

# **Node JS Course Content**

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

- What is Node.js?
- Features and advantages of Node.js
- Understanding asynchronous programming
- Installing Node.js
- Setting up the development environment

## **2. Node.js Architecture**

- Event-driven, non-blocking I/O
- Single-threaded nature of Node.js
- Understanding the Node.js event loop

## **3. Working with Node.js Modules**

- What are modules?
- Built-in modules (e.g., fs, http, path)
- Creating custom modules
- Using require() to import modules
- Understanding module.exports and exports

## **4. File System Module**

- Reading and writing files in Node.js
- Asynchronous vs. synchronous file operations
- File system methods (readFile, writeFile, appendFile, etc.)
- Working with directories and file streams

## **5. Working with HTTP**

- Creating a basic HTTP server
- Handling requests and responses
- Working with URL routing
- Serving static files (HTML, CSS, JavaScript)
- Understanding HTTP methods (GET, POST, PUT, DELETE)

## **6. NPM (Node Package Manager)**

- Introduction to npm
- Installing and managing packages
- Understanding package.json

- Global vs local packages
- Using npm scripts

## **7. Express.js Framework**

- Introduction to Express.js
- Setting up a basic Express application
- Creating routes and handling requests
- Middleware in Express.js
- Handling form data and file uploads
- Working with query parameters and URL routing

## **8. Working with Databases**

- Introduction to databases (SQL vs NoSQL)
- Connecting Node.js to a MongoDB database using Mongoose
- Performing CRUD operations (Create, Read, Update, Delete)
- Querying the database
- Handling database connections and errors

## **9. RESTful API with Node.js**

- Introduction to REST architecture
- Creating RESTful routes using Express
- Working with HTTP verbs (GET, POST, PUT, DELETE)
- Handling JSON data
- Building a REST API with Node.js and Express

## **10. Asynchronous Programming in Node.js**

- Understanding callbacks
- Promises in Node.js
- async and await for handling asynchronous code
- Error handling in asynchronous operations

## **11. Real-time Communication with WebSockets**

- Introduction to WebSockets
- Setting up WebSocket communication using Socket.io
- Creating real-time applications (e.g., chat apps)
- Broadcasting events to multiple clients

## **12. Authentication and Security**

- Implementing user authentication with JWT (JSON Web Tokens)
- Password hashing with bcrypt
- Handling sessions and cookies
- Securing applications against common vulnerabilities (e.g., XSS, CSRF, SQL Injection)

## **13. Error Handling and Debugging**

- Error handling in Node.js
- Using try-catch and error-first callbacks
- Debugging Node.js applications
- Logging errors using winston or morgan

## **14. Testing in Node.js**

- Introduction to unit testing
- Writing test cases with Mocha and Chai
- Testing asynchronous code
- Test-driven development (TDD) basics

## **15. Deploying Node.js Applications**

- Preparing Node.js applications for production
- Deployment platforms (Heroku, DigitalOcean, AWS, etc.)
- Using environment variables
- Scaling Node.js applications with PM2 and clustering
- Introduction to Docker and containerization