Optimizing .NET Application Development

Target Audience

This course is designed for:

- Software Developers and Engineers
- Technical Leads and Architects
- IT Professionals

Course Description

Optimizing .NET Application Development is an intensive, hands-on course designed to equip developers with the skills and knowledge needed to build high-performance, scalable, and maintainable .NET applications. Over the span of 10 days, participants will delve into advanced .NET topics, including design patterns, SOLID principles, microservices, and performance optimization techniques.

.NET Design Patterns and SOLID Principles (5days)

Module 1: Introduction to OO Design Thinking

- Design Patterns makes use of OOPs concepts - Composition, Aggregation, Inheritance, Encapsulation
- Interface-vs-Implementation
- Dependency Inversion
- SOLID principles implementation

Module 2: Introduction of Creational Pattern

Implementing the list of Creational Patterns:

- Understanding & Implementing the Factory Design Pattern
- Advantages of using Abstract Factory Design Pattern
- Overview of Builder Design Pattern & how it is different from Abstract factory Design Pattern
- Cloning of an object using Prototype Design Pattern
- Understanding & Implementing the Singleton
- Demo: Implementing the Factory, Abstract Factory, Builder, Prototype & Singleton Design Pattern

Module 3: Introduction of Structural Pattern

Implementing the list of Structural Patterns:

- Overview of Adapter Design Pattern & Understand when to use it
- Introduction of Bridge Design Pattern
- Compose objects using Composite Design Pattern
- Add additional functionalities using Decorator Design Pattern
- Create object without constructor call using Façade Design Pattern
- Implementing Flyweight Design Pattern
- Understanding Proxy Design Pattern
- Demo: Implementing the Adapter, Composite, Decorator, Faced, Fly weight & Proxy Pattern

Module 4: Introduction of Behavioural Pattern
Implementing the list of Behavioural Pattern

- Overview of Chain of Responsibility Design Pattern
- Understanding the Command Design Pattern
- Usage of Interpreter Design Pattern
- Introduction of Iterator Design Pattern
- Allow objects to communicate using Mediator Design Pattern
- Overview of Memento Design Pattern
- Understanding the Observer Design Pattern
- Alter the behaviour of an object using State Design Pattern
- Overview of Strategy Design Pattern
- Overview of Visitor Design Pattern
- Implementing the Template Method Design Pattern
- Demo: Chain of Responsibility, Mediator, Observer, Template Method

Module 5: Asp.net core, DI and Repository pattern
Implementing Repository pattern

- Understanding Request pipeline in asp.net core
- Dependency Injection in asp.net core
- Layering a project into infrastructure, business and presentation

ADO.NET/EF/Dapper and LINQ (2 days)

Module 1: Ado.net, LINQ

- Extension methods and LINQ
- Understanding ado.net architecture
- IEnumerable vs IQueryable
- ORM and common design patterns

Module 2: Entity Framework and Dapper

- EF architecture
- Design patterns implemented in EF
- Dapper architecture
- Implementing Generic Repository pattern using EF and Dapper

Best practices of Coding (1 day)

Module 1: Microservices and Micro frontend

- Understanding Microservices architecture
- Understanding Domain Driven Design
- Decomposing Monoliths to Microservices
- Creating a catalog microservice
- Creating a basket microservice
- Understanding Micro front ends
- Building a micro front end

Docker and Containerization (1day)

- Understanding Containerization
- installing docker
- Common docker commands

- running multiple containers
- containerizing a asp.net core project
- running a full stack microservices project in container

Performance Improvement Techniques: Multithreading and Parallel programming in .NET / Memory Management/DB Query (1day)

Module 1: Multithreading and TPL

- Multithreaded Programming in C#
- Async and Await concepts
- Async programming in a MVC controller
- Improving performance using Async actions
- Task Parallel Library
- Locking, shared variables and best practices
- Singleton pattern and thread safety
- Parallel LINQ

Module 2: Garbage Collection

- Introducing Garbage Collection
- Value types and reference types
- Anonymous types,tuples,nullable reference types
- Boxing /UnBoxing
- Using IDisposable pattern

Module 2: DB query optimization

- Databases and Indexing
- Using the SQL Profiler Tools
- Understand LINQ Query Execution Location