### **Week 1: Basics of Node.js & JavaScript Fundamentals**

#### **Day 58: Introduction to Node.js**

* What is Node.js and why use it?
* Installing Node.js and setting up a development environment
* Understanding the Node.js runtime
* Hello World in Node.js

#### **Day 59: JavaScript Fundamentals Review**

* Asynchronous programming in JavaScript
* Promises and callbacks
* Async/Await

#### **Day 60: Node.js Modules**

* Core modules in Node.js (fs, http, path, os, etc.)
* Creating custom modules
* Importing and exporting modules using require and module.exports

#### **Day 61: npm (Node Package Manager)**

* What is npm and how to use it?
* Installing and managing packages
* Understanding package.json and package-lock.json

#### **Day 62: Working with File System (fs)**

* Reading files with fs.readFile and fs.createReadStream
* Writing to files with fs.writeFile and fs.createWriteStream
* Understanding synchronous vs. asynchronous file operations

#### **Day 63: Introduction to Express.js**

* Installing Express.js
* Setting up a basic Express.js server
* Routes and HTTP methods (GET, POST, PUT, DELETE)
* Middleware in Express.js

#### **Day 64: Express.js Basics**

* Understanding the request and response objects
* Query parameters and URL parameters
* Sending JSON responses
* Handling form data (req.body)

### **Week 2: Intermediate Node.js Concepts**

#### **Day 65: Routing in Express.js**

* Setting up routes with Express
* Route parameters and handling dynamic routes
* Router objects and modular routing

#### **Day 66: Error Handling & Debugging**

* Handling errors in Node.js
* Understanding try-catch and middleware for error handling
* Debugging Node.js applications using console and debuggers

#### **Day 67: Working with Databases (MongoDB + Mongoose)**

* Introduction to NoSQL databases
* Installing MongoDB locally or using cloud services (MongoDB Atlas)
* Connecting Node.js with MongoDB using Mongoose
* Basic CRUD operations (Create, Read, Update, Delete)

#### **Day 68: Authentication and Authorization**

* Introduction to authentication (JWT, OAuth)
* Building simple authentication system (JWT)
* Securing routes with middleware

#### **Day 69: Sessions and Cookies**

* What are sessions and cookies?
* Working with sessions in Express.js
* Storing and managing cookies

#### **Day 69: File Uploads in Node.js**

* Handling file uploads with Express
* Using multer or other libraries for file uploads
* Storing uploaded files

#### **Day 70: Review & Practice**

* Implement a small project using the concepts learned so far (simple API or web app)

### **Week 3: Advanced Node.js Concepts**

#### **Day 71: RESTful API Design**

* Introduction to RESTful APIs
* Structuring APIs and following REST principles
* HTTP status codes and best practices

#### **Day 72: Working with WebSockets (Real-Time Communication)**

* Introduction to WebSockets
* Setting up WebSocket server with Socket.io
* Real-time communication example (chat app)

#### **Day 73: Asynchronous Programming in Node.js**

* Event Loop and non-blocking I/O
* Understanding callbacks, promises, and async/await
* Using async libraries (like async.js)

#### **Day 74: Testing in Node.js**

* Introduction to testing frameworks (Mocha, Chai, Jest)
* Unit testing API routes and functions
* Writing and running tests in Node.js

#### **Day 75: Deployment with Docker**

* Introduction to Docker
* Creating Docker container for Node.js apps
* Running and deploying your app using Docker

#### **Day 76: Integrating External APIs**

* Working with third-party APIs (REST APIs)
* Fetching data using axios or native http module
* Handling API responses and errors

#### **Day 78: Review & Practice**

* Build a medium-level project (e.g., a CRUD API or simple chat app)

### **Week 4: Full-Stack Node.js & Final Project**

#### **Day 79: Introduction to Full-Stack Development**

* Introduction to full-stack concepts (Frontend + Backend)
* Using Node.js with frontend technologies (React, Angular, etc.)
* Creating a simple full-stack app (Backend API with Node.js and frontend UI)

#### **Day 80: Advanced Authentication & Security**

* OAuth2, JWT, and Passport.js for complex authentication
* Securing API endpoints and preventing common attacks (CSRF, XSS)
* Rate limiting and IP blocking

#### **Day 81: Advanced Performance Optimization**

* Profiling and optimizing Node.js applications
* Using clustering for better performance
* Understanding load balancing and scaling Node.js apps

#### **Day 82: Caching in Node.js**

* Introduction to caching and why it’s important
* Caching strategies (in-memory, Redis)
* Implementing caching in Node.js

#### **Day 83: Server-Side Rendering (SSR) with Node.js**

* Introduction to SSR and its use cases
* Using Node.js with frameworks like Next.js for SSR

#### **Day 84: Final Project Setup**

* Choose a final project (Full-stack app, RESTful API, etc.)
* Set up project structure and start coding

#### **Day 85: Continue Final Project**

* Work on the project, integrating all learned concepts (CRUD operations, authentication, WebSockets, etc.)

#### **Day 86: Testing and Debugging Final Project**

* Write tests for your project
* Debug and optimize your project

#### **Day 87: Deploying Final Project**

* Deploying to a cloud platform (Heroku, AWS, DigitalOcean, etc.)
* Setting up environment variables and production configurations