

UYGULAMA 1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Http;
using uyg01.ViewModel;

namespace uyg01.Controllers
{
    public class ServisController : ApiController
    {
        [HttpGet]
        [Route("api/test")]
        public string Deneme()
        {
            string ad = "Ali";
            return ad;
        }
        [HttpGet]
        [Route("api/yetkili")]
        [Authorize]
        public string yetkili()
        {
            string ad = "Ali";
            return ad;
        }
        [HttpGet]
        [Route("api/sayiuret")]
        public int sayi()
        {
            Random r = new Random();
            int s = r.Next(100);
            return s;
        }
        [HttpGet]
        [Route("api/isimliste")]
        public List<string> IsimListeYaz()
        {
            List<string> liste = new List<string>();
            liste.Add("Ali");
            liste.Add("Veli");
            liste.Add("Selami");
            liste.Add("Ayşe");
            liste.Add("Fatma");
            liste.Add("Hayriye");

            return liste;
        }
        [HttpGet]
        [Route("api/sayiliste")]
        public List<int> SayiListeYaz()
        {
            List<int> liste = new List<int>();
            Random r = new Random();
            for (int i = 0; i < 10; i++)
            {
                int s = r.Next(100);
                liste.Add(s);
            }
        }
    }
}
```

```
        return liste;
    }

    [HttpGet]
    [Route("api/tekmiciftmi/{sayi}")]
    public string TekmiCiftmi(int sayi)
    {
        string sonuc = "Sayı Yok";
        if (sayi % 2 == 0)
        {
            sonuc = "Sayı Çift";
        }
        else
        {
            sonuc = "Sayı Tek";
        }
        return sonuc;
    }

    [HttpGet]
    [Route("api/ortalama/{vnot}/{fnot}")]
    public double Ortalama(int vnot, int fnot)
    {
        double ort = (vnot * 0.4) + (fnot * 0.6);
        return Math.Round(ort, 2);
    }

    List<kayitModel> kayitlar = new List<kayitModel> {
        new kayitModel() { id = 1, adsoyad = "Ali", mail = "ali@mail.com", yas = 19 },
        new kayitModel() { id = 2, adsoyad = "Veli", mail = "veli@mail.com", yas = 20 },
        new kayitModel() { id = 3, adsoyad = "Selami", mail = "selami@mail.com", yas = 21 },
        new kayitModel() { id = 4, adsoyad = "Ayşe", mail = "ayse@mail.com", yas = 19 },
        new kayitModel() { id = 5, adsoyad = "Fatma", mail = "fatma@mail.com", yas = 20 },
        new kayitModel() { id = 6, adsoyad = "Hayriye", mail = "hayriye@mail.com", yas = 2 },
    };

    [HttpGet]
    [Route("api/kayitliste")]
    public List<kayitModel> KayitListe()
    {
        return kayitlar;
    }

    [HttpGet]
    [Route("api/kayit/{id}")]
    public kayitModel KayitById(int id)
    {
        kayitModel kayit = kayitlar.Where(s => s.id == id).SingleOrDefault();
        return kayit;
    }
}
```

UYGULAMA 2

```

using System.Collections.Generic;
using System.Linq;
using System.Web.Http;
using uyg02.Models;
using uyg02.ViewModel;

namespace uyg02.Controllers
{
    public class ServisController : ApiController
    {
        DB01Entities db = new DB01Entities();
        SonucModel sonuc = new SonucModel();

        [HttpGet]
        [Route("api/ogrenciliste")]
        public List<OgrenciModel> OgrenciListe()
        {
            List<OgrenciModel> liste = db.Ogrenci.Select(x => new OgrenciModel()
            {
                ogrId = x.ogrId,
                ogrNo = x.ogrNo,
                ogrAdsoyad = x.ogrAdsoyad,
                ogrMail = x.ogrMail,
                ogrYas = x.ogrYas
            }).ToList();

            return liste;
        }

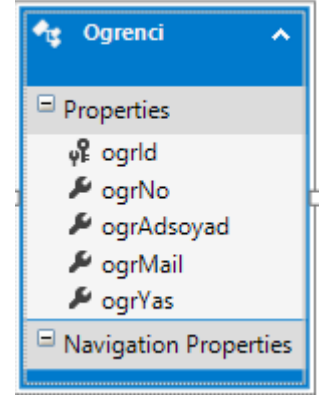
        [HttpGet]
        [Route("api/ogrencibyid/{ogrId}")]
        public OgrenciModel OgrenciById(int ogrId)
        {
            OgrenciModel kayit = db.Ogrenci.Where(s => s.ogrId == ogrId).Select(x => new
            OgrenciModel()
            {
                ogrId = x.ogrId,
                ogrNo = x.ogrNo,
                ogrAdsoyad = x.ogrAdsoyad,
                ogrMail = x.ogrMail,
                ogrYas = x.ogrYas
            }).SingleOrDefault();

            return kayit;
        }

        [HttpPost]
        [Route("api/ogrenciekle")]
        public SonucModel OgrenciEkle(OgrenciModel model)
        {
            if (db.Ogrenci.Count(c => c.ogrNo == model.ogrNo) > 0)
            {
                sonuc.islem = false;
                sonuc.mesaj = "Girilen Öğrenci Numarası Kayıtlıdır!";
                return sonuc;
            }

