That’s a great goal! Below is a structured **Node.js learning roadmap** that takes you from **absolute beginner to advanced level**, covering fundamentals, backend development, real-world projects, and best practices.

**🚀 Node.js Course Outline: Zero to Hero**

**📌 Module 1: JavaScript & Programming Fundamentals (Prerequisite)**

* Basic JavaScript Syntax (Variables, Functions, Loops)
* ES6+ Features (Arrow Functions, Destructuring, Async/Await, Modules)
* Working with Arrays & Objects
* Callbacks, Promises, and Async Programming
* Basic Error Handling

**📌 Module 2: Introduction to Node.js**

* What is Node.js? (Event Loop, Non-blocking I/O)
* Installing Node.js & NPM (Node Package Manager)
* Running Node.js Scripts (node command)
* REPL (Read-Eval-Print Loop) Basics
* Global Objects (process, global, Buffer)
* CommonJS vs ES Modules (require vs import)

**📌 Module 3: Core Node.js Modules**

* File System (fs) – Reading/Writing Files
* HTTP Module – Creating a Basic Server
* Path Module – Handling File Paths
* Events Module – Event Emitters & Listeners
* Streams (Readable, Writable, Transform)
* Utilities (util, os, url)

**📌 Module 4: Working with NPM & External Packages**

* Understanding package.json & package-lock.json
* Installing, Updating & Removing Packages
* Creating Your Own NPM Package
* Semantic Versioning (^, ~)
* Popular Packages (lodash, axios, moment, dotenv)

**📌 Module 5: Asynchronous Programming in Node.js**

* Callbacks & Callback Hell
* Promises (then/catch vs async/await)
* Error Handling in Async Code
* Promise.all, Promise.race, Promise.any
* Event Loop Deep Dive (Phases, nextTick, setImmediate)

**📌 Module 6: Building REST APIs with Express.js**

* Introduction to Express.js (Middleware, Routing)
* Creating a Basic REST API (CRUD Operations)
* Handling Requests (Query, Params, Body)
* Middleware (app.use, Custom Middleware)
* Error Handling Middleware
* Best Practices (Folder Structure, MVC Pattern)
* Rate Limiting, CORS, Security Headers

**📌 Module 7: Data Persistence (Databases)**

* **SQL Databases** (PostgreSQL, MySQL)
  + Connecting with pg or mysql2
  + Writing Queries (Basic CRUD)
  + ORMs (Sequelize, TypeORM)
* **NoSQL Databases** (MongoDB)
  + Connecting with mongoose
  + Defining Schemas & Models
  + CRUD Operations
* **Caching** (Redis for Session & Cache Management)

**📌 Module 8: Authentication & Authorization**

* Sessions vs Tokens (JWT)
* Password Hashing (bcrypt)
* OAuth (Google, GitHub Login)
* Role-Based Access Control (RBAC)
* Securing APIs (Helmet, CSRF, CORS)

**📌 Module 9: Testing & Debugging**

* Unit Testing with Jest or Mocha
* Integration Testing with Supertest
* Mocking (Sinon, Nock)
* Debugging with node-inspect, ndb

**📌 Module 10: Performance & Optimization**

* Clustering & Child Processes
* Load Balancing
* Profiling Node.js Apps (CPU, Memory)
* Caching Strategies (Redis, CDN)

**📌 Module 11: Real-time Applications (WebSockets)**

* Introduction to WebSockets (ws library)
* Building a Chat App with Socket.io
* Broadcasting & Rooms

**📌 Module 12: Microservices & Advanced Architectures**

* Introduction to Microservices
* Communication (REST, gRPC, Message Queues)
* Docker & Containerization Basics
* Deploying Node.js Apps (PM2, Nginx, AWS, Heroku)

**📌 Module 13: Final Projects (Apply Your Knowledge)**

1. **Todo App** (Express + MongoDB)
2. **E-commerce API** (JWT Auth, Stripe Payments)
3. **Chat Application** (Socket.io)
4. **Blog Platform** (Next.js + Node.js)
5. **URL Shortener** (Redis Caching)

**📌 Bonus: DevOps & CI/CD**

* Dockerizing Node.js Apps
* CI/CD with GitHub Actions
* Monitoring & Logging (Winston, ELK Stack)

**🎯 Learning Resources**

* **Official Docs**: [Node.js Docs](https://nodejs.org/en/docs/), [Express.js Docs](https://expressjs.com/)
* **Books**: "Node.js Design Patterns" by Mario Casciaro
* **YouTube**: Traversy Media, The Net Ninja, Academind
* **Courses**:
  + [The Complete Node.js Developer Course (Udemy)](https://www.udemy.com/course/the-complete-nodejs-developer-course-2/)
  + [Node.js Microservices (Pluralsight)](https://www.pluralsight.com/)

**🔥 Tips for Success**

✔ **Code Daily** – Practice small projects  
✔ **Read Documentation** – Get comfortable with official docs  
✔ **Join Communities** – Stack Overflow, Discord, Reddit  
✔ **Build a Portfolio** – Showcase your projects

Would you like recommendations for hands-on exercises or project ideas at each stage? 🚀