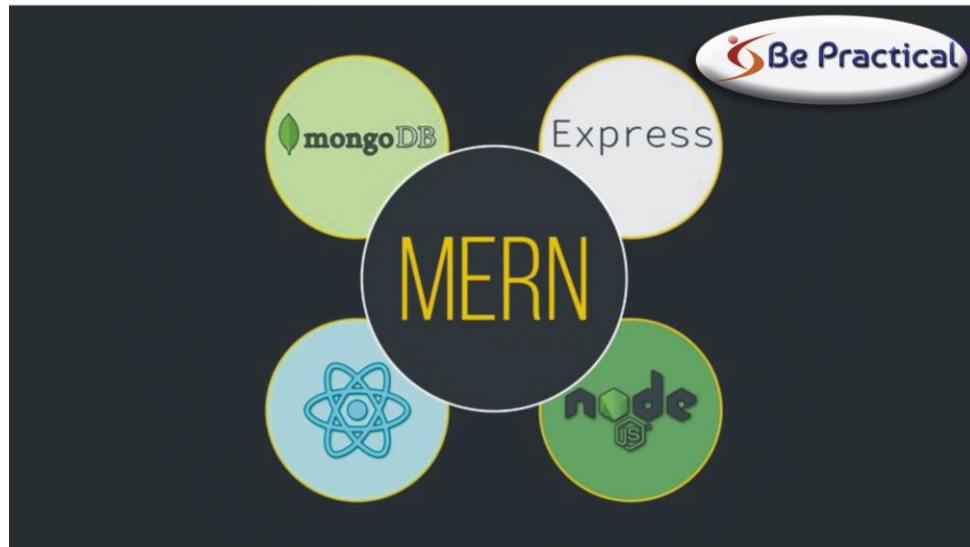


MERN FULLSTACK DEVELOPMENT



- Introduction
- HTML Attribute
- HTML5
- HTML API

CSS

- CSS Introduction
- CSS selector
- CSS Declaration
- CSS Properties
- CSS Combinators

CSS3

- CSS3 Animation
- CSS3 Design
- Layout
- Table

Bootstrap

Module 1:- Introduction to Bootstrap
Module 2:- Bootstrap Grid
Module 3:- Bootstrap Components
Module 4:- Bootstrap Plug-Ins

JAVASCRIPT

- JavaScript Basics
- JavaScript Objects
- JavaScript BOM
- JavaScript DOM
- JavaScript Validation
- JavaScript OOPs
- JavaScript Cookies
- JavaScript Events
- JavaScript Misc
- JavaScript Advanced

JQUERY

- Introduction
- jQuery Effects
- jQuery HTML / CSS
- jQuery Form
- jQuery Events

jQuery-UI

- JQ UI Introduction
- jQuery Interactions
- jQuery Widgets
- jQuery Effects

Typescript

- Introduction
- Typescript Environment Setup
- First Program
- Type Annotation
- Variables
- Datatypes
- Type inference
- Type Assertion
- If
- Switch
- For loop
- While loop
- Function
- Typescript Interface
- Classes

- Abstract Classes
- Data modifiers
- ReadOnly
- Module
- NameSpace
- Generic

ECMA Script (ES6/ES7)

- Introduction
- The For Each Helper
- The Map
- Filter Helper
- The every and Some helper
- Const and Let
- Template Strings
- Arrow Functions
- Enhanced Object Literals
- Default function arguments
- Rest and Spread
- Destructuring
- Classes
- Generators
- Promises and Fetch



BabelJS

- What is Babel.js?
- Environment setup
- Babel CLI Installation
- Project setup using Babel 6
- Project setup using Babel 7
- ES6 features to ES5
- ES6 Modules to ES5
- Transpile ES7 features to ES5
- Transpile ES8 features to ES5
- Babel Plugins
- Babel Polyfill
- Babel CLI
- Babel Presets
- Working with babel and Webpack

React JS

- Getting Started
 - Installation steps
 - What is React?
 - Real world SPA v/s React
 - Creating First Project
 - Building a Project

- Installing Dependencies
 - React Alternatives
 - Course Outline
 - Useful resources and Link
- Refreshing JavaScript
 - Understanding “let” and Const
 - Functions
 - Exports and Imports
 - Understanding Classes
 - Class Props and Methods
 - Spread
 - Map
 - Filter
 - Class Object
 - Interfaces
 - Subclasses
 - Array
- Understanding Base features and Syntax
 - Creating react app
 - Using create-react-app
 - Understanding component basics
 - Understanding JSX
 - JSX Restrictions
 - Creating Components
 - Working With Components
 - Reusing Components
 - Outputting Dynamic Content
 - Working With Props
 - Understanding Children Property
 - Props and State
 - Handling Events and Methods
 - Manipulating State
 - Functional Components naming
 - Stateless and State full components
 - Passing Methods References between components
 - Adding Two way data binding
 - Working with Styles
- Working with List and Conditionals
 - Module Introduction
 - Rendering modules conditionally
 - Handling dynamic contents
 - Outputting lists
 - List and State
 - Lists and Keys
 - Flexible Lists
 - Practices
- Styling React Components
 - Module Introduction

