

Enterprise .NET MAUI, Cloud & DevSecOps Engineering

Duration: 12 weeks (576 hours)

Week 1: Core .NET, C# Fundamentals & AI-Assisted Engineering

Modules & Topics

Module 1: .NET Development Environment

- .NET ecosystem and architecture
- .NET SDK, Runtime and CLI
- Visual Studio configuration
- Solution and project structure
- NuGet package management
- Build and configuration concepts
- Debug vs Release configuration

- .NET CLI commands

Module 2: C# Fundamentals & Advanced Language Constructs

- C# type system
- Value vs reference types
- Nullable reference types
- Properties and access modifiers
- Exception handling
- Pattern matching
- Records and immutable models
- Modern C# syntax and language features

Module 3: Object-Oriented Programming

- Encapsulation
- Abstraction
- Inheritance
- Polymorphism
- Interfaces
- Composition vs inheritance
- Interface segregation

Module 4: SOLID Principles & Clean Code

- Single Responsibility Principle
- Open/Closed Principle
- Liskov Substitution Principle
- Interface Segregation Principle
- Dependency Inversion Principle
- Clean code practices
- Separation of concerns

Module 5: Design Patterns & Dependency Injection

- Repository Pattern
- Factory Pattern
- Singleton Pattern
- Dependency Injection
- Inversion of Control
- Transient, Scoped and Singleton lifetimes

Module 6: GitHub Copilot for .NET Developers

- Copilot setup in Visual Studio
- Effective prompt engineering
- Code generation
- Refactoring
- Documentation generation
- Unit-test suggestions
- Reviewing AI-generated code
- Secure and responsible AI-assisted development

Hands-on Labs

- Configure .NET development environment.
- Build/refactor an enterprise C# console application.
- Apply SOLID principles.
- Implement Factory and Repository patterns.
- Configure dependency injection.
- Generate and review code using GitHub Copilot.

Incremental Project

Create the solution structure, domain models, interfaces, services, DI configuration and Git repository for the enterprise case-study application.

Key Deliverables

- SOLID-compliant C# application
- DI-enabled architecture
- Initial Git repository
- GitHub Copilot prompt library

Week 2: Advanced C#, LINQ, Async Programming & Enterprise Git

Modules & Topics

Module 1: Advanced C# Programming

- Generics
- Generic constraints
- Delegates
- Events
- Func and Action
- Lambda expressions

- Extension methods
- Attributes
- Reflection

Module 2: LINQ

- LINQ architecture
- Query syntax vs method syntax
- Filtering and projections
- Grouping and joins
- Aggregation
- Deferred execution
- IEnumerable vs IQueryable
- Performance considerations
- Custom LINQ extensions

Module 3: Asynchronous Programming

- Threading fundamentals
- Task-based asynchronous programming
- async/await
- Task vs ValueTask
- ConfigureAwait
- CancellationToken
- Exception handling
- Parallel operations
- Deadlock prevention

Module 4: Enterprise Git & GitHub

- Repository management
- Feature branching
- GitFlow concepts
- Pull requests
- Code reviews
- Merge conflict resolution
- Interactive rebase
- Squash commits
- Cherry-pick
- Tags and releases
- Branch protection strategy

Hands-on Labs

- Develop LINQ queries over enterprise datasets.
- Create asynchronous processing services.
- Implement cancellation and exception handling.
- Create feature branches.
- Simulate multi-developer Git workflow.
- Resolve merge conflicts.
- Perform rebase and squash.

Incremental Project

Add asynchronous business services and establish the enterprise Git/GitHub workflow for the case-study project.

Key Deliverables

- Async C# processing library
 - Optimized LINQ implementation
 - Multi-branch Git repository with clean PR history
-

Week 3: MSSQL, Entity Framework Core, Couchbase & Offline Persistence

Modules & Topics

Module 1: Microsoft SQL Server

- Relational database concepts
- Database/schema design
- Tables and relationships
- Constraints
- Views
- Stored procedures
- Indexes
- Query optimization
- Execution plans

Module 2: Entity Framework Core

- DbContext
- Entities
- Relationships
- Fluent API
- Code First
- Migrations
- Tracking and no-tracking queries
- Raw SQL
- Transactions

Module 3: Enterprise Data Access Patterns

- Repository Pattern
- Generic Repository
- Unit of Work
- Domain vs persistence models
- DTOs
- Data mapping

Module 4: Couchbase Lite & Mobile NoSQL

- NoSQL fundamentals
- Couchbase architecture

- JSON documents
- Collections and scopes
- CRUD operations
- Key-value access
- SQL++/N1QL concepts
- Indexing

