ScholarHat

Swipe

.NET 8

DEVELOPER ROADMAP

2024 EDITION

LEVEL UP YOUR CAREER

.NET

SHAILENDRA CHAUHAN
Microsoft MVP, Founder & CEO - ScholarHat

1 .NET Fundamentals

- **Basics of C#** including syntax, data types, control flow, and object-oriented programming concepts.
- **dotnet CLI** including commands and VS code way to create and build .NET Core based applications.
- **Master IDEs** like Visual Studio 2022 or Visual Code.
- Understand **Nuget Package Management** System.
- **Master C# Collections** (like IEnumerable, List, Dictionary etc.) and **Asynchronous Programming.**
- Understand **.NET versions and libraries** including .NET Framework, .NET Core, Mono and .NET Standard
- Focus on **.NET 6** or above and **C# 10** or above features. Ideally, **.NET 8** and **C# 12**.

2 Patterns and Practices

- Practice **SOLID, YAGNI, DRY and KISS Principles**
- **Master Design Patterns** including Repository, Dependency Injection and **Gang-of-four patterns** like: Singleton, Factory, Builder, Adapter, Façade, Decorator, Chain of Responsibility Pattern, Iterator Pattern, State, Strategy.
- Understand **Software Architectures** like SOA, REST, N-Tier/N-Layer, Clean Architecture and Microservices.
- Understand **Architectural Patterns** like MVC, MVVM and Domain Driven Design.
- Practice clean code and code refactoring
- Master **Version Control** like TFS, Git - GitHub/GitLab



3 ASP.NET Core 8

- **Master ASP.NET Core MVC Fundamentals** including MVC pattern, Razor syntax, Routing, Helpers, Forms, Validations and Data Passing Techniques.
- Understand ASP.NET Core **Request Pipeline.**
- Use Cases for **Middleware and Filters.**
- Understand **Configurations** and **appsettings.json**
- **Dependency Injection** using built-In Service Container methods **(Scoped, Transient & Singleton).**
- **Authentication & Authorization** including ASP.NET Identity, cookie based authentication, token-based authentication etc.
- **Exception Handling** and **Error Logging**
- Building REST API using **WebAPI Core**

4 Data Access Technologies

- **Master LINQ Queries** to query various data sources like SQL/NoSQL data, C# collections and APIs.
- Master **Entity Framework Core**: **Code First, Database First** Approaches.
- Configure **EF Core power tool in** Visual Studio to follow Database First Approach efficiently.
- Learn to **optimize LINQ queries** by understanding LINQ query execution methods and data loading styles.
- Understand EF Core **change tracker API** and Entities relationships.
- Know **micro ORM** like **Dapper** to execute RAW SQL queries efficiently.
- Know traditional **ADO.NET** to perform database operations in ASP.NET Core.



5 Databases: SQL/NoSQL

- Basics of **Database design** (normal forms, SQL keys, table relationships)
- Learn **SQL database fundamentals** including SQL Commands, SQL Queries, SQL Queries Execution Order, Joins and SQL Predicates.
- Master SQL **Stored Procedures**, SQL **Functions, Indexing** and **Triggers, Cursors** too! (you might not use this regularly).
- Understand **SQL vs NoSQL Databases** use cases.
- Learn about Database Optimizations.
- Learn to work with **SQL Databases** like SQL Server, PostgreSQL, Azure SQL, and **NoSQL databases** like MongoDB, CosmosDB etc.

6 Testing

- Understand Test Driven Development (**TDD**) and Behavior Driven Development (**BDD**)
- Master Unit Test frameworks like MSTest, xUnit for writings test cases.
- Know Test runners and test explorers
- Know Asserts and test attributes
- Understand **Mocking libraries** (Moq, TypeMock etc.)
- Generate **dummy data for test cases** using Bogus or FakeItEasy or AutoFixture libaries.
- K6 or JMeter for **load testing**.
- Know SpecFlow for **Behavior Testing**.



7 Frontend Frameworks/Libs

- Understanding of **HTML, CSS, and JavaScript**.
- Master frontend like **Angular or React or Blazor** for building interactive web UIs.
- Understand **component based architecture and practices** to build UI using Angular or React or Blazor.
- Learn **Blazor to build UI using C#** only and serve client and server side both.
- Learn **React** to build UI using JavaScript/TypeScript and a flexible way to build UI using various libraries.
- Learn **Angular** to build UI using TypeScript and build end-to-end using Angular itself.

8 Cloud and DevOps

- **Know Cloud** to quickly provision, configure, and deploy .NET based applications.
- The most popular clouds are **Azure, AWS and GCP**.
- With **.NET based applications**, **first choice is Azure**. Learn I**aaS services** (VM, Storage, Networking) and **PaaS services** (WebApp, Functions, Azure SQL, Cosmos DB) and management using Azure Entra ID.
- Master to **build and secure** .NET-based application with **various Azure services**.
- Learn to **automate code deployment** to cloud using CI/CD with the help of Azure DevOps or GitHub Actions.
- Get yourself **Microsoft Certified Azure Developer** or **Microsoft Certified Azure Solution Architect**.

