

# ASP.NET Core MVC



- [Güncel PDF'i İndir](#)
- [En güncel eğitimlerimiz için www.vebende.com.tr](http://www.vebende.com.tr) ziyaret edin

## Eğitim Süresi

- **Süre:** 20 gün
- **Ders Süresi:** 50 dakika
- **Eğitim Saati:** 10:00 - 17:00
- Eğitim formatında eğitimler 50 dakika + 10 dakika moladır. 12:00-13:00 saatleri arasında 1 saat yemek arasındaki verilir. Günde toplam 6 saat eğitim verilir. 20 günlük formatta 120 saat eğitim verilmektedir.
- Eğitimler uzaktan eğitim formatında tasarlanmıştır. Her eğitim için teams linkleri gönderilir. Katılımcılar bu linklere girerek eğitimlere katılırlar. Ayrıca farklı remote çalışma araçları da eğitmen tarafından tüm katılımlara sunulur. Katılımcılar bu araçları kullanarak eğitimlere katılırlar.

- Eğitim içeriğinde github ve codespace kullanılır. Katılımcılar bu platformlar üzerinden örnek projeler oluşturur ve eğitimle birlikte eğitimlerde sorulan sorulara ve taleplere uygun içeriğe cevap verir. Katılımcılar bu araçlarla eğitimlerde sorulan sorulara ve taleplere uygun içeriğe cevap verir.
- Eğitim yapay zeka destekli kendi kendine öğrenme formasyonu ile tasarlanmıştır. Katılımcılar eğitim boyunca kendi kendine öğrenme formasyonu ile eğitimlere katılırlar. Bu eğitim formatı sayesinde tüm katılımcılar gelecek tüm yaşamlarında kendilerini güncellemeye devam edebilecekler ve her türlü sorunun karşısında çözüm bulabilecekleri yeteneklere sahip olacaklardır.

## ASP.NET MVC Core

ASP.NET MVC Core, Microsoft tarafından geliştirilen modern, esnek ve yüksek performanslı bir web uygulama çatısıdır. Geleneksel ASP.NET MVC'nin geliştirilmiş bir versiyonu olan bu framework, platform bağımsız çalışabilme, hafifletilmiş mimari ve daha iyi performans gibi birçok avantaj sunar. MVC (Model-View-Controller) tasarım deseni üzerine kurulu olan ASP.NET Core, web uygulamalarını modüler ve yönetilebilir hale getirerek geliştirme sürecini kolaylaştırır.

ASP.NET MVC Core, .NET Core çatısı üzerine inşa edildiği için Windows, macOS ve Linux gibi farklı işletim sistemlerinde çalışabilir. Bu sayede geliştiricilere geniş bir esneklik sunar. Performans açısından optimize edilmiş yapısı sayesinde, geleneksel ASP.NET frameworküne kıyasla daha hızlı çalışır ve daha az kaynak tüketir.

Framework'ün temel avantajlarından bazıları şunlardır:

- **Platform Bağımsızlık:** Windows dışında Linux ve macOS gibi farklı işletim sistemlerinde çalışabilir.
- **Yüksek Performans:** Daha hafif bir runtime ve optimize edilmiş işlem süreçleri sayesinde hızlıdır.
- **Bağımsız Middleware Yapısı:** Modüler tasarımı sayesinde ihtiyaç duyulan bileşenler eklenebilir veya çıkarılabilir.
- **Güçlü Dependency Injection (Bağımlılık Enjeksiyonu):** Yerleşik DI desteği ile bağımlılıkları yönetmek daha kolaydır.
- **Razor Pages Desteği:** MVC'nin yanında Razor Pages ile daha sade ve etkili sayfa yönetimi sağlanabilir.

ASP.NET MVC Core, modern web uygulamaları geliştirmek isteyenler için güçlü ve esnek bir altyapı sunar. Geliştiricilere hız, esneklik ve yönetilebilirlik avantajları sağlayarak, web projelerinin başarılı bir şekilde hayata geçirilmesini destekler.

