**📘 .NET Training – Bangalore (Boeing)**

**Prerequisites**

* Basic knowledge of programming in C#
* Familiarity with object-oriented concepts (classes, interfaces, inheritance)
* Awareness of web development basics (HTTP, REST)
* Exposure to Visual Studio / Visual Studio Code
* Familiarity with Git for version control

**Lab Requirements**

* Windows 10/11 laptop with admin rights
* Visual Studio 2022 Community/Enterprise OR Visual Studio Code with C# extension
* .NET SDK (latest LTS, e.g., .NET 8.0) installed
* Azure subscription (trial or enterprise)
* SQL Server Express / Azure Cosmos DB emulator
* Docker Desktop (for containerized deployments, optional)
* WIX Toolset installed for MSI packaging
* Postman for API testing

**Day-wise Schedule**

**Day 1 – .NET Core Fundamentals & Asynchronous Programming**

**Topics:**

* Overview of .NET Core features and improvements
* Project structure and SDK-style projects
* Dependency Injection (DI) in .NET Core
* Configuration and options pattern
* Asynchronous Programming: async/await, Task, Task<T>, ValueTask<T>

**Labs:**

* Create a simple .NET Core console app
* Implement DI for a service layer
* Use async/await with I/O-bound operations

**Outcome:**  
Participants will understand .NET Core architecture, project structure, DI, and async programming.

**Summary:**  
Day 1 sets the foundation of modern .NET development with strong focus on DI, configuration, and async/await patterns.

**Day 2 – Building APIs with ASP.NET Core**

**Topics:**

* RESTful API development with ASP.NET Core Web API
* Routing, controllers, and action methods
* Model binding and validation
* Middleware pipeline and custom middleware
* Delegates & Events in C#: delegates, multicast delegates, events

**Labs:**

* Build a CRUD API for a sample domain (e.g., Employee API)
* Implement model validation with Data Annotations
* Create custom middleware for logging requests
* Use delegates to implement custom event handling

**Outcome:**  
Ability to create and extend APIs using ASP.NET Core Web API with validation, middleware, and delegates/events.

**Summary:**  
Day 2 introduces hands-on API creation with ASP.NET Core and extends into event-driven programming using delegates/events.

**Day 3 – Azure Functions & Data Access (Cosmos DB + LINQ)**

**Topics:**

* Creating and deploying Azure Functions with .NET Core
* Triggers and bindings (HTTP, Timer, Queue)
* Dependency Injection in Functions
* Monitoring/logging with Application Insights
* Cosmos DB SDK for .NET, CRUD with LINQ
* LINQ queries, deferred vs immediate execution, LINQ to SQL/XML

**Labs:**

* Create an Azure Function with HTTP Trigger
* Add DI services to Azure Functions
* Connect Function to Cosmos DB and perform CRUD operations
* Write LINQ queries over collections and XML

**Outcome:**  
Participants gain skills in serverless app development and cloud-native data access using Cosmos DB and LINQ.

**Summary:**  
Day 3 connects APIs with Azure Functions and Cosmos DB, with LINQ as the unifying querying technique.

**Day 4 – Security, Testing & Debugging**

**Topics:**

* Authentication & Authorization in APIs (JWT, OAuth2)
* Securing Azure Functions
* Role-based access control (RBAC) with Azure AD
* Managing secrets with Azure Key Vault
* Unit testing with xUnit/NUnit
* Integration testing for APIs
* Mocking dependencies
* Debugging Azure Functions locally

**Labs:**

* Secure an API with JWT authentication
* Configure Azure AD RBAC for API access
* Use Azure Key Vault to manage secrets
* Write xUnit tests for services and controllers
* Debug Azure Functions locally with Visual Studio

**Outcome:**  
Ability to secure .NET applications and write automated tests for reliable deployments.

**Summary:**  
Day 4 builds expertise in securing and testing APIs with modern authentication, authorization, and debugging strategies.

**Day 5 – Packaging, Deployment, Optimization & Monitoring**

**Topics:**

* WIX Toolset for MSI installers
  + WIX project files (.wxs)
  + Components, features, installation logic
  + Custom actions & UI customization
  + Building/testing MSI installers
* Deployment strategies for Azure Functions & APIs
* Performance optimization: caching, async patterns, profiling
* Monitoring & diagnostics with Azure Monitor & Application Insights
* Logging best practices, alerts, dashboards

**Labs:**

* Package a sample .NET Core app using WIX Toolset (MSI installer)
* Deploy Azure Function to Azure portal
* Apply in-memory/distributed caching to improve performance
* Configure Application Insights to monitor API calls and set alerts

**Outcome:**  
Participants will know how to package .NET apps, deploy them securely, optimize performance, and monitor production workloads.

**Summary:**  
Day 5 closes the loop with deployment, packaging, performance tuning, and enterprise-grade monitoring for production-readiness.

**Overall Course Outcomes**

By the end of this course, participants will be able to:

* Build, secure, and deploy scalable APIs using ASP.NET Core
* Develop serverless applications with Azure Functions
* Work with Cosmos DB and LINQ for cloud-native data access
* Implement async programming, delegates, events, and dynamic features in .NET
* Package apps with WIX Toolset and deploy them to Azure
* Optimize performance and set up monitoring with Azure Monitor & Application Insights
* Apply industry best practices in testing, debugging, and security