9 Best Practices & Libraries

- Master various **Cache practices** including in-memory cache, distributed cache, database cache and CDN.
- Know to configure **environment (Dev, UAT, Staging and Prod)** based settings
- Handle **errors globally** and log errors using logging libraries like Serilog, NLog etc.
- Run **background tasks** using Hangfire or Quartz.NET.
- Know **Polly** library for implementing Circuit Breaker and Retry Mechanism.
- For **Code Quality and Security** use Sonar Analyzer.
- Use Open API/Swashbuckle for **API Documentation**.
- Know **Benchmark.NET** for evaluating your application performance

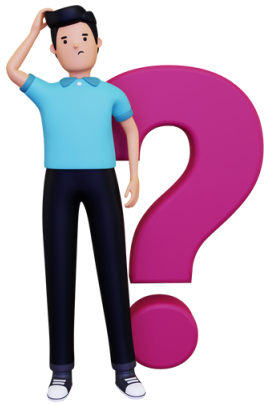
10 Soft Skills

- **Read technical books or articles** to expressing thoughts and ideas concisely.
- Learn to **prioritize and manage tasks** efficiently.
- Utilize platforms like **GitHub** for showcasing your technical projects and **LinkedIn** for networking..
- Practice real-world scenarios, or study analytical techniques to **develop a logical and systematic approach** to problem-solving.
- **Volunteer for leadership roles** or lead small projects to build leadership skills.
- **Stay updated with industry trends** and advancements to remain relevant and valuable.
- Practice to use **Chat GPT** or **Github Copilot** like tools to improve your productivity and save your time.





How to follow this roadmap?



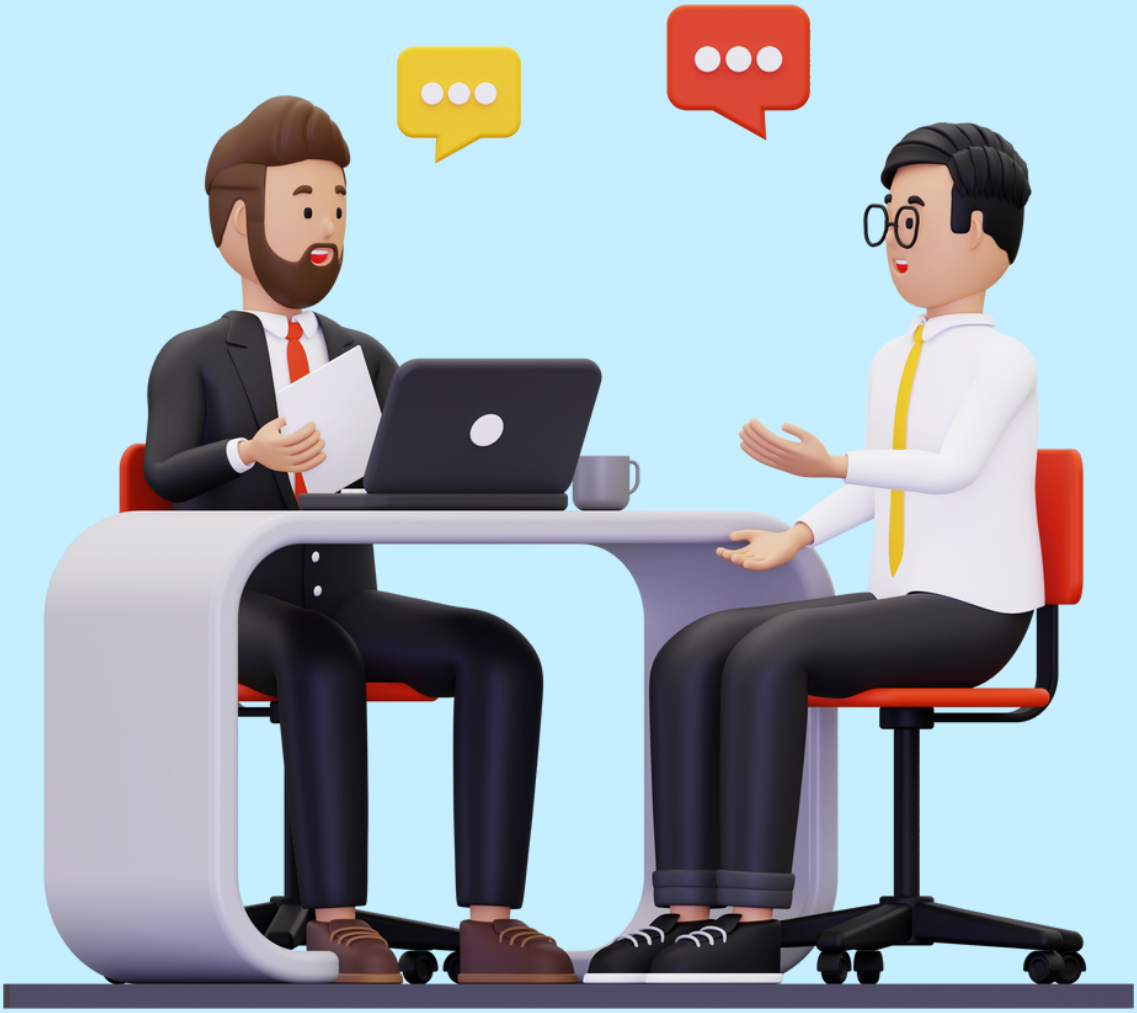
At ScholarHat, we believe **mastering a technology** is a **three-step process** as mentioned below:

- **Step1 - Learn Skills:** You can learn .NET skills by using **Microsoft official docs** on .NET, C#, ASP.NET Core, EF Core etc. or **through Videos** on YouTube or **Videos based courses**. For topic revision and recalling make **short notes**.
- **Step2 - Build Experience:** You can build hands-on experience by creating **coding workflow** like Login workflow, cart workflow, payment gateway workflow, Security workflow. Further **build end-to-end real world applications** like Dominoz, Flipkart, OTT platform etc.
- **Step3 - Empower Yourself:** Build your **strong profile** by mentioning all the above skills with **hands-on experience** on projects. Prepare yourself with interview **Q&A about .NET** to crack your next job interview.

Swipe

Congrats!

You are just one interview away!



WAS THIS HELPFUL?

Share with your friend who needs it!

Love. Like. Comment. Share.

Learn. Build. Empower.