Module 5: Offline-First Data Architecture

- Offline storage
- Local caching
- Change listeners
- Synchronization concepts
- Conflict handling
- Local state management
- Online/offline detection strategies

Module 6: Copilot for Database Development

- Generating SQL
- Refactoring queries
- Index suggestions
- Stored procedure generation
- Couchbase query assistance
- Reviewing AI-generated database code

Hands-on Labs

- Design enterprise MSSQL schema.
- Build EF Core models.
- Generate migrations.
- Implement Repository + Unit of Work.
- Configure Couchbase Lite.
- Implement offline CRUD.
- Perform local queries and conflict-handling exercises.

Incremental Project

Add MSSQL persistence and Couchbase-based offline persistence to the case-study application.

Week 4: ASP.NET Core APIs, Microservices & Azure Fundamentals

Modules & Topics

Module 1: ASP.NET Core Web API

- API architecture
- Controllers
- Routing
- HTTP methods
- Request/response models

- Status codes
- Model validation
- Exception handling
- Middleware

Module 2: Swagger & OpenAPI

- Swagger UI
- OpenAPI specification
- API documentation
- API versioning
- Contract-first API concepts
- Client generation

Module 3: Secure API Development

- Authentication vs authorization
- JWT
- OAuth 2.0 fundamentals
- API keys
- HTTPS/TLS
- Claims and policies
- Secure configuration

Module 4: Reliable API Integration

- HttpClient
- IHttpClientFactory
- JSON serialization
- Timeout management
- Polly
- Retry policies
- Circuit breaker
- Resilience patterns

Module 5: Microservices Fundamentals

- Monolith vs microservices
- Service boundaries
- Service decomposition
- API Gateway
- Distributed systems concepts
- Event-driven systems
- Inter-service communication
- Contract-first development

Module 6: Cloud Fundamentals

- Traditional infrastructure vs cloud
- Scalability
- Elasticity
- Availability

- IaaS
- PaaS
- SaaS
- Public/private/hybrid cloud
- Shared responsibility

Module 7: Microsoft Azure Fundamentals

- Subscription
- Resource groups
- Regions
- Azure Resource Manager
- Azure App Service
- Azure SQL
- Azure Storage
- Azure Key Vault
- Application Insights
- AKS overview

Hands-on Labs

- Create ASP.NET Core REST API.
- Configure Swagger.
- Implement JWT authentication.
- Consume APIs through IHttpConnectionFactory.
- Configure Polly retry and circuit breaker.
- Design Azure deployment architecture.

Incremental Project

Expose the case-study business services through secure REST APIs and create the initial Azure architecture.

Week 5: Native Android & Windows Development Environment

Modules & Topics

Module 1: Windows Development Environment

- Windows development prerequisites
- .NET SDK installation
- Visual Studio workloads
- Environment configuration
- Android workload configuration

Module 2: Android Environment

- Android SDK
- SDK Manager
- Build tools
- Emulator

- Android Virtual Devices
- Android Studio overview

Module 3: ADB

- Device discovery
- Emulator connectivity
- Installing applications
- Log collection
- Package management
- Basic troubleshooting

Module 4: Android Lifecycle

- Activity lifecycle
- Fragment lifecycle
- Application lifecycle
- State management
- Process termination
- Configuration changes

Module 5: Android Architecture Components

- ViewModel
- LiveData concepts
- Navigation
- WorkManager

Module 6: Android Platform Features

- Permissions
- Storage Access Framework
- Background services
- Foreground services
- Device capabilities

Module 7: Android Debugging

- Logcat
- CPU profiler
- Memory profiler
- Battery profiler
- Crash troubleshooting

Hands-on Labs

- Configure Windows + Android development environment.
- Create and configure AVD.
- Use ADB.
- Develop multi-screen Android application.
- Implement permissions.
- Execute background operations.
- Analyze memory and CPU usage.

Incremental Project

Validate Android environment and prototype the platform-specific features required by the case study.

