**IdentityApp - ASP.NET Core Identity Projesi Tanıtım ve Teknoloji Raporu**

**I. Proje Tanıtımı**

**Proje Adı:** IdentityApp  
**Kullanılan Teknolojiler:** ASP.NET Core MVC, Identity, Entity Framework Core, SQLite, Razor Pages, Bootstrap, SMTP  
**Proje Amacı:** Kimlik doğrulama ve yetkilendirme işlemlerinin profesyonel düzeyde gerçekleştirildiği bir ASP.NET Core MVC web uygulaması geliştirmek. Bu kapsamda kullanıcı yönetimi, rol yönetimi, şifre sıfırlama gibi önemli özellikler uygulanmıştır.

**II. Proje Özellikleri**

* Kullanıcı kayıt olma ve giriş yapma
* Rol bazlı yetkilendirme ([Authorize] ile)
* Admin paneli ile kullanıcı listesi ve yönetimi
* Kullanıcılara rol atama
* Şifre yenileme işlemi (token bazlı ve e-posta ile)
* E-posta servisi ile bildirim gönderme
* Admin için kullanıcı silme işlemi
* Seed verisi ile ilk admin ve rollerin otomatik oluşturulması

**III. Kullanılan Teknolojiler ve Bana Kattıkları**

**1. ASP.NET Core MVC**

* Katmanlı mimariyi etkin şekilde öğrendim.
* Razor yapısıyla sayfa yönetimini ve veri bind etmeyi kullandım.
* Routing, controller yapıları, view ve model ilişkilendirmeleri uygulamalı olarak öğrenildi.

**2. ASP.NET Core Identity**

* Kimlik yönetimi sistemlerinin temeli kavrandı.
* Kullanıcı ve rol sınıflarını özelleştirerek proje ihtiyaçlarına göre yapılandırma öğrendim.
* Kullanıcıya rol atama, rol bazlı erişim gibi işlemler pratik edildi.

**3. Entity Framework Core**

* Veritabanı işlemlerini kod üzerinden yönetmeyi öğrendim (Code First).
* Migration komutlarıyla veritabanı senkronizasyonu sağladım.
* LINQ kullanarak sorgulama ve listeleme işlemleri gerçekleştirdim.

**4. SQLite**

* Kolay kurulumlu ve geliştirici dostu bir veritabanı ile çalışmayı öğrendim.
* Proje başlangıcında hızlı prototipleme için ideal bir çözüm olduğunu fark ettim.

**5. SMTP (Mail Gönderimi)**

* Gerçek dünya senaryolarında kullanıcıya mail gönderme işlemini deneyimledim.
* Şifre sıfırlama bağlantısı oluşturma ve e-posta ile gönderme işlemi yapıldı.
* Mail güvenlik hataları ile karşılaşılarak çözüm üretildi.

**6. ViewModel Kullanımı**

* Sınıflar arası veri taşımayı daha düzenli ve güvenli hale getirmeyi öğrendim.
* Her işlem için özel ViewModel sınıfları (GirisViewModel, KayitViewModel, SifreYenileViewModel) oluşturuldu.
* Validation işlemlerini doğrudan ViewModel'e entegre ettim.

**7. Yetkilendirme ve Güvenlik**

* [Authorize] attribute ile sayfa bazlı erişim kontrolleri sağladım.
* Rol bazlı yönetim sistemi kurularak farklı yetkilere sahip kullanıcı arayüzleri geliştirildi.
* Kimlik doğrulama, parola güvenliği ve şifre sıfırlama gibi güvenlik önlemleri uygulandı.

**IV. Admin Paneli ve Rol Yönetimi**

* Admin panelinde tüm kullanıcılar listelenebilir.
* Kullanıcılara tek tıkla rol atama arayüzü sunulmuştur.
* Kullanıcı silme özelliği ile yöneticinin tam kontrol sağlaması hedeflenmiştir.
* Roller tablosu özelleştirilmiş ve Açıklama alanı eklenerek genişletilmiştir.

**V. Bu Projeden Öğrendiklerim ve Kazanımlarım**

* Kimlik yönetimi sistemlerinin temeli ve işleyişi uygulamalı olarak öğrenildi.
* ASP.NET Core mimarisi, Identity yapısı ve Entity Framework detaylı şekilde tecrübe edildi.
* Modern yazılım projelerinde kullanılan teknolojilerle (EF Core, SMTP, Identity) gerçekçi bir sistem kuruldu.
* UI, veritabanı ve iş mantığını kapsayan uçtan uca bir çözüm geliştirildi.

**VI. Gelecekte Yapılabilecek Geliştirmeler**

* Email doğrulama ile kayıt işlemi daha güvenli hale getirilebilir.
* İki faktörlü kimlik doğrulama (2FA) eklenebilir.
* Blazor veya React gibi modern frontend teknolojileri entegre edilebilir.
* Token tabanlı (JWT) API desteğiyle mobil ve SPA yapılar desteklenebilir.
* Kullanıcı aktiviteleri için loglama sistemi eklenebilir.

**Hazırlayan:**  
Kübra Durak  
Bilgisayar Mühendisliği Öğrencisi  
2025