# Node.js Learning Roadmap

Node.js Learning Roadmap

Phase 1: Prerequisites (If not strong already)

- JavaScript fundamentals (ES6+)
- Variables, functions, arrays, objects
- Promises, async/await
- Arrow functions, destructuring
- Basic HTML/CSS (for testing full apps)

Practice Projects:

- JS DOM calculator, Weather app using API

Phase 2: Node.js Core

- What is Node.js (event loop, single-threaded nature)
- npm (install, init, use packages)
- Modules (CommonJS / ES Modules)
- File system, path, and process modules

Practice Projects:

- CLI tool that reads and summarizes a .txt file
- JSON-based todo app in terminal

Phase 3: Express.js Framework

- Set up your first Express app
- Handle routes (GET, POST, etc.)
- Handle request/response, query params, JSON
- Serve static files
- Middleware basics (auth, logging)

Practice Projects:

- Build a REST API for a Notes app

Phase 4: Real-World Features

- Connect to MongoDB (use Mongoose)

- Add CRUD operations (Create, Read, Update, Delete)
- User Authentication (JWT or sessions)
- Input validation (Joi, express-validator)
- Environment variables with .env

Practice Projects:

- Full-stack app backend: Task manager / Blog API

Phase 5: Full-stack & Production Basics

- Connect Node.js backend with a frontend (React or HTML)
- Deploy backend on:
- Render / Railway (easy)
- or Vercel + external DB
- Add logging, error handling
- Secure your API: rate-limiting, helmet, CORS

Practice Projects:

- Full-stack blog or expense tracker (deployed)

Optional (Advanced but valuable):

- Learn WebSockets (real-time apps)
- Add email notifications (Nodemailer)
- Build a dashboard (e.g., analytics API)
- Learn unit testing with Jest or Supertest

Tools You'll Use:

- Postman / Thunder Client: Test APIs
- MongoDB Atlas: Free online DB
- Render / Railway: Easy deployment
- VS Code: Editor
- GitHub: Version control and portfolio

Final Tips:

- Build small projects every step of the way.
- Keep a GitHub profile updated with your code.

- Try contributing to open-source or cloning a real-world API.
- Once you're comfortable, learn TypeScript for even more job value.