Mastering Nest JS

Duration: 5 days
Prerequisites: Knowledge of API, JavaScript and Node.js

Day 1: Introduction to NestJS and Project Setup

Topics:

- What is NestJS? Why NestJS?
- Overview of Node.js, TypeScript, and Express.js
- NestJS Architecture: Modules, Controllers, Services
- Installing Nest CLI and Creating Your First Project
- Project Structure Deep Dive
- Creating First Controller and Service

Labs:

- Install NestJS CLI and create a new app
- Create a basic controller with a GET route
- Add a service to return dynamic data
- Run and test the application

Day 2: Routing, Modules, and Services

Topics:

- Creating and using Modules
- Creating multiple Controllers and Services
- Handling route parameters and query params
- Working with DTOs and validation
- Dependency Injection in NestJS

Labs:

- Create multiple routes (GET, POST, PUT, DELETE)
- Create custom DTOs for validation using class-validator
- Refactor logic into service and inject into controller
- Modularize features using submodules

Day 3: Database Integration with PostgreSQL and TypeORM

Topics:

- Introduction to TypeORM and Relational Databases
- PostgreSQL Installation and Setup
- Connecting NestJS with PostgreSQL using TypeORM
- Creating Entities and Repositories
- Performing CRUD operations
- Error Handling and Validation Pipes

Labs:

- Connect NestJS app to PostgreSQL
- Create and migrate an entity (e.g., Task/User)
- Build complete CRUD using TypeORM Repository
- Validate input data and handle exceptions

Day 4: Authentication, Middleware, and Security

Topics:

- What are Middleware, Pipes, and Guards?
- Creating custom middleware for logging
- Exception filters and global error handling
- JWT-based Authentication (Login + Protect Routes)
- Creating Auth Module
- Hashing passwords with bcrypt

Labs:

- Create login and signup routes
- Hash passwords and generate JWT tokens
- Protect routes using JWT Guards
- Implement custom middleware and global exception filters

Day 5: Final Project, Testing & Deployment

Topics:

- Overview of Final Mini Project
- Application Structure and Feature Planning
- Writing Unit Tests with Jest (Basics)
- Environment Configuration using .env
- Deployment Options:
    - Deploying to **Vercel** (for API)
    - Deploying to **Heroku**
    - Using PM2 for production Node apps

Labs:

- Build and test a mini-project:
    - Authenticated Task Manager with CRUD
- Write basic unit tests for service and controller
- Create .env for config and sensitive values
- Deploy the app to Render/Heroku using GitHub repo
- Test deployed endpoints using Postman