

## Node JS Course

**Course Duration: 40 Hours**

**Course Code: NODE-401**

### 1. Course Overview

The *Node.js Course* is designed to help learners gain expertise in **server-side JavaScript development**. This course covers everything from **Node.js fundamentals to building scalable backend applications**, including working with **Express.js, REST APIs, databases, authentication, and deployment**. By the end of this course, learners will be able to develop, test, and deploy **real-world full-stack applications** using Node.js.

### 2. What You'll Learn?

- Fundamentals of **Node.js and the event-driven architecture**.
- Building **server-side applications** with Express.js.
- Creating and consuming **RESTful APIs**.
- Integrating Node.js with **databases** (SQL & NoSQL).
- Handling **authentication and security** in Node apps.
- Working with **asynchronous programming (callbacks, promises, async/await)**.
- Real-time applications using **WebSocket's**.
- Deploying Node.js applications to the **cloud**.

### 3. Target Audience

- Beginners who want to start backend development with **Node.js**.
- Frontend developers looking to become **full-stack developers**.
- JavaScript developers interested in **server-side programming**.
- Professionals aiming for **Node.js developer roles**

### 4. Pre-Requisites

- Basic knowledge of **JavaScript, HTML, and CSS**.
- Familiarity with ES6+ features.
- Understanding of web fundamentals (HTTP, APIs, JSON).

### 5. Course Content

#### **Module 1: Introduction to Node.js**

- What is Node.js and why use it?
- Node.js architecture and event loop
- Installing Node.js and npm
- Creating your first Node.js application

#### **Module 2: Node.js Fundamentals**

- Core modules: FS, HTTP, Path, Events, OS
- Asynchronous programming in Node.js
- Working with streams and buffers
- Handling errors in Node.js

#### **Module 3: Building Web Servers with Express.js**

- Introduction to Express.js framework
- Creating REST APIs with Express
- Middleware functions
- Handling requests and responses

#### **Module 4: Databases with Node.js**

- Connecting Node.js with MongoDB (Mongoose)
- CRUD operations in MongoDB
- Working with SQL databases (MySQL/PostgreSQL)
- Designing data models for applications

#### **Module 5: Authentication & Security**

- User authentication with JWT and OAuth
- Password hashing with bcrypt
- Implementing session management
- Securing APIs with middleware

#### **Module 6: Advanced Node.js Concepts**

- Working with WebSockets and Socket.io (real-time apps)
- File uploads and handling multipart data
- Using environment variables with dotenv
- Performance optimization techniques

#### **Module 7: Testing & Debugging**

- Debugging Node.js applications
- Unit testing with Mocha/Chai/Jest
- API testing with Postman

#### **Module 8: Node.js with Frontend Frameworks**

- Integrating Node.js APIs with Angular, React, or Vue



- Serving static files and single-page apps
- Full-stack project structure

### **Module 9: Deployment & DevOps**

- Preparing apps for production
- Deploying Node.js apps on Heroku, AWS, or Azure
- Using PM2 for process management
- CI/CD for Node.js applications

### **Module 10: Capstone Project**

- Building a full-fledged application using Node.js, Express, and MongoDB
- Implementing authentication, CRUD, and API integrations
- Deploying and showcasing the project