            Ogrenci yeni = new Ogrenci();

```



```
        yeni.ogrNo = model.ogrNo;
        yeni.ogrAdsoyad = model.ogrAdsoyad;
        yeni.ogrMail = model.ogrMail;
        yeni.ogrYas = model.ogrYas;

        db.Ogrenci.Add(yeni);
        db.SaveChanges();

        sonuc.islem = true;
        sonuc.mesaj = "Öğrenci Eklendi";

        return sonuc;
    }

    [HttpPut]
    [Route("api/ogrenciduzenle")]
    public SonucModel OgrenciDuzenle(OgrenciModel model)
    {
        Ogrenci kayit = db.Ogrenci.Where(s => s.ogrId == model.ogrId).FirstOrDefault
();

        if (kayit == null)
        {
            sonuc.islem = false;
            sonuc.mesaj = "Kayıt Bulunamadı!";
            return sonuc;
        }

        kayit.ogrNo = model.ogrNo;
        kayit.ogrAdsoyad = model.ogrAdsoyad;
        kayit.ogrMail = model.ogrMail;
        kayit.ogrYas = model.ogrYas;

        db.SaveChanges();

        sonuc.islem = true;
        sonuc.mesaj = "Öğrenci Güncellendi";

        return sonuc;
    }

    [HttpDelete]
    [Route("api/ogrencisil/{ogrId}")]
    public SonucModel OgrenciSil(int ogrId)
    {
        Ogrenci kayit = db.Ogrenci.Where(s => s.ogrId == ogrId).FirstOrDefault();

        if (kayit == null)
        {
            sonuc.islem = false;
            sonuc.mesaj = "Kayıt Bulunamadı!";
            return sonuc;
        }

        db.Ogrenci.Remove(kayit);
        db.SaveChanges();

        sonuc.islem = true;
        sonuc.mesaj = "Öğrenci Silindi";
    }
}
```

```
        return sonuc;
    }

    [HttpGet]
    [Route("api/ogrencisayisi")]
    public int OgrenciSayisi()
    {
        return db.Ogrenci.Count();
    }

    [HttpGet]
    [Route("api/ogrnobul/{ogrId}")]
    public string OgrNoBul(int ogrId)
    {
        //1. yol

        Ogrenci kayit = db.Ogrenci.Where(s => s.ogrId == ogrId).FirstOrDefault();
        string ogrNo1 = null;
        if (kayit != null)
        {
            ogrNo1 = kayit.ogrNo;
        }

        //2. yol (tercih edilen)
        string ogrNo2 = db.Ogrenci.Where(s => s.ogrId == ogrId).Select(x => x.ogrNo)
        .FirstOrDefault();

        return ogrNo2;
    }
}
```

UYGULAMA 3

```
using System.Collections.Generic;
using System.Linq;
using System.Web.Http;
using uyg03.Models;
using uyg03.ViewModel;
```

```
namespace uyg03.Controllers
{
```

```
    public class ServisController : ApiController
    {
```

```
        DB02Entities db = new DB02Entities();
        SonucModel sonuc = new SonucModel();
```

```
        #region Kategori
```

```
        [HttpGet]
```

```
        [Route("api/kategoriliste")]
```

```
        public List<KategoriModel> KategoriListe()
        {
```

```
            List<KategoriModel> liste = db.Kategori.Select(x => new KategoriModel()
            {
                katId = x.katId,
                katAdi = x.katAdi,
                katUrunSay = x.Urun.Count()
            }).ToList();
```

```
            return liste;
```

```
        }
```

```
        [HttpGet]
```

```
        [Route("api/kategoriById/{katId}")]
```

```
        public KategoriModel KategoriById(int katId)
        {
```

```
            KategoriModel kayit = db.Kategori.Where(s => s.katId == katId).Select(x => n
ew KategoriModel()
            {
                katId = x.katId,
                katAdi = x.katAdi,
                katUrunSay = x.Urun.Count()
            }).FirstOrDefault();
```

