Self-introduction for .NET Lead

Hi, I’m Rupesh Singh. I have over 11 years of experience as a .NET Full Stack Developer, and in the last few years, I’ve also worked in **team lead roles**, driving delivery for enterprise web applications.

My core skills are in **.NET Core, C#, Angular, and AWS**. I’ve led small to mid-sized teams — usually around 4 to 6 developers — where I’ve handled everything from **code reviews, mentoring**, and **technical solutioning**, to helping teams stay aligned with **project timelines and quality standards**.

On the backend, I work with .NET Core Web APIs, LINQ, Entity Framework, and SOLID design principles. On the frontend, I build with **Angular (v12+)**, using RxJS, reactive forms, and modular architecture for performance.

Recently, I led the development of a **cloud-based e-commerce microservices platform** hosted on **AWS**, where we used:

* Lambda, SQS/SNS, API Gateway, DynamoDB/PostgreSQL
* CloudWatch for monitoring and Cognito for authentication
* CI/CD using GitHub Actions and Dockerized deployments to EKS

I’ve also worked with **CQRS**, **Mediator pattern**, and **Domain-Driven Design**, and I’m comfortable with writing unit and integration tests using xUnit, Moq, and Jasmine/Karma for Angular.

Apart from coding, I’m very focused on **team collaboration**, **agile delivery**, and aligning development practices with the business. I believe in taking ownership, improving processes, and helping my team continuously grow.

Happy to share more about any specific project or technical detail if you'd like.

Recent project

**In my previous role, I worked on a migration project where we converted a legacy monolithic publishing tool built on .NET Framework into a modern microservices-based solution using .NET Core, Angular, and AWS.”**

**Situation:**  
The legacy system had performance issues, tight coupling, and was hard to scale or maintain. It was a desktop-based application with limited deployment flexibility.

**Task:**  
My responsibility as a Team Lead was to design the architecture, split modules into services, and implement the new stack using best practices.

**Action:**

* We redesigned the solution using **.NET Core Web APIs** as microservices.
* Each module (like content editor, workflow engine, and publishing scheduler) became a **separate API**.
* On the frontend, we built an Angular 14+ application for rich UI and faster delivery.
* For deployment and scalability, we used **AWS services** like EC2 for hosting, S3 for static content, and RDS for data storage.
* We also implemented **JWT-based authentication**, centralized logging, and CI/CD using GitHub + CodeDeploy.

**Result:**  
The system became **modular, faster, and easier to maintain**. Deployment time reduced by 70%, and onboarding new developers became smoother.