# 12-Month .NET Core + Angular Full-Stack Developer Roadmap to 50 LPA

## Phase 1 (Months 1–3): Core Foundations

### .NET Core / C#

* Value types vs reference types, Span<T>, Memory<T>
* Async/Await, Task Parallelism, ValueTask
* Delegates, Func/Action, Events
* Records vs Classes, Immutability
* Reflection & Dynamic Programming
* Middleware pipeline & custom middleware
* Dependency Injection (transient, scoped, singleton)
* Filters (Auth, Action, Exception)
* Configuration providers & Options pattern
* API versioning & routing
* LINQ performance, AsNoTracking
* Eager vs lazy loading
* Migrations, Seeding
* Raw SQL queries
* Transactions & Concurrency

### Angular

* Component lifecycle hooks
* Reactive forms vs template-driven
* RxJS operators (mergeMap, switchMap, concatMap, forkJoin)
* Angular routing (lazy loading, guards, resolvers)
* Change detection strategy (Default vs OnPush)
* State management (NgRx, Akita)
* Angular Signals (new reactivity model)
* Unit testing with Jasmine/Karma

### System Design Basics

* Load balancers, Reverse proxies
* Horizontal vs Vertical scaling
* Database partitioning & sharding
* Caching layers (in-memory, Redis)
* Basic design patterns (Factory, Observer, Mediator, CQRS)

## Phase 2 (Months 4–6): Cloud & DevOps

### Azure Cloud

* App Services, Functions, Blob Storage
* Azure SQL, Cosmos DB
* Service Bus vs Event Grid
* Application Insights, Log Analytics
* VNETs, private endpoints
* API Management (throttling, caching, rate limiting)
* Key Vault (secret rotation, managed identities)

### Containers & Orchestration

* Docker basics (images, layers, volumes, networking)
* Docker Compose (multi-container apps)
* Kubernetes: Pods, Deployments, Services, Ingress
* ConfigMaps, Secrets, Persistent Volumes
* AKS (Azure Kubernetes Service)

### CI/CD

* Azure DevOps Pipelines (YAML)
* GitHub Actions (jobs, runners, secrets)
* Branching strategies (GitFlow, trunk-based)
* Infrastructure as Code (Terraform/Bicep)
* Blue-Green & Canary deployments

## Phase 3 (Months 7–9): Architecture & Scalability

### Microservices Architecture

* Domain-Driven Design (Entities, Value Objects, Aggregates)
* Bounded Contexts
* Service communication: REST, gRPC, messaging
* API Gateway patterns (Ocelot, YARP, Azure API Management)
* Circuit breaker, Retry, Timeout (Polly in .NET)

### Event-Driven Systems

* Kafka vs RabbitMQ vs Azure Service Bus
* Pub/Sub vs Point-to-Point messaging
* Event Sourcing concepts
* Idempotency & message deduplication
* Saga pattern (distributed transactions)

### Security

* OAuth 2.0, OpenID Connect
* JWT tokens (access vs refresh)
* Claims-based authentication
* IdentityServer / Azure AD
* OWASP Top 10
* API Gateway rate limiting & throttling

## Phase 4 (Months 10–12): Emerging Tech & Leadership

### AI & Integration

* Azure OpenAI (text embeddings, completions, chat)
* ML.NET basics (training + serving models)
* Retrieval-Augmented Generation (RAG)
* Integrating AI in .NET APIs & Angular frontends

### Data & Analytics

* SQL tuning: indexes, execution plans, partitioning
* Redis caching strategies
* NoSQL design: Cosmos DB, MongoDB
* Data streaming pipelines (Kafka → Spark → Data Lake)
* Event-driven analytics dashboards

### Leadership & Practices

* TDD & BDD: xUnit, NUnit, Moq
* Static code analysis: SonarQube, StyleCop
* Clean Architecture in .NET (Domain, Application, Infrastructure, API layers)
* Agile ceremonies & sprint leadership
* Architecture documentation (C4 model diagrams)
* Mentorship & team guidance