```
            return kayit;
```

```
        }
```

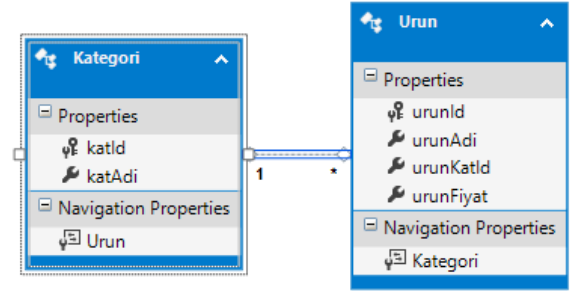
```
        [HttpPost]
```

```
        [Route("api/kategoriekle")]
```

```
        public SonucModel KategoriEkle(KategoriModel model)
        {
```

```
            if (db.Kategori.Count(s => s.katAdi == model.katAdi) > 0)
            {
                sonuc.islem = false;
                sonuc.mesaj = "Girilen Kategori Adı Kayıtlıdır!";
                return sonuc;
            }
```

```
            Kategori yeni = new Kategori();
            yeni.katAdi = model.katAdi;
            db.Kategori.Add(yeni);
```



```
db.SaveChanges();

sonuc.islem = true;
sonuc.mesaj = "Kategori Eklendi";

return sonuc;
}

[HttpPut]
[Route("api/kategoriduzenle")]
public SonucModel KategoriDuzenle(KategoriModel model)
{
    Kategori kayit = db.Kategori.Where(s => s.katId == model.katId).FirstOrDefault();

    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }

    kayit.katAdi = model.katAdi;
    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Kategori Düzenlendi";
    return sonuc;
}

[HttpDelete]
[Route("api/kategorisil/{katId}")]
public SonucModel KategoriSil(int katId)
{
    Kategori kayit = db.Kategori.Where(s => s.katId == katId).FirstOrDefault();
    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }

    if (db.Urun.Count(s => s.urunKatId == katId) > 0)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Üzerinde Ürün Kaydı Olan Kategori Silinemez!";
        return sonuc;
    }

    db.Kategori.Remove(kayit);
    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Kategori Silindi";
    return sonuc;
}
#endregion

#region Urun
```

```
[HttpGet]
[Route("api/urunliste")]
public List<UrunModel> UrunListe()
{
    List<UrunModel> liste = db.Urun.Select(x => new UrunModel()
    {
        urunId = x.urunId,
        urunAdi = x.urunAdi,
        urunKatId = x.urunKatId,
        urunKatAdi = x.Kategori.katAdi,
        urunFiyat = x.urunFiyat

    }).ToList();

    return liste;
}

[HttpGet]
[Route("api/urunlistebykatid/{katId}")]
public List<UrunModel> UrunListeByKatId(int katId)
{
    List<UrunModel> liste = db.Urun.Where(s => s.urunKatId == katId).Select(x =>
new UrunModel()
    {
        urunId = x.urunId,
        urunAdi = x.urunAdi,
        urunKatId = x.urunKatId,
        urunKatAdi = x.Kategori.katAdi,
        urunFiyat = x.urunFiyat

    }).ToList();

    return liste;
}

[HttpGet]
[Route("api/urunbyid/{urunId}")]
public UrunModel UrunById(int urunId)
{
    UrunModel kayit = db.Urun.Where(s => s.urunId == urunId).Select(x => new Uru
nModel()
    {
        urunId = x.urunId,
        urunAdi = x.urunAdi,
        urunKatId = x.urunKatId,
        urunKatAdi = x.Kategori.katAdi,
        urunFiyat = x.urunFiyat

    }).FirstOrDefault();

    return kayit;
}

[HttpPost]
[Route("api/urunekle")]
public SonucModel UrunEkle(UrunModel model)
{

```



```
        if (db.Urun.Count(s => s.urunAdi == model.urunAdi && s.urunKatId == model.urunKatId) > 0)
        {
            sonuc.islem = false;
            sonuc.mesaj = "Girilen Ürün İlgili Kategoride Kayıtlıdır!";
            return sonuc;
        }

        Urun yeni = new Urun();
        yeni.urunAdi = model.urunAdi;
        yeni.urunFiyat = model.urunFiyat;
        yeni.urunKatId = model.urunKatId;

        db.Urun.Add(yeni);
        db.SaveChanges();

        sonuc.islem = true;
        sonuc.mesaj = "Ürün Eklendi";

        return sonuc;
    }

    [HttpPut]
    [Route("api/urunduzenle")]
    public SonucModel UrunDuzenle(UrunModel model)
    {
        Urun kayit = db.Urun.Where(s => s.urunId == model.urunId).FirstOrDefault();

        if (kayit == null)
        {
            sonuc.islem = false;
            sonuc.mesaj = "Kayıt Bulunamadı!";
            return sonuc;
        }
        kayit.urunAdi = model.urunAdi;
        kayit.urunFiyat = model.urunFiyat;
        kayit.urunKatId = model.urunKatId;
        db.SaveChanges();

        sonuc.islem = true;
        sonuc.mesaj = "Ürün Düzenlendi";

        return sonuc;
    }

    [HttpDelete]
    [Route("api/urunsil/{urunId}")]
    public SonucModel UrunSil(int urunId)
    {
        Urun kayit = db.Urun.Where(s => s.urunId == urunId).FirstOrDefault();

        if (kayit == null)
        {
            sonuc.islem = false;
            sonuc.mesaj = "Kayıt Bulunamadı!";
            return sonuc;
        }