Week 6: .NET MAUI, XAML, MVVM & Application Architecture

Modules & Topics

Module 1: .NET MAUI Architecture

- Single-project architecture
- Multi-targeting
- Platform abstraction
- Platform-specific folders
- Native handlers

Module 2: XAML

- XAML syntax
- Grid
- StackLayout
- FlexLayout
- ScrollView
- CollectionView
- Responsive UI design

Module 3: Resources & Styling

- Resource dictionaries
- Static resources
- Dynamic resources
- Styles
- Dark/light themes
- Reusable UI resources

Module 4: MVVM

- View
- ViewModel
- Model
- INotifyPropertyChanged
- ICommand
- RelayCommand
- AsyncCommand
- Separation of concerns

Module 5: Practical Data Binding

- One-way
- Two-way

- OneTime
- Compiled binding
- x:DataType
- Validation
- Essential converters

Module 6: Shell Navigation

- AppShell
- Routes
- Navigation
- Query parameters
- Deep-linking concepts

Module 7: MAUI Community Toolkit

- ObservableProperty
- RelayCommand generators
- Behaviors
- Converters
- Common helpers

Module 8: Copilot for MAUI

- Generating ViewModels
- XAML assistance
- Refactoring UI code
- Generating repetitive components
- AI-code validation

Hands-on Labs

- Build Android/Windows MAUI application.
- Create responsive XAML screens.
- Implement MVVM.
- Configure bindings.
- Implement validation.
- Configure Shell navigation.
- Use Community Toolkit.

Incremental Project

Transform the case-study UI into a functional .NET MAUI module connected to enterprise services.

Week 7: Linux, Shell Scripting, Docker, Kubernetes, Rancher, Azure & Kafka

Modules & Topics

Module 1: Linux Administration for Developers

- Linux architecture
- Filesystem navigation

- Users and groups
- Permissions
- Processes
- Services
- systemd fundamentals
- Networking commands
- Log files
- Disk and memory monitoring
- Package management

Module 2: Linux Patching

- Package repositories
- Updating packages
- Security updates
- Patch planning
- Pre/post-patch checks
- Service verification
- Rollback awareness
- Production patching considerations

Module 3: Shell Scripting for Production Automation

- Bash fundamentals
- Variables
- Conditions
- Loops
- Functions
- Parameters
- Exit codes
- Environment variables
- File processing
- Log processing
- Automation scripts
- Error handling
- Cron jobs
- Production deployment scripts

Module 4: Docker

- Images
- Containers
- Dockerfile
- Layers
- Volumes
- Ports
- Environment variables
- Networking
- Docker build/run
- Containerizing ASP.NET Core

Module 5: Kubernetes

- Kubernetes architecture
- Pod
- Deployment
- ReplicaSet
- Service
- ConfigMap
- Secret
- YAML manifests
- Health checks
- Scaling fundamentals

Module 6: Rancher

- Rancher overview
- Kubernetes cluster visibility
- Cluster/project concepts
- Namespace management
- Workload deployment
- Application monitoring
- ConfigMaps and secrets
- Developer-level troubleshooting

Module 7: Azure Kubernetes Service

- AKS architecture
- Azure Container Registry
- Deployment flow
- Scaling concepts
- Application configuration
- Health monitoring

Module 8: Azure Key Vault

- Secrets
- Keys
- Certificates
- Application integration
- Managed identity concepts
- Eliminating hardcoded credentials

Module 9: Kafka for .NET Developers

- Brokers
- Topics
- Partitions
- Producers
- Consumers
- Consumer groups
- Message ordering
- .NET producer/consumer

- Deployment awareness

Hands-on Labs

- Linux administration exercises.
- Write Bash automation scripts.
- Perform sample Linux patch workflow.
- Containerize ASP.NET Core application.
- Write Kubernetes manifests.
- View/manage workload through Rancher.
- Deploy application architecture to AKS.
- Integrate secure configuration.
- Implement Kafka producer and consumer.

Incremental Project

Establish the Linux, automation, containerization, Kubernetes/Rancher and messaging foundation for the application backend.

Week 8: Xamarin Foundations & Migration to .NET MAUI

Modules & Topics

Module 1: Xamarin Foundations

- Xamarin architecture
- Xamarin.Forms
- Shared projects
- XAML
- MVVM
- Xamarin.Essentials
- DependencyService
- Custom Renderers

Module 2: Xamarin vs .NET MAUI

- Architectural comparison
- Project structure differences
- Namespace changes
- API mapping
- Single-project model

Module 3: Migration Assessment

- Application inventory
- NuGet dependency analysis
- Compatibility assessment
- Breaking changes
- Migration planning

Module 4: Upgrade Assistant

- .NET Upgrade Assistant
- Automated migration
- Manual conversion
- Project-file changes

Module 5: Renderer Migration

- Xamarin Custom Renderers
- .NET MAUI Handlers
- Mapper architecture
- Platform customization

Module 6: DependencyService Migration

- DependencyService
- IServiceProvider
- Dependency injection
- Platform service abstraction

Module 7: Third-Party Migration

- NuGet package compatibility
- Replacement libraries
- Conditional compilation
- Platform-specific code

Hands-on Labs

- Analyze Xamarin.Forms application.
- Run Upgrade Assistant.
- Resolve migration issues.
- Replace DependencyService.
- Convert custom renderer into handler.
- Upgrade incompatible packages.