- Outlining the problem set
- Setting Styles Dynamically
- Setting Class names Dynamically
- Enabling css modules
- Adding Pseudo selectors
- Working with media Queries
- Components
 - Module Introduction
 - Splitting app into Components
 - Comparing Stateless and Stateful Components
 - Class Components (ES6)
 - Functional Components
 - Lifecycle Methods (Functional Components)
 - Lifecycle Hook in Class Components
 - Creating Default Props
 - Validating Prop Types
 - Higher Order Components
 - Another form of Higher Order Components
 - Using refs
 - Using the Context API
 - Handling Providers
- Music Plyer Project – Spotify API
 - Project Introduction
 - Project outline
 - Project Creation
 - Building Search Component
 - Handling API response from server
 - Consuming API response to console
 - Display API response in Artist Component
 - Display Tracks list in Tracks Component
 - Finalizing Project
- Custom Resume Builder – Accessing Local API
 - Project Introduction
 - Project outline
 - Project Creation
 - Building Resume Component
 - Creating Rest API using JSON Server
 - Consuming API response to console
 - Display API response in Components
 - Assigning Props to various components
 - Finalizing Project
- Routing
 - Module Introduction
 - Building Checkout Container
 - Setting up Routing and Routes
 - Navigating between components
 - Navigating Back & to next page
 - Implementing Children navigation

- Implementing Component based navigation
- Forms
 - Analysing Component
 - Creating Custom Dynamic Input Component
 - Setting Up event and state for the form input
 - Handling user input
 - Adding Custom form Validation
 - Working with Refs
- Redux
- CRUD Application Project – Accessing API
- Accessing JSON Responses using Axios
- Deploying

NodeJS

- Introduction
 - What is NodeJS?
 - Installation of Node js
- Core Fundamentals
 - The Global Object
 - The Process Object
 - Custom Events
 - Modules
 - Exporting
 - Child Process
- Core Modules
 - Assert
 - Buffer
 - Cluster
 - Crypto
 - Dns
 - Fs
 - http
 - net
 - os
 - path
 - punycode
 - querystring
 - timer
 - tls
 - tty
 - url
 - util
 - v8
 - vm
 - zlib

- Working with NoSQL database (MONGO)
 - How to Use mongodb
 - Installing mongodb on windows
 - Installing Mongodb module as global package using npm
 - Connecting mongodb using mongodb client
 - Mongodb client – inserting data
 - Reading all data from mongo db.
 - Updating data in mongodb
 - Deleting data from mongo db.
- Working with Relational Databases (MySQL)
 - Installation of MySQL
 - Connecting to MySQL
 - Inserting values into db.
 - Reading values from db.
 - Updating values of db.
 - Deleting values of db.
- Working with Buffers
 - Introduction
 - Creating buffers
 - Writing into buffers
 - Reading from buffers
 - Convert buffer to JSON
 - Concatenate Buffers
 - Compare Buffers
 - Copy Buffer
 - Buffer Length
 - Class Methods
- Node JS Streams
 - Introduction
 - Reading data from file using read streams
 - Writing data into the file using write streams
- Advance NodeJS
 - Handling Web Request
 - Building web server
 - Understanding the need of web sockets
 - Real time interaction
 - Routing
 - Parameters and queries in Routing
 - Building views using views engine
 - Displaying data
 - Serving Files
 - Working with cookies and Sessions
 - Authentication and Authorization
 - Exception Handling
 - The Child Process Model
 - NodeJS Inspector

ExpressJS

- Introduction
 - What is ExpressJS
 - Installation of ExpressJS
- Core Fundamentals
 - Variables
 - Constants
 - Global Variables
 - Functions
 - Modules
 - Exports
- Routing
 - Introduction
 - Route Methods
 - Route Paths
 - Route Parameters
 - Route Handlers
 - Response methods
 - End()
 - Download()
 - Json()
 - Jsonp()
 - Redirect()
 - Render()
 - Send()
 - sendFile()
 - sendStatus()
 - app.route()
 - express.Router
 - Practical Implementation
- Writing Middleware
 - Introduction
 - Middleware function myLogger
 - Middleware function requestTime
 - Configurable Middleware
- Using Middleware
 - Introduction
 - Application level Middleware
 - Router-level Middleware
 - Error-Handling in Middleware
 - Built In Middleware
 - Third Party Middleware
- Using Template engines
 - Introduction
 - Working with template engines
 - Pug
 - EJS

- React
- Error Handling
 - Introduction
 - Catching Errors
 - The Default Error Handler
 - Writing Error Handlers
- Debugging
- CRUD Application
 - Mongodb configuration
 - Creating Collection
 - Setting up module and Template engine
 - Project Overview
 - Installation of Mongodb, express, ejs modules
 - Creating of Views Directory
 - Updating UI of template engines
 - Configuring the router paths
 - Handling data from form submission
 - Displaying data into template engine
 - Updating the data
 - Deleting the data
- REST API
 - Introduction of REST API
 - Overview
 - Installing Dependencies
 - Creating Module
 - Configuring router paths
 - Testing API with local REST Client

HOSTING with live Projects

COURSE DURATION: 3 Months