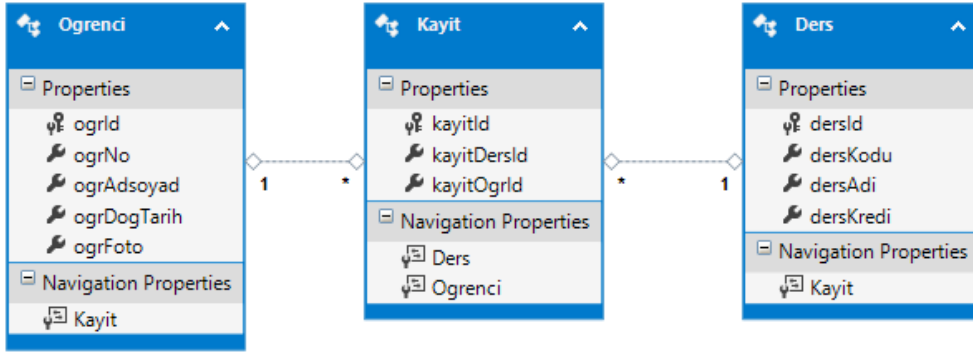
        db.Urun.Remove(kayit);
    }
}
```

```
db.SaveChanges();

sonuc.islem = true;
sonuc.mesaj = "Ürün Silindi";

return sonuc;
}
#endregion
}
```

UYGULAMA 4



```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Web.Http;
using uyg04.Models;
using uyg04.ViewModel;
namespace uyg04.Controllers
{
    public class ServisController : ApiController
    {
        DB03Entities db = new DB03Entities();
        SonucModel sonuc = new SonucModel();

        #region Ders
        [HttpGet]
        [Route("api/dersliste")]
        public List<DersModel> DersListe()
        {
            List<DersModel> liste = db.Ders.Select(x => new DersModel()
            {
                dersId = x.dersId,
                dersKodu = x.dersKodu,
                dersAdi = x.dersAdi,
                dersKredi = x.dersKredi,
                dersOgrSayisi = x.Kayit.Count()
            }).ToList();

            return liste;
        }

        [HttpGet]
        [Route("api/dersbyid/{dersId}")]
        public DersModel DersById(string dersId)
        {
            DersModel kayit = db.Ders.Where(s => s.dersId == dersId).Select(x => new DersModel()
            {
                dersId = x.dersId,
                dersKodu = x.dersKodu,
                dersAdi = x.dersAdi,
                dersKredi = x.dersKredi,
                dersOgrSayisi = x.Kayit.Count()
            }).SingleOrDefault();
        }
    }
}
  
```

```
        return kayit;
    }

    [HttpPost]
    [Route("api/dersekle")]
    public SonucModel DersEkle(DersModel model)
    {
        if (db.Ders.Count(s => s.dersKodu == model.dersKodu) > 0)
        {
            sonuc.islem = false;
            sonuc.mesaj = "Girilen Ders Kodu Kayıtlıdır!";
            return sonuc;
        }

        Ders yeni = new Ders();
        yeni.dersId = Guid.NewGuid().ToString();
        yeni.dersKodu = model.dersKodu;
        yeni.dersAdi = model.dersAdi;
        yeni.dersKredi = model.dersKredi;
        db.Ders.Add(yeni);
        db.SaveChanges();
        sonuc.islem = true;
        sonuc.mesaj = "Ders Eklendi";
        return sonuc;
    }

