



**ReactJS**

## OVERVIEW

- NodeJS is an open-source system which has been developed for developers to comprehend and build web applications with the use of JavaScript. NodeJS uses an event-driven, non-blocking IO model and server-side JavaScript environment. NodeJS is focused on a few significant ideas in it like file system working, streams, buffers and building HTTP worker with NodeJS.
- Node JS is basically an extension of the JavaScript programming for server-side programming. With the Node JS features JavaScript can be presently utilized for development of the complete web site both front end and backend

### Course Duration

5 Days (10 Half Days)

## Target Audience

- This course is appropriate for web developers, programmers, full-stack developers and backend developers.

## COURSE CONTENT

- **Introduction to NodeJS**
  - Introduction to Node.js
  - Applications of Node.js and installation
  - Writing your first Node.js Script
  - Synchronous and Asynchronous programming
  - Under the hood - understanding the event loop and Non-Blocking I/O
- **Interactive Node with REPL**
  - REPL and REPL Commands
  - Node.js CLI Commands
  - Global and Local Objects
- **Modular Programming and NPM**
  - Introduction to Modules
  - Process and OS Modules
  - Writing Your Own Module
  - Introducing NPM
  - Initializing a Project Using npm init
  - Understanding package.json
  - Installing and using packages from npm
  - Local vs.Global Install
  - Using NPX
  - Publishing Package on npm
- **Introduction to ECMAScript Modules**
  - ECMAScript Modules Versus CommonJS Modules
  - Enabling support
  - Writing an ECMAScript Module
  - Importing and Using an ECMAScript Module
  - ES6 syntax in detail

- **File System and Streams**
  - File System Modeling in Node.js
  - Directory and Path Resolution
  - Reading Files Synchronously and Asynchronously
  - Writing Files Synchronously and Asynchronously
  - Directory operations
- **Streams**
  - Understanding Streams
  - Reading a File Using Readable Stream
  - Writing a file using Writable Streams
  - Transform Streams
  - Streams vs Files
- **Events**
  - Event Emitter
  - Handling Events
- **Network I/O**
  - Introduction to the Net module
  - Creating a TCP Server and Listener
  - Creating a command-line chatbot
- **Web Servers and More**
  - Web Server: Web Application Architecture
  - HTTP module
  - Making HTTP requests
  - Serve JSON as a response
  - Serve HTML as a response
- **Debugging in NodeJS**
  - Introduction to debugging
  - Debug node js apps
  - Adding a basic web server with routes and streams: Serve a static website
  - Using HTTPS
  - Using HTTP/2
  - Understanding CORS
- **Cluster and Worker Threads**
  - Introduction to the Worker Threads API
  - Handling compute-intensive tasks using Worker Threads
  - Node.js on multi-core CPUs using Cluster
- **Introduction to Express**
  - What is a Node.js web application framework?
  - BYOF - Build Your Own Framework
  - Introducing Express
  - Who uses Express?

- **Hello Express**
  - Build your first Express app
  - Learn adding a route handler
  - Serving Static Assets and HTML files
- **Rendering**
  - View Engine – Pug
  - Building the Dashboard
  - Building the Post Card
- **Middleware**
  - What is middleware?
  - Parsing incoming requests
  - Logging with Morgan
  - Using the express-session middleware
  - Compression and Your own middleware
- **Routing**
  - Modular routing with Express Router
  - Organizing routes
  - Dynamic Routing and Route parameters
  - Basic route protection using middleware
- **Database Integration**
  - Working with MongoDB Atlas
  - Setting up Mongoose ODM
  - Building the Users schema and model
  - Building the Posts schema and model
  - Serving the client web application
- **Authentication, Controllers, and APIs**
  - Signing up an Admin and Password Hashing
  - Admin Authentication - Controller and Route
  - Authentication API - Controller and Routes
  - Posts API - Controller and Routes
  - Profanity Filtering and Post Moderation
- **Deployment**
  - Learn to deploy the Node and Express app on Heroku.