



**KOCAELİ ÜNİVERSİTESİ
KOCAELİ MESLEK YÜKSEKOKULU
BİLGİSAYAR PROGRAMCILIĞI PROGRAMI**

PROJE RAPOR DOSYASI

HUKUK BÜROSU YÖNETİM

**203531045
Mehmet Hakkı YILDIZ**

NİSAN 2023

İçindekiler

Özet	2
Projeye Eklenen Yeni Özellikler	3
Sonuçlar.....	5
Kaynakça	6

Özet

Nisan ayındaki ilerleme;

- veri tabanı bağlantısının kurulması ve test edilmesi,
- MVC mimarisine uygun modellerin oluşturulması,
- Entity Framework Core ile Code-First approach (önce kodlama yaklaşımı) yöntemiyle veri tabanı tablolarının oluşturulması,
- güvenlik için çerez, session ve JWT token kullanımları arasındaki farkların araştırılıp öğrenilmesi,
- veri tabanında işlem yapmadan nasıl kullanıcı doğrulaması yapıldığının araştırılıp öğrenilmesi,
- Identity kütüphanesinin nasıl kullanıldığının araştırılıp öğrenilmesi ve test edilmesi,
- RoleManager, UserManager, Claim ve Principal gibi sınıfların nasıl kullanıldığının araştırılıp öğrenilmesi,
- Identity kütüphanesinin programa eklenmesi,
- IdentityDbContext oluşturulması,
- Identity ayarlarının kişileştirilmesi,
- güvenlik çerezlerinin etkinleştirilmesi,
- admin hesabının veri tabanına, çevre değişkenleri kullanılarak, güvenli bir şekilde eklenmesi

şeklindedir.

```
var baglantiStr = builder.Configuration.GetConnectionString("local")
    ?? throw new KeyNotFoundException("Baglanti metni bulunamadi.");

builder.Services.AddDbContext<VeriTabani>(ayarlar => ayarlar.UseSqlServer(baglantiStr));

builder.Services.AddIdentity<Personel, IdentityRole>(ayarlar =>
{
    ayarlar.SignIn.RequireConfirmedPhoneNumber = false;
    ayarlar.SignIn.RequireConfirmedEmail = false;
    ayarlar.SignIn.RequireConfirmedAccount = false;

    ayarlar.Password.RequireNonAlphanumeric = false;
    ayarlar.Password.RequireUppercase = false;
    ayarlar.Password.RequireLowercase = false;
    ayarlar.Password.RequiredLength = 6;
})
    .AddRoles<IdentityRole>()
    .AddDefaultTokenProviders()
    .AddEntityFrameworkStores<VeriTabani>();

builder.Services.ConfigureApplicationCookie(ayarlar =>
{
    ayarlar.LogoutPath = "/profil/cikis";
})
```

Projeye Eklenen Yeni Özellikler

İlk olarak modeller oluşturuldu. Oluşturulan modeller aşağıdaki gibidir:

- Belge
- BolgeAdliyeMahkemesiBilgileri
- Dosya
- DosyaBaglantisi
- DosyaBelgesi
- DosyaDurumu
- DosyaFinansIslemi
- DosyaGorevi
- DosyaKategorisi
- DosyaSorumlusu
- DosyaTuru
- Durusma
- DurusmaAktiviteTuru
- Duyuru
- DuyuruKategorisi
- FinansIslemi
- FinansIslemTuru
- Gorev
- GorevDurumu
- KararBilgileri
- KararDuzeltmeBilgileri
- KesinlesmeBilgileri
- Kisi
- KisiBaglantisi
- KisiBelgesi
- KisiFinansIslemi
- KisiGorevi
- MahkemeBilgileriBase
- Ofis
- Personel
- PersonelFinansIslemi
- Randevu
- SorumluPersonel
- TarafKisi
- TarafTuru
- TemyizBilgileri
- Vekaletname