    [HttpPut]
    [Route("api/dersduzenle")]
    public SonucModel DersDuzenle(DersModel model)
    {
        Ders kayit = db.Ders.Where(s => s.dersId == model.dersId).SingleOrDefault();

        if (kayit == null)
        {
            sonuc.islem = false;
            sonuc.mesaj = "Kayıt Bulunamadı!";
            return sonuc;
        }

        kayit.dersKodu = model.dersKodu;
        kayit.dersAdi = model.dersAdi;
        kayit.dersKredi = model.dersKredi;
        db.SaveChanges();

        sonuc.islem = true;
        sonuc.mesaj = "Ders Düzenlendi";

        return sonuc;
    }

    [HttpDelete]
    [Route("api/derssil/{dersId}")]
    public SonucModel DersSil(string dersId)
    {
        Ders kayit = db.Ders.Where(s => s.dersId == dersId).SingleOrDefault();

        if (kayit == null)
```

```
{
    sonuc.islem = false;
    sonuc.mesaj = "Kayıt Bulunamadı!";
    return sonuc;
}

if (db.Kayit.Count(s => s.kayitDersId == dersId) > 0)
{
    sonuc.islem = false;
    sonuc.mesaj = "Derse Kayıtlı Öğrenci Olduğu İçin Ders Silinemez!";
    return sonuc;
}

db.Ders.Remove(kayit);
db.SaveChanges();
sonuc.islem = true;
sonuc.mesaj = "Ders Silindi";
return sonuc;
}
#endregion

#region Ogrenci

[HttpGet]
[Route("api/ogrenciliste")]
public List<OgrenciModel> OgrenciListe()
{
    List<OgrenciModel> liste = db.Ogrenci.Select(x => new OgrenciModel()
    {
        ogrId = x.ogrId,
        ogrNo = x.ogrNo,
        ogrAdsoyad = x.ogrAdsoyad,
        ogrDogTarih = x.ogrDogTarih,
        ogrFoto = x.ogrFoto,
        ogrDersSayisi = x.Kayit.Count()
    }).ToList();
    return liste;
}

[HttpGet]
[Route("api/ogrenciById/{ogrId}")]
public OgrenciModel OgrenciById(string ogrId)
{
    OgrenciModel kayit = db.Ogrenci.Where(s => s.ogrId == ogrId).Select(x => new
OgrenciModel()
    {
        ogrId = x.ogrId,
        ogrNo = x.ogrNo,
        ogrAdsoyad = x.ogrAdsoyad,
        ogrDogTarih = x.ogrDogTarih,
        ogrFoto = x.ogrFoto,
        ogrDersSayisi = x.Kayit.Count()
    }).SingleOrDefault();
    return kayit;
}

[HttpPost]
```

```
[Route("api/ogrenciEkle")]
public SonucModel OgrenciEkle(OgrenciModel model)
{
    if (db.Ogrenci.Count(s => s.ogrNo == model.ogrNo) > 0)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Girilen Öğrenci Numarası Kayıtlıdır!";
        return sonuc;
    }

    Ogrenci yeni = new Ogrenci();
    yeni.ogrId = Guid.NewGuid().ToString();
    yeni.ogrNo = model.ogrNo;
    yeni.ogrAdsoyad = model.ogrAdsoyad;
    yeni.ogrDogTarih = model.ogrDogTarih;
    yeni.ogrFoto = model.ogrFoto;
    db.Ogrenci.Add(yeni);
    db.SaveChanges();
    sonuc.islem = true;
    sonuc.mesaj = "Öğrenci Eklendi";
    return sonuc;
}
```

```
t());

[HttpPut]
[Route("api/ogrenciduzenle")]
public SonucModel OgrenciDuzenle(OgrenciModel model)
{
    Ogrenci kayit = db.Ogrenci.Where(s => s.ogrId == model.ogrId).SingleOrDefault();

    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }

    kayit.ogrNo = model.ogrNo;
    kayit.ogrAdsoyad = model.ogrAdsoyad;
    kayit.ogrDogTarih = model.ogrDogTarih;
    kayit.ogrFoto = model.ogrFoto;
    db.SaveChanges();
    sonuc.islem = true;
    sonuc.mesaj = "Öğrenci Düzenlendi";
    return sonuc;
}
```

```
[HttpDelete]
[Route("api/ogrencisil/{ogrId}")]
public SonucModel OgrenciSil(string ogrId)
{
    Ogrenci kayit = db.Ogrenci.Where(s => s.ogrId == ogrId).SingleOrDefault();
    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }
}
```

```
if (db.Kayit.Count(s => s.kayitOgrId == ogrId) > 0)
{
    sonuc.islem = false;
    sonuc.mesaj = "Öğrenci Üzerinde Ders Kaydı Olduğu İçin Öğrenci Silinemez
!";

    return sonuc;
}

db.Ogrenci.Remove(kayit);
db.SaveChanges();
sonuc.islem = true;
sonuc.mesaj = "Öğrenci Silindi";
return sonuc;
}

