

Table of Contents

Chapter 1: Introduction to Node.js

- What is Node.js?
- Why use Node.js? (Event-driven, non-blocking, single-threaded)
- Installing Node.js & npm
- Understanding the Node.js runtime and the V8 engine
- First Node.js script (console.log, creating a simple server)
- Understanding process, __dirname, __filename

Chapter 2: Node.js Core Modules

- File System module (fs)
- Path module (path)
- OS module
- Events module and EventEmitter
- http module – building basic HTTP server without Express
- Reading/Writing files asynchronously vs synchronously
- Using callbacks with fs

Chapter 3: Asynchronous Programming in Node.js

- Synchronous vs Asynchronous
- Callbacks in Node.js
- Callback hell and how to avoid it
- Promises: then, catch, chaining
- async/await usage and best practices
- Error handling with try/catch

Chapter 4: Getting Started with Express.js

- What is Express.js and why use it?
- Installing and setting up Express
- Basic Routing (GET, POST, PUT, DELETE)
- Handling request & response
- Middleware (application-level, router-level, built-in, third-party)
- Static file serving
- Route parameters and query strings
- Organizing Express code (separating routes, controllers)

Chapter 5: REST API Development with Express.js

- What is REST and RESTful APIs?
- Building CRUD APIs using Express.js
- REST API status codes and responses
- Postman walkthrough for testing APIs

- Creating custom middleware (e.g., logger)
- Validation using express-validator or custom checks
- Handling errors globally (Error-handling middleware)

Chapter 6: MongoDB with Mongoose

- What is MongoDB? How is it different from SQL?
- Installing MongoDB locally or using MongoDB Atlas
- Basic MongoDB commands (CRUD) using Mongo shell
- What is Mongoose? Why use it?
- Mongoose Setup & Connection with Node
- Defining Schemas and Models
- CRUD Operations using Mongoose
- Mongoose validations
- Relationships in MongoDB (References & Population)
- Indexes & performance optimization basics

Chapter 7: Real-life Project Building (Hands-On)

Functionalities to Cover:

- Add Operations
- Update Operations
- Delete Operations
- Get Operations
- Use MongoDB Atlas for cloud storage
- Validate all fields before saving
- Error handling and custom messages **Bonus Concepts:**
- MVC folder structure
- .env files and dotenv package
- Logger middleware (morgan or custom)

Chapter 8: Advanced Concepts & Best Practices

- API Security Basics (Helmet, CORS, rate-limiting)
- Authentication (JWT-based auth step-by-step)
- Hashing passwords with bcrypt
- Protecting private routes
- Role-based access control (Admin/User)
- Pagination and filtering
- Express error handling best practices
- Project structure for scalable apps
- Deploying Node.js App (Railway, Render, Vercel, or Heroku)
- Git & GitHub basics for deployment (.gitignore, package.json scripts)