# Eğitim Hedefi

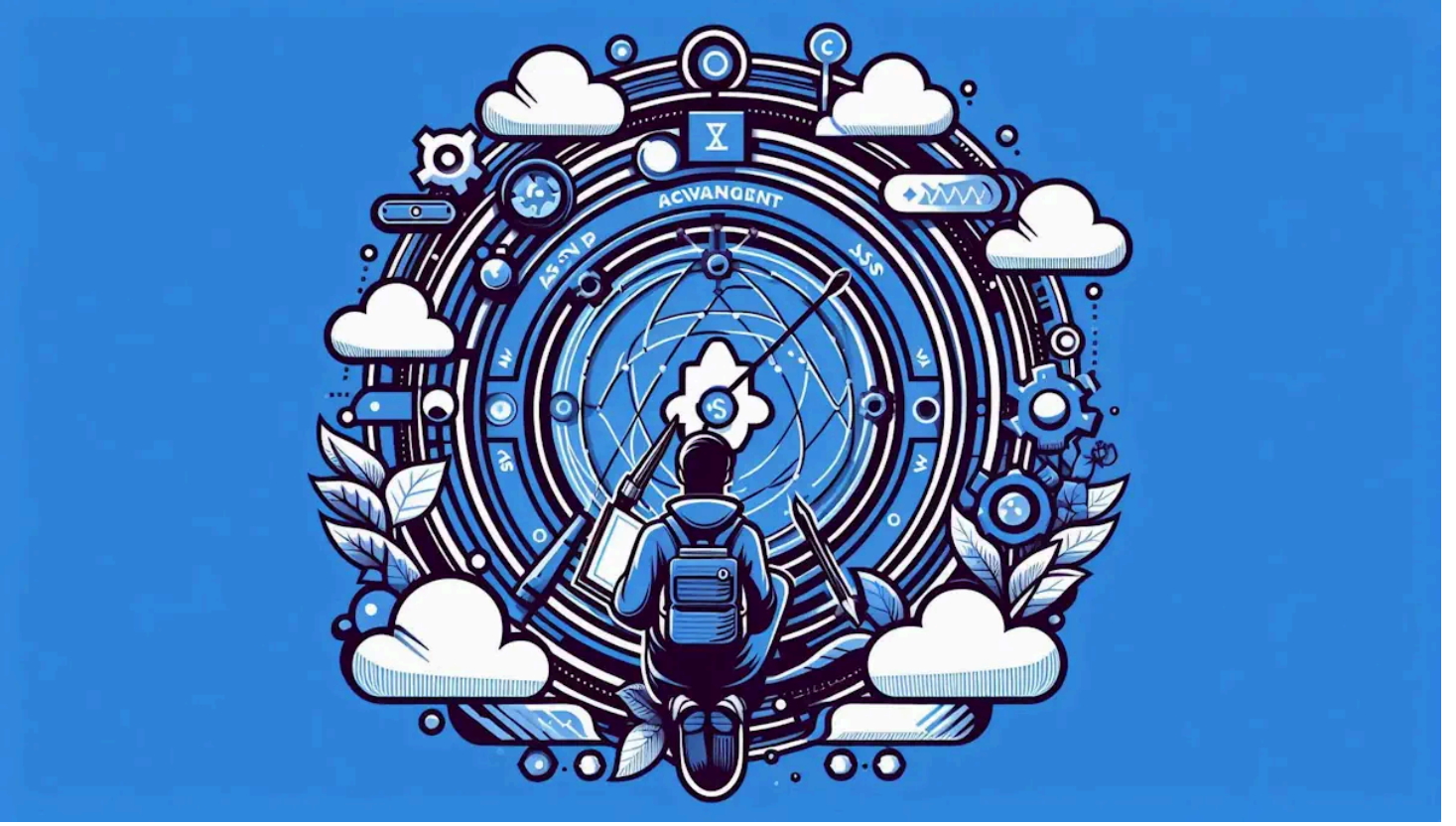
Bu eğitim, **ASP.NET MVC Core** framework'ünü kullanarak modern, performanslı ve ölçeklenebilir web uygulamaları geliştirmeyi öğretmeyi amaçlamaktadır. Katılımcılar, **MVC mimarisi**, **.NET Core altyapısı**, **Entity Framework Core**, **Dependency Injection**, **Middleware kullanımı** ve **güvenli kimlik doğrulama** gibi konulara hakim olacaklardır.

Eğitim sonunda katılımcılar:

- **ASP.NET MVC Core**'un temel yapı taşlarını anlayacak,
- **Model-View-Controller (MVC)** mimarisini etkin şekilde kullanabilecek,
- **RESTful API** geliştirme ve entegrasyon süreçlerini uygulayabilecek,
- Veritabanı işlemlerini **Entity Framework Core** ile gerçekleştirebilecek,
- **Kimlik doğrulama** ve yetkilendirme mekanizmalarını uygulayabilecek,
- **Performans optimizasyonu** ve güvenlik en iyi uygulamalarını öğrenerek projelerinde uygulayabileceklerdir.

Bu eğitim, hem **ASP.NET Core**'a yeni başlayanlar hem de mevcut bilgilerini güncellemek isteyen **geliştiriciler** için uygundur.