[HttpPost]
[Route("api/ogr-fotoguncelle")]
public SonucModel OgrFotoGuncelle(OgrFotoModel model)
{
    Ogrenci ogr = db.Ogrenci.Where(s => s.ogrId == model.ogrId).SingleOrDefault(
);

    if (ogr == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunmadı!";
        return sonuc;
    }

    if (ogr.ogrFoto != "profil.jpg")
    {
        string yol = System.Web.Hosting.HostingEnvironment.MapPath("~/Dosyalar/"
+ ogr.ogrFoto);
        if (File.Exists(yol))
        {
            File.Delete(yol);
        }
    }

    string data = model.fotoData;
    string base64 = data.Substring(data.IndexOf(',') + 1);
    base64 = base64.Trim('\0');
    byte[] imgbytes = Convert.FromBase64String(base64);
    string dosyaAdi = ogr.ogrId + model.fotoUzanti.Replace("image/", ".");
    using (var ms = new MemoryStream(imgbytes, 0, imgbytes.Length))
    {
        Image img = Image.FromStream(ms, true);
        img.Save(System.Web.Hosting.HostingEnvironment.MapPath("~/Dosyalar/" + d
osyaAdi));
    }
    ogr.ogrFoto = dosyaAdi;
    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Fotoğraf Güncellendi";

    return sonuc;
}
```

```
#endregion

#region Kayit

[HttpGet]
[Route("api/ogrencidersliste/{ogrId}")]
public List<KayitModel> OgrenciDersListe(string ogrId)
{
    List<KayitModel> liste = db.Kayit.Where(s => s.kayitOgrId == ogrId).Select(x
=> new KayitModel()
    {
        kayitId = x.kayitId,
        kayitDersId = x.kayitDersId,
        kayitOgrId = x.kayitOgrId,

    }).ToList();

    foreach (var kayit in liste)
    {
        kayit.ogrBilgi = OgrenciById(kayit.kayitOgrId);
        kayit.dersBilgi = DersById(kayit.kayitDersId);
    }
    return liste;
}

[HttpGet]
[Route("api/dersogrenciliste/{dersId}")]
public List<KayitModel> DersOgrenciListe(string dersId)
{
    List<KayitModel> liste = db.Kayit.Where(s => s.kayitDersId == dersId).Select
(x => new KayitModel()
    {
        kayitId = x.kayitId,
        kayitDersId = x.kayitDersId,
        kayitOgrId = x.kayitOgrId,

    }).ToList();

    foreach (var kayit in liste)
    {
        kayit.ogrBilgi = OgrenciById(kayit.kayitOgrId);
        kayit.dersBilgi = DersById(kayit.kayitDersId);
    }
    return liste;
}

[HttpPost]
[Route("api/kayitekle")]
public SonucModel KayitEkle(KayitModel model)
{
    if (db.Kayit.Count(s => s.kayitDersId == model.kayitDersId && s.kayitOgrId =
= model.kayitOgrId) > 0)
    {
        sonuc.islem = false;
        sonuc.mesaj = "İlgili Öğrenci Derse Önceden Kayıtlıdır!";
        return sonuc;
    }

    Kayit yeni = new Kayit();
```



```
yeni.kayitId = Guid.NewGuid().ToString();
yeni.kayitOgrId = model.kayitOgrId;
yeni.kayitDersId = model.kayitDersId;
db.Kayit.Add(yeni);
db.SaveChanges();
sonuc.islem = true;
sonuc.mesaj = "Ders Kaydı Eklendi";
return sonuc;
}

[HttpDelete]
[Route("api/kayitsil/{kayitId}")]
public SonucModel KayitSil(string kayitId)
{
    Kayit kayit = db.Kayit.Where(s => s.kayitId == kayitId).SingleOrDefault();

    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }
    db.Kayit.Remove(kayit);
    db.SaveChanges();
    sonuc.islem = true;
    sonuc.mesaj = "Ders Kaydı Silindi";
    return sonuc;
}
#endregion

}
}
```

UYGULAMA 5 – BLOG

```
using myoBlog29API.Models;
using myoBlog29API.ViewModel;
using System.Collections.Generic;
using System.Linq;
using System.Web.Http;

namespace myoBlog29API.Controllers
{
    // [Authorize]

    public class ServisController : ApiController
    {

        MyoBlog29DBEntities db = new MyoBlog29DBEntities();
        SonucModel sonuc = new SonucModel();

        #region Kategori

        [HttpGet]
        [Route("api/kategoriliste")]

        public List<KategoriModel> KategoriListe()
        {
            List<KategoriModel> liste = db.Kategori.Select(x => new KategoriModel()
            {
                KategoriId = x.KategoriId,
                KategoriAdi = x.KategoriAdi,
                KatMakaleSay = x.Makale.Count
            }).ToList();

            return liste;
        }

        [HttpGet]
        [Route("api/kategoribyyid/{katId}")]
        public KategoriModel KategoriById(int katId)
        {
            KategoriModel kayit = db.Kategori.Where(s => s.KategoriId == katId).Select(x
=> new KategoriModel()
            {
                KategoriId = x.KategoriId,
                KategoriAdi = x.KategoriAdi,
                KatMakaleSay = x.Makale.Count
            }).SingleOrDefault();
            return kayit;
        }

