**C# ASP.CORE 8 MICROSERVICES**

**Total Duration: 3 Days**

**Day 1**

**Identifying Business Case and Segmentation**

**C#.Net Advanced**

1.Generics (Hands on for all topics relevant to identified use case except why Generics)

* Generic Types
* Why Generics Exist
* Generic Methods
* Declaring Type Parameters
* typeof and Unbound Generic Types
* The default Generic Value
* Generic Constraints
* Subclassing Generic Types
* Self-Referencing Generic Declarations
* Static Data
* Type Parameters and Conversions
* Covariance
* Contravariance
* Tuples

2.Delegates (Hands on for all topics relevant to identified use case)

* Writing Plug-In Methods with Delegates
* Instance and Static Method Targets
* Multicast Delegates
* Generic Delegate Types
* The Func and Action Delegates
* Delegates Versus Interfaces
* Delegate Compatibility

3. Events (Hands on for all topics relevant to identified use case publisher and subscriber with Kafka)

* Standard Event Pattern
* Event Accessors
* Event Modifiers

Day 2

4. Lambda Expressions (Hands on for all topics relevant to identified use case CRUD with SQL Server)

* Explicitly Specifying Lambda Parameter and Return Types
* Capturing Outer Variables
* Lambda Expressions Versus Local Methods
* Anonymous Methods
* Extension Method

**LINQ Queries** (Hands on for all topics relevant to identified use case CRUD with SQL Server)

* Fluent Syntax
  + Chaining Query Operators
  + Composing Lambda Expressions
  + Natural Ordering
  + Other Operators
* 2. Query Expressions
  + Range Variables
  + Query Syntax Versus SQL Syntax
  + Query Syntax Versus Fluent Syntax
  + Mixed-Syntax Queries
* 3. Deferred Execution
  + Reevaluation
  + Captured Variables
  + How Deferred Execution Works
  + Chaining Decorators
  + How Queries Are Executed
* 4. Subqueries
  + Subqueries and Deferred Execution
* 5. Composition Strategies
  + Progressive Query Building
  + The into Keyword
  + Wrapping Queries
* 6. Projection Strategies
  + Object Initializers
  + Anonymous Types
  + The let Keyword
* 7. Interpreted Queries
  + How Interpreted Queries Work
  + Combining Interpreted and Local Queries
  + AsEnumerable

**Day 3**

**Concurrency and Asynchrony** (Hands on for all topics relevant to identified use case Thread, Multithread, Locks, Sleep and Join, Thread Pool, Synchronization, Producer and Consumer with RabbitMQ, Thread Pool and Tasks)

* Introduction
* Threading
  + Creating a Thread
  + Join and Sleep
  + Blocking
  + Local Versus Shared State
  + Locking and Thread Safety
  + Passing Data to a Thread
  + Exception Handling
  + Foreground Versus Background Threads
  + Thread Priority
  + Signaling
  + Threading in Rich Client Applications
  + Synchronization Contexts
  + The Thread Pool
  + Atomic Methods Thread Safety and Race Conditions in C#
* Tasks
  + Starting a Task
  + Returning values
  + Exceptions
  + Continuations
  + TaskCompletionSource
  + Task.Delay
  + Principles of Asynchrony
  + Synchronous Versus Asynchronous Operations
  + What Is Asynchronous Programming?
  + Asynchronous Programming and Continuations
  + Why Language Support Is Important
* Asynchronous Functions in C#
  + Awaiting
  + Writing Asynchronous Functions
  + Asynchronous Lambda Expressions
  + Asynchronous Streams
  + Asynchronous Methods in WinRT
  + Asynchrony and Synchronization Contexts
  + Optimizations

**Parallel Programming** (Hands on for all topics relevant to identified use case with Large Files)

* PLINQ
  + Parallel Execution Ballistics
  + PLINQ and Ordering
  + LINQ Limitations
  + Example: Parallel Spellchecker
  + Functional Purity
  + Setting the Degree of Parallelism
  + Cancellation
  + Optimizing PLINQ
* The Parallel Class
  + Parallel.Invoke
  + Parallel.For and Parallel.ForEach
* Task Parallelism
  + Creating and Starting Tasks
  + Waiting on Multiple Tasks
  + Canceling Tasks
  + Continuations
  + Task Schedulers
  + TaskFactory