**Full Course Content: Node.js, Express, MongoDB Bootcamp**

**1. Introduction to Node.js**

* What is Node.js?
* Installing Node.js
* The Node.js architecture and event-driven, non-blocking I/O model
* Running your first Node.js application
* Node.js REPL (Read-Eval-Print-Loop)

**2. Node.js Fundamentals**

* The Core Modules (fs, path, os, etc.)
* Working with the Node.js file system
* Event Emitters in Node.js
* Buffers and Streams in Node.js
* Handling Asynchronous Code (Callbacks, Promises, and Async/Await)
* Building your first simple HTTP server

**3. Introduction to Express.js**

* Setting up an Express.js application
* Understanding Middleware in Express
* Routing and handling requests
* Working with query strings and URL parameters
* Handling form data and file uploads
* Error handling in Express.js
* Serving static files

**4. Building RESTful APIs**

* Introduction to REST API principles
* Creating RESTful routes and controllers
* Handling CRUD operations (Create, Read, Update, Delete)
* Data validation and sanitization
* Managing error responses and status codes
* Pagination, sorting, and filtering dat
* API versioning strategies

**5. Introduction to MongoDB and Mongoose**

* What is MongoDB and why NoSQL?
* Installing and configuring MongoDB
* Connecting Node.js to MongoDB
* Introduction to Mongoose for MongoDB modeling
* Defining schemas and models in Mongoose
* CRUD operations using Mongoose
* Data validation with Mongoose

**6. Advanced API Features**

* Authentication and Authorization using JWT (JSON Web Tokens)
* User roles and permissions
* Securing APIs (rate limiting, CORS, helmet.js)
* Password hashing and encryption with bcrypt.js
* Protecting sensitive information
* Implementing 2-factor authentication (optional advanced)

**7. Application Architecture and Best Practices**

* Structuring Node.js projects (MVC pattern)
* Understanding middleware stacks in Express
* Separating concerns: controllers, routes, models
* Handling asynchronous errors with error middleware
* Configuring environment variables (dotenv)
* Logging and monitoring APIs

**8. File Uploads & Image Processing**

* Handling file uploads in Express.js
* Processing and storing images in MongoDB/GridFS
* Image resizing and optimization
* File validation and securite

**9. Real-Time Applications with WebSockets**

* Introduction to WebSockets
* Using Socket.io for real-time communication
* Creating a real-time chat or notification system
* Broadcasting messages in real-time
* Handling WebSocket events

**10. Testing Node.js Applications**

* Introduction to testing
* Writing unit tests for Node.js code
* Using Mocha and Chai for testing
* Testing API endpoints with Supertest
* Test-driven development (TDD) fundamentals
* Mocking external APIs and services

**11. Deployment and Production Setup**

* Preparing Node.js applications for production
* Environment configuration (staging, production)
* Using AWS,Azure for deploying Node.js apps
* Scaling Node.js applications with PM2 and Nginx
* Securing your Node.js application in production
* Monitoring and logging with tools like Morgan and Loggly

**12. Performance Optimization**

* Profiling Node.js applications for performance bottlenecks
* Optimizing Node.js event loop and memory usage
* Caching strategies (Redis/Memcached)
* Handling heavy traffic and load balancing
* Database indexing and query optimization in MongoDB

**13. Security Best Practices**

* Preventing common security vulnerabilities (XSS, CSRF, SQL Injection)
* Rate limiting and IP blocking
* Data sanitization and validation
* Secure session management
* Implementing HTTPS/SSL for secure connections

**14. Final Project: Full-Stack Web Application**

* Planning the project structure
* Building a real-world application combining Node.js, Express, MongoDB, and JWT-based authentication
* Implementing user registration, login, and authentication
* CRUD operations for managing data
* Deployment to a cloud service (AWS,Azure)
* Performance tuning and scaling

**Core Sections**

1. **Introduction to Node.js** – 5-7 hours
2. **Node.js Fundamentals** – 10-15 hours
3. **Introduction to Express.js** – 10-12 hours
4. **Building RESTful APIs** – 15-20 hours
5. **Introduction to MongoDB and Mongoose** – 15-20 hours
6. **Advanced API Features** – 15-20 hours
7. **Application Architecture and Best Practices** – 10-15 hours
8. **File Uploads & Image Processing** – 8-10 hours
9. **Real-Time Applications with WebSockets** – 8-10 hours
10. **Testing Node.js Applications** – 10-15 hours
11. **Deployment and Production Setup** – 10-12 hours
12. **Performance Optimization** – 8-10 hours
13. **Security Best Practices** – 10-12 hours
14. **Final Project: Full-Stack Web Application** – 20-25 hours

**Additional Topics**

1. **DevOps Basics for Node.js** – 10-15 hours
2. **GraphQL Basics** – 8-10 hours

**Total Estimated Time**

* **Core Topics**: Approximately **150-180 hours**
* **Additional Topics**: Approximately **18-25 hours**