Incremental Project

Migrate a representative Xamarin.Forms module and integrate it into the ongoing .NET MAUI solution.

Week 9: Unit Testing, TDD, Mocking & BDD

Modules & Topics

Module 1: Unit Testing

- Testing principles
- xUnit
- NUnit overview
- Test structure
- AAA pattern
- Assertions

- Parameterized tests

Module 2: Mocking

- Moq
- Mocking interfaces
- Service isolation
- HTTP mocking
- Repository mocking
- ViewModel testing

Module 3: Test-Driven Development

- Red
- Green
- Refactor
- Test-first development
- Designing testable applications

Module 4: BDD & Gherkin

- BDD concepts
- Given-When-Then
- Feature files
- Scenario Outline
- Reqnroll
- Step definitions

Module 5: Code Coverage

- Coverage measurement
- Coverage reports
- Critical code coverage
- Coverage gates

Module 6: GitHub Copilot for Testing

- Test case generation
- Edge-case suggestions
- Mock generation
- Gherkin drafting
- Reviewing AI-generated tests

Hands-on Labs

- Write unit tests.
- Mock application services.
- Perform TDD exercise.
- Develop Gherkin feature files.
- Automate BDD scenarios.
- Generate coverage reports.

Incremental Project

Build comprehensive automated tests for business services and MAUI ViewModels.

Target

>80% code coverage for core business assemblies

Week 10: DevSecOps, Vulnerability Scanning, Observability & Troubleshooting

Modules & Topics

Module 1: SonarQube

- Static code analysis
- Quality profiles
- Quality gates
- Code smells
- Technical debt
- Cognitive complexity
- Coverage integration

Module 2: CheckmarxOne

- SAST concepts
- Vulnerability scanning
- SQL Injection
- XSS
- Hardcoded secrets
- Insecure data storage
- Security remediation

Module 3: Vulnerability Scanning

- Application vulnerability scanning
- Dependency vulnerability scanning
- Container-image scanning concepts
- CVE and CVSS basics
- Dependency/package vulnerabilities
- Base image vulnerabilities
- Vulnerability prioritization
- Remediation workflow
- Security quality gates

Module 4: Advanced Debugging

- Conditional breakpoints
- Function breakpoints
- Watch windows

- Call stack
- Exception settings
- Threads
- Tasks
- Memory debugging
- Dumps
- Crash analysis
- Visual Studio Diagnostic Tools

Module 5: Structured Logging

- Serilog
- Log levels
- Structured events
- Correlation IDs
- Traceability

Module 6: Prometheus + Grafana + Azure Monitoring

- Monitoring vs logging vs tracing
- Application metrics
- Prometheus fundamentals
- Metrics collection
- Grafana dashboards
- Azure Monitor
- Application Insights
- Log Analytics concepts
- Alerts
- Service health
- Latency/error monitoring
- Correlating Azure telemetry with application metrics

Module 7: Mobile Security

- SecureStorage
- Android KeyStore
- Keychain concepts
- Certificate pinning
- Encryption fundamentals
- Obfuscation concepts

Hands-on Labs

- Scan code using SonarQube.
- Perform CheckmarxOne vulnerability analysis.
- Review dependency/container vulnerabilities.
- Fix identified vulnerabilities.
- Configure quality gates.
- Troubleshoot a runtime problem.
- Add Serilog logging.
- Review Prometheus/Grafana and Azure monitoring dashboards.

Incremental Project

Implement security scanning, vulnerability remediation, quality gates, secure storage and operational monitoring.