# Eğitim İçeriği



## ASP.NET Core MVC ile Modern Web Uygulamaları Geliştirme

### 1. Giriş ve Kurulum

- **ASP.NET** Core Nedir?
- **ASP.NET** Core'un Avantajları
- Gerekli Yazılımlar: Visual Studio, .NET SDK Kurulumu
- İlk **ASP.NET** Core Projesinin Oluşturulması
- Proje Yapısı ve Dosya Açıklamaları

### 2. C# Temelleri ve **ASP.NET** Core ile İlişkisi

- C# Dilinde Temel Kavramlar
- Değişkenler, Veri Tipleri, Döngüler, Koşul İfadeleri
- Nesne Tabanlı Programlamaya Giriş (OOP)
- C# Sınıflarının **ASP.NET** Core Projelerindeki Rolü

### 3. MVC (Model-View-Controller) Mimarisi

- MVC Nedir?
- Model, View, Controller Bileşenlerinin Açıklanması
- Basit Bir **ASP.NET** Core MVC Uygulaması Oluşturma
- Routing ve Controller İşleyişi

### 4. Model Binding ve Validation

- Model Binding Nedir? (GET ve POST İşlemleri)
- Veri Doğrulama (Data Annotations) ve Custom Validation Kullanımı
- Kullanıcı Girdilerinin Doğrulanması

### 5. Dependency Injection ve Servisler

- Dependency Injection (DI) Nedir?
- DI Kullanarak Servis Yönetimi
- **ASP.NET** Core'da Dependency Injection Yapılandırması
- Singleton, Scoped ve Transient Servisler
- Örnek Uygulama: Bir Servis ile Veri Yönetimi

### 6. Entity Framework Core ile Veri Tabanı İşlemleri

- Entity Framework Core Nedir?
- Veritabanı Bağlantısı ve Migration İşlemleri
- CRUD Operasyonları ile Veri Yönetimi
- Lazy Loading ve Eager Loading

### 7. RESTful API ve HTTP İstekleri

- RESTful API Nedir?
- Web API'lerin Temel Yapı Taşları
- **ASP.NET** Core'da API Oluşturma
- GET, POST, PUT, DELETE Metodları ile API İşlemleri
- JSON ve XML Formatlarında Veri İletimi
- API'lerde Hata Yönetimi ve Exception Handling

### 8. Authentication ve Authorization

- Kullanıcı Kimlik Doğrulama (Authentication) Nedir?
- Yetkilendirme (Authorization) Kavramı
- JWT ve Cookie Tabanlı Kimlik Doğrulama



- Role-Based Authorization Kullanımı
- **ASP.NET** Core Identity Kullanarak Kullanıcı Yönetimi
- OAuth ve OpenID Connect ile Üçüncü Parti Uygulama Entegrasyonu
- IdentityServer4 Kullanarak Merkezi Kimlik Doğrulama

## 9. **ASP.NET** Core Routing ve URL Yapıları

- Routing Kavramı Nedir?
- Custom Routing ve Route Constraints Kullanımı
- Parametrelili Route Yapıları
- Route Değerlerini İşleme
- Route Testing ve Validation

## 10. Middleware ve Pipeline Yönetimi

- Middleware Nedir ve Nasıl Çalışır?
- **ASP.NET** Core Uygulamalarında Middleware Kullanımı
- Özel Middleware Yazımı
- Pipeline Yönetimi ve Order of Execution
- Exception Handling Middleware ve Logging

## 11. WebSockets ve SignalR ile Gerçek Zamanlı Uygulamalar

- WebSockets Nedir ve Nasıl Çalışır?
- SignalR Kütüphanesinin Kullanımı
- Gerçek Zamanlı Chat Uygulaması Yapma
- WebSockets ile Veri İletimi
- SignalR ile Canlı Bildirimler, Online Oyunlar ve Yaygın Uygulama Senaryoları

## 12. Dosya Yükleme ve İndirme İşlemleri

- **ASP.NET** Core ile Dosya İşleme
- Dosya Güvenliği ve Sınırlamaları
- Dosya Boyutu ve Türü Kısıtlamaları
- Örnek Uygulama: Kullanıcı Dosya Yükleme Özelliği

## 13. Unit Test ve Test Driven Development (TDD)

- Unit Test Nedir?
- **ASP.NET** Core Projelerinde Test Yazma
- Test Framework'leri (xUnit, NUnit, MSTest)

- Mocking ve Dependency Injection ile Test Yazma
- Test Driven Development (TDD) ve En İyi Uygulamalar
- Integration Testing ve API Testleri

