

---

# NODE JS

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

### **1.1. What is Node.js?**

- 1.1.1. Introduction to Server-Side JavaScript
- 1.1.2. Node.js vs. Traditional Server

Technologies

### **1.2. Setting Up the Development Environment**

- 1.2.1. Installing Node.js and npm
- 1.2.2. Creating Your First Node.js Application

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

### **2.1. Node.js**

- 2.1.1. Creating and Using
- 2.1.2. Core vs Custom

### **2.2. Asynchronous JavaScript**

- 2.2.1. Understanding Callbacks and Event Loop
- 2.2.2. Promises and Async/Await in Node.js

## **3: File System and Streams**

### **3.1. File System Operations**

- 3.1.1. Reading and Writing Files
- 3.1.2. Working with Directories

### **3.2. Streams in Node.js**

- 3.2.1. Readable and Writable Streams
- 3.2.2. Piping and Transforming Streams

## **4: Node.js Web Servers**

### **4.1. Building a Basic HTTP Server**



- 4.1.1. Handling HTTP Requests and Responses
- 4.1.2. Serving Static Files with Node.js

## **4.2. Express.js Framework**

- 4.2.1. Introduction to Express.js
- 4.2.2. Building RESTful APIs with Express

## **5: Node Package Manager (npm)**

### **5.1. Working with npm**

- 5.1.1. Installing and Managing Packages
- 5.1.2. Creating and Publishing npm Packages

### **5.2. Package.json and Dependency Management**

- 5.2.1. Managing Dependencies in Package.json
- 5.2.2. Semantic Versioning (Semver)

## **6: Working with Databases**

### **6.1. Using Relational Databases**

- 6.1.1. Connecting to MySQL, PostgreSQL, etc.
- 6.1.2. Performing CRUD Operations

### **6.2. Using NoSQL Databases**

- 6.2.1. Connecting to MongoDB, Redis, etc.
- 6.2.2. Working with NoSQL Data

## **7: RESTful API Development**

### **7.1. API Design Principles**

- 7.1.1. RESTful API Guidelines
- 7.1.2. Structuring API Endpoints

### **7.2. Express.js and API Routing**

- 7.2.1. Creating RESTful Routes
- 7.2.2. Handling JSON Data and Responses



## **8: Authentication and Authorization**

### **8.1. User Authentication**

- 8.1.1. User Registration and Login
- 8.1.2. Token-Based Authentication

### **8.2. Authorization and Access Control**

- 8.2.1. Role-Based Access Control
- 8.2.2. Middleware for Authentication and Authorization

## **9: Real-Time Applications with Node.js**

### **9.1. WebSockets and Socket.io**

- 9.1.1. Real-Time Communication Concepts
- 9.1.2. Integrating Socket.io with Node.js

### **9.2. Building a Chat Application**

- 9.2.1. Implementing Real-Time Chat Functionality
- 9.2.2. Broadcasting Messages

## **10: Security and Best Practices**

### **10.1. Security Considerations**

- 10.1.1. Common Security Vulnerabilities
- 10.1.2. Protecting Against Attacks

### **10.2. Best Practices**

- 10.2.1. Code Quality and Style Guidelines
- 10.2.2. Performance Optimization

## **11: Testing and Debugging**

### **11.1. Unit Testing with Mocha and Chai**

- 11.1.1. Writing and Running Tests



- 11.1.2. Mocking Dependencies in Tests

## **11.2. Debugging Node.js Applications**

- 11.2.1. Debugging Techniques and Tools

- 11.2.2. Troubleshooting Common Issues



+91 8618501719



[www.virajetech.com](http://www.virajetech.com)



info@virajetech.com