓ Installing the Express package
- ✓ Routing
- ✓ HTTP Methods (Get, Put, Post, Delete)
- ✓ URL Building
- ✓ Form Data
- ✓ ExpressJS – Body-Parser to read data from incoming requests
- ✓ Database
- ✓ Cookies
- ✓ Sessions
- ✓ Authentication
- ✓ RESTful APIs
- ✓ ExpressJS & Mongo DB Integration

## Node.js

- ✓ Introduction to the Node.js Framework
- ✓ Installing Node.js
- ✓ Overview of Blocking vs Non-Blocking IO
- ✓ Using Node.js to execute scripts
- ✓ The Node Package Manager
- ✓ The package.json configuration file
- ✓ Using & compiling Typescript for Node.js
- ✓ Working with the HTTP protocol
- ✓ Building an HTTP Server
- ✓ Accepting a request
- ✓ Rendering a response
- ✓ Using Representational State Transfer Protocol (REST)
- ✓ Understanding concept of Web-Services
- ✓ Introduction to REST Full W/S