        [HttpPost]
        [Route("api/kategoriekle")]
        public SonucModel KategoriEkle(KategoriModel model)
        {
            if (db.Kategori.Count(s => s.KategoriAdi == model.KategoriAdi) > 0)
            {
                sonuc.islem = false;
                sonuc.mesaj = "Girilen Kategori Adı Kayıtlıdır!";
                return sonuc;
            }
        }
    }
}
```

The diagram illustrates the database structure for the Blog application. It includes four main entities: Uye (User), Makale (Article), Yorum (Comment), and Kategori (Category). Each entity has a set of properties and navigation properties. Uye is linked to Makale, Makale is linked to Yorum, and Kategori is linked to Makale. The relationships are one-to-many.

```
Kategori yeni = new Kategori();
yeni.KategoriAdi = model.KategoriAdi;
db.Kategori.Add(yeni);
db.SaveChanges();

sonuc.islem = true;
sonuc.mesaj = "Kategori Eklendi";
return sonuc;
}

[HttpPut]
[Route("api/kategoriduzenle")]
public SonucModel KategoriDuzenle(KategoriModel model)
{
    Kategori kayit = db.Kategori.Where(s => s.KategoriId == model.KategoriId).Fi
rstOrDefault();

    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }

    kayit.KategoriAdi = model.KategoriAdi;
    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Kategori Düzenlendi";
    return sonuc;
}

[HttpDelete]
[Route("api/kategorisil/{katId}")]
public SonucModel KategoriSil(int katId)
{
    Kategori kayit = db.Kategori.Where(s => s.KategoriId == katId).FirstOrDefaul
t();

    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }

    if (db.Makale.Count(s => s.KategoriId == katId) > 0)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Üzerinde Makale Kayıtlı Kategori Silinemez!";
        return sonuc;
    }

    db.Kategori.Remove(kayit);
    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Kategori Silindi";
    return sonuc;
}
```

```
#endregion

#region Makale

[HttpGet]
[Route("api/makaleliste")]
public List<MakaleModel> MakaleListe()
{
    List<MakaleModel> liste = db.Makale.Select(x => new MakaleModel()
    {
        MakaleId = x.MakaleId,
        Baslik = x.Baslik,
        Icerik = x.Icerik,
        Foto = x.Foto,
        KategoriId = x.KategoriId,
        KategoriAdi = x.Kategori.KategoriAdi,
        Okunma = x.Okunma,
        Tarih = x.Tarih,
        UyeId = x.UyeId,
        UyeKadi = x.Uye.KullaniciAdi

    }).ToList();

    return liste;
}

[HttpGet]
[Route("api/makalelistesoneklenenler/{s}")]
public List<MakaleModel> MakaleListeSonEklenenler(int s)
{
    List<MakaleModel> liste = db.Makale.OrderByDescending(o => o.MakaleId).Take(
s).Select(x => new MakaleModel()
    {
        MakaleId = x.MakaleId,
        Baslik = x.Baslik,
        Icerik = x.Icerik,
        Foto = x.Foto,
        KategoriId = x.KategoriId,
        KategoriAdi = x.Kategori.KategoriAdi,
        Okunma = x.Okunma,
        Tarih = x.Tarih,
        UyeId = x.UyeId,
        UyeKadi = x.Uye.KullaniciAdi

    }).ToList();

    return liste;
}

[HttpGet]
[Route("api/makalelistebykatid/{katId}")]
public List<MakaleModel> MakaleListeByKatId(int katId)
{
    List<MakaleModel> liste = db.Makale.Where(s => s.KategoriId == katId).Select
(x => new MakaleModel()
    {
        MakaleId = x.MakaleId,
        Baslik = x.Baslik,
        Icerik = x.Icerik,
        Foto = x.Foto,
```

```
KategoriId = x.KategoriId,
KategoriAdi = x.Kategori.KategoriAdi,
Okunma = x.Okunma,
Tarih = x.Tarih,
UyeId = x.UyeId,
UyeKadi = x.Uye.KullaniciAdi

}).ToList();

return liste;
}
[HttpGet]
[Route("api/makalelistebyuyeid/{uyeId}")]
public List<MakaleModel> MakaleListeByUyeId(int uyeId)
{
    List<MakaleModel> liste = db.Makale.Where(s => s.UyeId == uyeId).Select(x =>
new MakaleModel()
    {
        MakaleId = x.MakaleId,
        Baslik = x.Baslik,
        Icerik = x.Icerik,
        Foto = x.Foto,
        KategoriId = x.KategoriId,
        KategoriAdi = x.Kategori.KategoriAdi,
        Okunma = x.Okunma,
        Tarih = x.Tarih,
        UyeId = x.UyeId,
        UyeKadi = x.Uye.KullaniciAdi