Week 11: CI/CD, Azure DevOps, GitHub Actions, Jenkins & Terraform

Modules & Topics

Module 1: CI/CD Fundamentals

- Continuous Integration
- Continuous Delivery
- Continuous Deployment
- Pipeline architecture
- Build stages
- Testing stages
- Security stages
- Deployment stages

Module 2: GitHub Actions

- YAML
- Workflows
- Events
- Jobs
- Steps
- Actions
- GitHub-hosted runners
- Self-hosted runners

Module 3: GitHub Actions for .NET

- Checkout
- Setup .NET
- Restore
- Build
- Test
- Code coverage
- SonarQube
- Vulnerability scanning
- Android build
- Windows build
- Artifact generation

Module 4: Azure DevOps + GitHub Actions

- Azure DevOps overview
- Azure Repos concepts
- Azure Boards overview
- Azure Pipelines fundamentals

- GitHub Actions comparison
- GitHub + Azure deployment integration
- Environment approvals
- Pipeline variables
- Secrets
- Deployment environments

Module 5: Jenkins Fundamentals

- Jenkins architecture
- Controller and agents
- Jobs
- Plugins
- Jenkinsfile
- Declarative pipeline
- Build stages
- Webhooks
- Credentials
- Pipeline triggers
- Basic .NET pipeline
- Jenkins vs GitHub Actions vs Azure Pipelines

Module 6: Terraform with Azure

- Infrastructure as Code
- Terraform architecture
- Providers
- Resources
- Variables
- Outputs
- Terraform state
- init
- plan
- apply
- destroy
- AzureRM provider
- Resource Groups
- App Service
- Storage
- Azure SQL concepts
- Key Vault concepts
- AKS infrastructure concepts

Module 7: CI/CD Security

- Repository secrets
- Environment secrets
- Azure Key Vault integration
- Service principals/managed identity concepts
- Android signing credentials

- Secure pipeline practices

Module 8: Release & Deployment

- Build artifacts
- Versioning
- Docker image build
- Container registry
- Kubernetes/AKS deployment hand-off
- Environment promotion
- Rollback concepts

Hands-on Labs

- Create GitHub Actions CI pipeline.
- Configure PR validation.
- Execute build/test/security scan.
- Generate Android/Windows artifacts.
- Build a foundation Azure DevOps pipeline.
- Create a basic Jenkins pipeline.
- Build Azure infrastructure with Terraform.
- Container image build and deployment workflow.

Incremental Project

Implement automated CI/CD and Infrastructure-as-Code for the case-study solution.

Key Deliverable

A pipeline demonstrating:

Code → Build → Test → Coverage → SonarQube → Vulnerability Scan → Artifact → Docker Image → Deployment Hand-off

Week 12: Knowledge Transfer, Agile Delivery & Capstone

Modules & Topics

Module 1: Existing Application Knowledge Transfer

- Existing application architecture
- Business workflow
- Functional modules
- Integration dependencies
- Data flow
- Operational processes

Module 2: Agile Delivery

- Agile/Scrum
- Sprint planning
- Daily Scrum
- Backlog grooming

- User stories
- Acceptance criteria
- Retrospectives
- Definition of Done

Module 3: Jira

- Epics
- Stories
- Tasks
- Bugs
- Sprint boards
- Tracking

Module 4: Confluence

- Technical documentation
- Architecture documentation
- KT documentation
- Runbooks
- Release notes

Module 5: Capstone Integration

Participants integrate:

- .NET MAUI
- Android/Windows
- MVVM
- MSSQL
- Couchbase offline persistence
- ASP.NET Core REST APIs
- Authentication
- Kafka
- Docker
- Kubernetes
- AKS
- Rancher concepts
- Azure Key Vault
- Terraform
- GitHub Actions/Azure DevOps
- Jenkins awareness
- SonarQube
- CheckmarxOne
- Vulnerability scanning
- Prometheus/Grafana
- Azure Monitor/Application Insights
- Automated testing

Module 6: Production Readiness

- Code review

- Security validation
- Performance checks
- Monitoring readiness
- Deployment readiness
- Documentation review
- Handover checklist

Module 7: Technical Defense & Evaluation

- Solution demonstration
- Architecture presentation
- Code walkthrough
- Security justification
- Testing results
- CI/CD demonstration
- Troubleshooting questions
- Panel evaluation

Capstone

Develop an **Enterprise Retail Service Application** consisting of:

Frontend

- .NET MAUI
- Android and Windows
- MVVM
- Responsive XAML
- Offline support

Data

- MSSQL
- EF Core
- Couchbase Lite

Backend

- ASP.NET Core REST APIs
- JWT authentication
- Kafka integration

Cloud & Infrastructure

- Docker
- Kubernetes
- AKS
- Rancher awareness
- Terraform
- Azure Key Vault

DevSecOps

- GitHub Actions

- Azure DevOps
- Jenkins fundamentals
- SonarQube
- CheckmarxOne
- Vulnerability scanning

Monitoring

- Prometheus
- Grafana
- Azure Monitor
- Application Insights

Quality

- Unit testing
- TDD
- BDD