İkinci olarak, IdentityDbContext oluşturularak kullanıcıların giriş yapmasını sağlayan bilgiler uygulamaya entegre edildi.

```
using HukukBuro.Models;
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;

namespace HukukBuro.Data;

public sealed class VeriTabani : IdentityDbContext<Personel>
{
    public VeriTabani(DbContextOptions<VeriTabani> ayarlar) : base(ayarlar)
    {
    }

    public DbSet<Duyuru> Duyurular { get; set; }
    public DbSet<DuyuruKategorisi> DuyuruKategorileri { get; set; }
    public DbSet<BolgeAdliyeMahkemesiBilgileri> BolgeAdliyeMahkemesiBilgileri { get; set; }
    public DbSet<Dosya> Dosyalar { get; set; }
    public DbSet<DosyaBaglantisi> DosyaBaglantilari { get; set; }
    public DbSet<DosyaBelgesi> DosyaBelgeleri { get; set; }
    public DbSet<DosyaDurumu> DosyaDurumu { get; set; }
    public DbSet<DosyaFinansIslemi> DosyaFinansIslemleri { get; set; }
    public DbSet<DosyaGorevi> DosyaGorevleri { get; set; }
    public DbSet<DosyaKategorisi> DosyaKategorileri { get; set; }
    public DbSet<DosyaSorumlusu> DosyaSorumlulari { get; set; }
    public DbSet<DosyaTuru> DosyaTurleri { get; set; }
    public DbSet<Durusma> Durusmalar { get; set; }
    public DbSet<DurusmaAktiviteTuru> DurusmaAktiviteTurleri { get; set; }
    public DbSet<FinansIslemTuru> FinansIslemTurleri { get; set; }
    public DbSet<KisiGorevi> KisiGorevleri { get; set; }
    public DbSet<GorevDurumu> GorevDurumlari { get; set; }
    public DbSet<KararBilgileri> KararBilgileri { get; set; }
    public DbSet<KararDuzeltmeBilgileri> KararDuzeltmeBilgileri { get; set; }
    public DbSet<KesinlesmeBilgileri> KesinlesmeBilgileri { get; set; }
    public DbSet<Kisi> Kisiler { get; set; }
    public DbSet<KisiBaglantisi> KisiBaglantilari { get; set; }
    public DbSet<KisiBelgesi> KisiBelgeleri { get; set; }
    public DbSet<KisiFinansIslemi> KisiFinansIslemleri { get; set; }
    public DbSet<Ofis> Ofis { get; set; }
    public DbSet<PersonelFinansIslemi> PersonelFinansIslemleri { get; set; }
    public DbSet<Randevu> Randevular { get; set; }
    public DbSet<SorumluPersonel> SorumluPersonel { get; set; }
    public DbSet<TarafKisi> TarafKisiler { get; set; }
    public DbSet<TarafTuru> TarafTurleri { get; set; }
    public DbSet<TemyizBilgileri> TemyizBilgileri { get; set; }
    public DbSet<Vekaletname> Vekaletnameler { get; set; }
}
```

Ardından, kullanıcı girişi için bazı ayarlar yapıldı. Bu ayarlar;

- telefon numarası doğrulamasının devre dışı bırakılması,
- email doğrulamasının devre dışı bırakılması,
- parolanın uzunluğu ve içinde bulundurabileceği karakterlerin ayarlanması

şeklinde.

Ve son olarak da admin hesabının bilgilerinin çevre değişkenleri ile veri tabanına aktarılması sağlandı.

Sonuçlar

Sonuç olarak; modeller tamamlandı, kullanıcı girişi ve kontrolcülerin programlanması için uygun altyapı oluşturuldu. Güvenlik ile ilgili yeterli araştırma yapıldı ve bilgi edinildi.

Önümüzdeki süreçte yapılması planlanan işlemler;

- kullanıcı girişinin programlanması,
- kişi kontrolcüsünün ve gerekli formların oluşturulması,
- dosya kontrolcüsünün ve gerekli formların oluşturulması,
- kullanıcı rolleri kullanarak, programın daha güvenli hale getirilmesi

şeklindedir.

Kaynakça

Entity Framework Core kullanımı, <https://learn.microsoft.com/en-us/aspnet/core/data/ef-mvc/intro?view=aspnetcore-7.0>, Erişim Tarihi: 12.04.2023

Code-First Approach, <https://learn.microsoft.com/en-us/aspnet/core/data/ef-mvc/migrations?view=aspnetcore-7.0>, Erişim Tarihi: 14.04.2023

Asp.Net Core Identity, <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-7.0>, Erişim Tarihi: 20.04.2023