## 14. API Güvenliği ve Hata Yönetimi

- CORS (Cross-Origin Resource Sharing) Yönetimi
- Rate Limiting ve Throttling
- Hata Yönetimi ve Global Exception Handling
- API Authentication ve Authorization Senaryoları
- OAuth 2.0 ve API Security
- API Rate-Limiting ve Request Validation

## 15. Performans Optimizasyonu ve Cache Yönetimi

- Performans Optimizasyonu Stratejileri
- [ASP.NET](#) Core'da Cache Kullanımı
- Memory Cache ve Distributed Cache Yönetimi
- Response Compression ve Caching
- Performans Analizi Araçları
- Asenkron Programlama ve Paralel İşlem

## 16. Dağıtım ve Yayınlama

- [ASP.NET](#) Core Uygulamasını Dağıtmak
- IIS ve Kestrel Sunucusunda [ASP.NET](#) Core Yayınlama
- Docker Kullanarak Konteynerize Etme
- Cloud Platformlarına (Azure, AWS) Deploy İşlemleri
- CI/CD (Continuous Integration/Continuous Deployment) Pipeline Kurulumu
- Azure App Services ve Kubernetes ile Dağıtım

## 17. İleri Düzey Konular

- Microservices Mimarisi ve [ASP.NET](#) Core
- Event-Driven Architecture (EDA) Kullanımı
- Cloud-Native [ASP.NET](#) Core Uygulamaları
- [ASP.NET](#) Core ile GraphQL
- Dependency Injection ve Service Lifetimes İleri Düzey Uygulamaları
- Distributed Systems ve Data Consistency

## **18. Ekstra Konular (İleri Düzey)**

- **Event Sourcing ve CQRS**
- **Distributed Transactions**
- **Message Brokers ve Queue Management (RabbitMQ, Kafka)**



# Eğitim Yöntemi (**ASP.NET Core MVC**)

- **Teorik Bilgi:** **ASP.NET** Core MVC'nin temel kavramları ve mimarisi detaylı olarak anlatılacaktır.
- **Uygulamalı Örnekler:** Gerçek dünya senaryoları üzerinden **ASP.NET** Core MVC projeleri geliştirilecektir.
- **Etkileşimli Tartışmalar:** Katılımcılar, proje yapıları ve uygulama süreçleri hakkında soru-cevap şeklinde tartışmalara katılacaktır.
- **Proje Tabanlı Öğrenme:** Eğitimin sonunda, katılımcılar kendi **ASP.NET** Core MVC projelerini geliştirecekleri kapsamlı bir uygulama yapacaktır.

## Hedef Kitle

1. **Yazılım Geliştiriciler:** **ASP.NET** Core MVC ile web uygulamaları geliştirmek isteyen yazılımcılar.
2. **Backend Geliştiriciler:** **ASP.NET** Core'un sunucu tarafında veri yönetimi ve API entegrasyonlarını öğrenmek isteyenler.
3. **Full-Stack Geliştiriciler:** **ASP.NET** Core MVC'yi frontend ve backend ile bütünleşik olarak kullanmak isteyen geliştiriciler.
4. **Yeni Mezunlar ve Stajyerler:** **ASP.NET** Core MVC ile modern web geliştirme süreçlerine hakim olmak isteyenler.
5. **Proje Yöneticileri:** Yazılım geliştirme süreçlerinde **ASP.NET** Core MVC projelerini nasıl yöneteceklerini öğrenmek isteyen yöneticiler.
6. **Ekip Liderleri:** **ASP.NET** Core MVC projelerinde takım yönetimi ve kod organizasyonu konusunda bilgi sahibi olmak isteyen liderler.
7. **Danışmanlar:** **ASP.NET** Core MVC tabanlı çözümler geliştiren ve bu alanda danışmanlık yapmak isteyen profesyoneller.
8. **İş Analistleri:** **ASP.NET** Core MVC projelerinin yapısını anlamak ve proje gereksinimlerini daha iyi analiz etmek isteyen analistler.

## Katılımcılardan Beklentilerimiz (**ASP.NET Core MVC**)

- Katılımcıların temel seviyede C# ve .NET bilgisine sahip olmaları.
- Web teknolojileri (HTML, CSS, JavaScript) hakkında temel bilgi sahibi olmaları önerilir, ancak zorunlu değildir.
- MVC mimarisi ve web uygulamaları geliştirme süreçlerine ilgi duymaları.
- Aktif katılım, soru sorma ve öğrenmeye açık bir yaklaşım sergilemeleri.
-

