



TECHNOCATION FREELANCING TRAINING INSTITUTE & SOFTWARE HOUSE

# Professional Node.js Course Outline

## Module 1: Introduction to Node.js

- What is Node.js?
  - Understanding Node.js Architecture & Event Loop
  - Node.js vs. Traditional Backend Technologies
  - Installing Node.js & npm
  - Running JavaScript in Node.js
- 

## Module 2: Core Concepts & Modules

- Understanding CommonJS & ES Modules (`require`, `import/export`)
  - Built-in Node.js Modules (`fs`, `path`, `os`, `events`, `crypto`)
  - Creating & Using Custom Modules
  - Asynchronous Programming in Node.js (`callbacks`, `promises`, `async/await`)
  - Handling Errors (`try-catch`, `error-first callbacks`)
- 

## Module 3: File System & Streams

- Reading & Writing Files (`fs.readFile()`, `fs.writeFile()`)
  - File System Operations (`fs.rename()`, `fs.unlink()`)
  - Understanding Streams & Buffers
  - Working with Readable & Writable Streams
  - Using `zlib` for File Compression
- 

## Module 4: Building Web Servers with Node.js

- Understanding HTTP Protocol & Request/Response Lifecycle
  - Creating a Simple HTTP Server (`http.createServer()`)
  - Handling Routes & Query Parameters
  - Serving Static Files with Node.js
  - Using `nodemon` for Auto-Restarting Server
- 

## Module 5: Express.js – The Popular Node.js Framework

- Introduction to Express.js & Why Use It
  - Setting Up an Express Server
  - Working with Middleware (`req`, `res`, `next`)
  - Handling Different HTTP Methods (`GET`, `POST`, `PUT`, `DELETE`)
  - Using Express Router for Modular Code
- 

## Module 6: Working with Databases in Node.js

### Relational Databases (SQL & PostgreSQL)

- Connecting Node.js to PostgreSQL/MySQL
- Performing CRUD Operations with Sequelize/Knex.js

### NoSQL Databases (MongoDB & Firebase)

- Connecting Node.js to MongoDB (`mongoose`, `MongoClient`)
  - Designing Schemas & Models with Mongoose
  - Performing CRUD Operations in MongoDB
- 

## Module 7: Authentication & Security

- Understanding Authentication vs. Authorization
  - Implementing JWT-Based Authentication (`jsonwebtoken`)
  - Using OAuth with Google, Facebook, GitHub Login
  - Securing API Routes (`bcrypt`, `helmet`, `cors`)
  - Rate Limiting & Preventing API Abuse
- 

## Module 8: RESTful APIs & GraphQL in Node.js

- Designing RESTful APIs with Express
  - Handling API Responses & Error Handling
  - Implementing CRUD Operations in a REST API
  - Introduction to GraphQL & Setting Up Apollo Server
  - Querying & Mutating Data with GraphQL
- 

## **Module 9: Real-Time Applications with WebSockets**

- Introduction to WebSockets & Socket.io
  - Setting Up a Real-Time Chat Application
  - Broadcasting Events & Notifications
  - Implementing WebRTC for Video Calling
  - Optimizing WebSockets for Performance
- 

## **Module 10: Node.js & Microservices**

- Understanding Microservices Architecture
  - Building a Microservices-Based Application
  - Using RabbitMQ/Kafka for Event-Driven Communication
  - Implementing API Gateways with Express
  - Deploying Microservices with Docker & Kubernetes
- 

## **Module 11: Caching & Performance Optimization**

- Implementing Caching with Redis
  - Using Memory Caching for Faster Responses
  - Optimizing Database Queries in Node.js
  - Load Balancing with Node.js
  - Using PM2 for Process Management
- 

## **Module 12: Testing & Debugging in Node.js**

- Unit Testing with Jest & Mocha
- Integration Testing with Supertest
- Debugging Node.js Applications (`console.log`, `debugger`)
- Writing Mock Tests & Using Test Coverage Tools

---

## Module 13: Deploying Node.js Applications

- Deploying on AWS, Heroku, DigitalOcean, Vercel
- Setting Up CI/CD Pipelines (GitHub Actions, Jenkins)
- Using Docker to Containerize Node.js Apps
- Configuring Production Environment Variables
- Monitoring & Logging (Winston, PM2 logs)

### Real-World Projects & Hands-On Labs

- ✓ Building a RESTful API for an E-Commerce Platform
- ✓ Creating a Real-Time Chat Application with WebSockets
- ✓ Developing a Secure Authentication System with JWT
- ✓ Deploying a Scalable Node.js Microservices Architecture
- ✓ Implementing a GraphQL API for a Blogging Platform