Boilerplate Adoption Roadmap

Session-1	Session-2	Session-3	Session-4	Session-5
Asp.Net Core 6.0	EF Core 6.0	CQRS & MediatR	Authentication & Authorization	Asp.Net Core API

Roadmap for Team Leads
- **Facilitator**: Team Leads
- **Audience**: Team Members

Session-10	Session-9	Session-8	Session-7	Session-6
CI-CD	Integration Tesing	Unit Tesing	Docker	HealthCheckUI

Session-11	Session-12
Clean Architecture	Boilerplate

Roadmap for CoE Leads
- **Facilitator**: CoE Leads
- **Audience**: Team Leads & Eligible Team Members

Roadmap Guidelines for Dot Net Team Leads

Sessions	Topics	To-Do
1	**Asp.Net Core 6.0** • Dependency Injection (services) • Middleware • Configuration • Environments • Routing • Error Handling • SOLID Principle <table><tr><td>References</td><td>Ref-1</td></tr><tr><td>Efforts</td><td>1hr</td></tr><tr><td>Facilitator</td><td>Team Lead</td></tr><tr><td>Audience</td><td>Team Members</td></tr></table>	Create Web Project • Understand Program.cs • Understand appsettings.json • Understand Environments • Understand Middleware's Create Interface Create Service • Read Value from appsettings.json and Environment (Demo purpose) Create Controller • Inject Service • Register Service Create Action Method • Invoke Service in Action Method Exception Handling • Global Exception Handling Run Project

2	**EF Core 6.0 (Code First Approach)** • CRUD • Scaffold, Add, Update, Migration • Repository Pattern • Automapper 	References	Ref-1
Efforts	2hrs
Facilitator	Team Lead
Audience	Team Members		Create Web Project Create Persistence Layer (Add Project) • Install EF Core via NuGet • Add Entities • Create DbContext • Add Migration • Update Database Go to Web Project > Add Persistence Project Reference • Create Controller • Create Action Method • Create Interface • Create Service • Map Entities Using Automapper • Implement CRUD in Service (Without Repository) Run Project Replicate above steps with Repository Pattern
3	**CQRS & MediatR Pattern** • Context and problem • Solution • Implementation issues and considerations • When to use CQRS pattern • Event Sourcing and CQRS pattern • Implement CQRS using MediatR 	References	Ref-1 Ref-2
Efforts	1hr
Facilitator	Team Lead
Audience	Team Members		Create Web Project Create Persistence Layer Implement CQRS using MediatR Run Project

4	**Asp.Net Core API** • Swagger Implementation • Logging Implementation • Call API's using Postman • API Versioning • Request-Response • Encoding-Decoding | References	Ref-1, Ref-2, Ref-3, Ref-4, Ref-5
Efforts	1hr
Facilitator	Team Lead
Audience	Team Members		Create Web API Project Implement Swagger Implement Logger Implement API Versioning Run Project
5	**Authentication & Authorization using JWT** • Authentication • Authorization • JWT • JWT Bearer • User Manager • Sign-In Manager | References	Ref-1, Ref-2, Ref-3, Ref-4, Ref-5
Efforts	2hrs
Facilitator	Team Lead
Audience	Team Members		Explain Mechanism of Authentication and Authorization Explain Identity Provider Explain Token Based Authentication Refresh Token Implement Authentication & Authorization in API
6	**Health Monitoring (Optional)** • HealthCheck • HealthCheckUI • HealthCheck Packages • Watchdogs • Monitoring Health | References	Ref-1, Ref-2
Efforts	1hr
Facilitator	Team Lead
Audience	Team Members		Create Web/API Project Install HealthCheck NuGet Packages Configure Healthcheck Demonstrate HealthCheck UI

7	**Docker (Optional)** • Context and problem • Why Docker? • What is Docker? • How Docker Works? • Usage of Docker 	References	Ref-1	 |---|---|
Efforts	1hr
Facilitator	Team Lead
Audience	Team Members		Install Docker Desktop Go thru Docker Commands Create Demo API Project Create Docker Image for API Run API in Docker Push images to DocketHub
8	**Automated Testing (Optional)** • Unit Tests • Integration Tests | References	Ref-1
Efforts	1hr
Facilitator	Team Lead
Audience	Team Members		Create Demo API Project • Write Unit Test Cases for API • Run Unit Tests Create Demo API Project • Write Integration Test Cases to check if API endpoint is able to connect to DB which is running on different Server (use Docker to simulate this) • Run Integration Tests
9	**CI-CD (with GitHub Actions) (Optional)** • Continuous Integration • Continuous Delivery/Deployment • GitHub Actions • Workflows | References	Ref-1
Efforts	2hrs
Facilitator	Team Lead
Audience	Team Members		Create Repository in GitHub Create Demo API Project • Write Unit Test Cases for API Create Demo API Project • Write Integration Test Cases to check if API endpoint is able to connect to DB which is running on different Server (use Docker to simulate this) Push Code to Repo Setup CI-CD Pipeline using GitHub Actions • Configure Actions • Build App • Run Unit Test • Run Integration Test Push Code to Main/Master Branch Demonstrate CI-CD Pipeline