    }).ToList();

    return liste;
}

[HttpGet]
[Route("api/makalebyid/{makaleId}")]
public MakaleModel MakaleById(int makaleId)
{
    MakaleModel kayit = db.Makale.Where(s => s.MakaleId == makaleId).Select(x =>
new MakaleModel()
    {
        MakaleId = x.MakaleId,
        Baslik = x.Baslik,
        Icerik = x.Icerik,
        Foto = x.Foto,
        KategoriId = x.KategoriId,
        KategoriAdi = x.Kategori.KategoriAdi,
        Okunma = x.Okunma,
        Tarih = x.Tarih,
        UyeId = x.UyeId,
        UyeKadi = x.Uye.KullaniciAdi
    }).SingleOrDefault();
    return kayit;
}

[HttpPost]
[Route("api/makaleekle")]
public SonucModel MakaleEkle(MakaleModel model)
{

```

```
if (db.Makale.Count(s => s.Baslik == model.Baslik) > 0)
{
    sonuc.islem = false;
    sonuc.mesaj = "Girilen Makale Başlığı Kayıtlıdır!";
    return sonuc;
}

Makale yeni = new Makale();
yeni.Baslik = model.Baslik;
yeni.Icerik = model.Icerik;
yeni.Tarih = model.Tarih;
yeni.Okunma = model.Okunma;
yeni.KategoriId = model.KategoriId;
yeni.UyeId = model.UyeId;
yeni.Foto = model.Foto;
db.Makale.Add(yeni);
db.SaveChanges();

sonuc.islem = true;
sonuc.mesaj = "Makale Eklendi";
return sonuc;
}

[HttpPut]
[Route("api/makaleduzenle")]
public SonucModel MakaleDuzenle(MakaleModel model)
{
    Makale kayit = db.Makale.Where(s => s.MakaleId == model.MakaleId).SingleOrDefault(
fault());
    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }
    kayit.Baslik = model.Baslik;
    kayit.Icerik = model.Icerik;
    kayit.Tarih = model.Tarih;
    kayit.Okunma = model.Okunma;
    kayit.KategoriId = model.KategoriId;
    kayit.UyeId = model.UyeId;
    kayit.Foto = model.Foto;
    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Makale Düzenlendi";
    return sonuc;
}

[HttpDelete]
[Route("api/makalesil/{makaleId}")]
public SonucModel MakaleSil(int makaleId)
{
    Makale kayit = db.Makale.Where(s => s.MakaleId == makaleId).SingleOrDefault(
);
    if (kayit == null)
    {
        sonuc.islem = false;
```

```
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }

    if (db.Yorum.Count(s => s.MakaleId == makaleId) > 0)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Üzerinde Yorum Kayıtlı Makale Silinemez!";
        return sonuc;
    }

    db.Makale.Remove(kayit);
    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Makale Silindi";
    return sonuc;
}
#endregion

#region Üye

[HttpGet]
[Route("api/uyeliste")]
public List<UyeModel> UyeListe()
{
    List<UyeModel> liste = db.Uye.Select(x => new UyeModel()
    {
        UyeId = x.UyeId,
        AdSoyad = x.AdSoyad,
        Email = x.Email,
        KullaniciAdi = x.KullaniciAdi,
        Foto = x.Foto,
        Sifre = x.Sifre,
        UyeAdmin = x.UyeAdmin
    }).ToList();

    return liste;
}

[HttpGet]
[Route("api/uyebyid/{uyeId}")]
public UyeModel UyeById(int uyeId)
{
    UyeModel kayit = db.Uye.Where(s => s.UyeId == uyeId).Select(x => new UyeMode
1()
    {
        UyeId = x.UyeId,
        AdSoyad = x.AdSoyad,
        Email = x.Email,
        KullaniciAdi = x.KullaniciAdi,
        Foto = x.Foto,
        Sifre = x.Sifre,
        UyeAdmin = x.UyeAdmin
    }).SingleOrDefault();

    return kayit;
}
```

```
[HttpPost]
[Route("api/uyeeekle")]
public SonucModel UyeEkle(UyeModel model)
{
    if (db.Uye.Count(s => s.KullaniciAdi == model.KullaniciAdi || s.Email == model.Email) > 0)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Girilen Kullanıcı Adı veya E-Posta Adresi Kayıtlıdır!";
        return sonuc;
    }

    Uye yeni = new Uye();
    yeni.AdSoyad = model.AdSoyad;
    yeni.Email = model.Email;
    yeni.KullaniciAdi = model.KullaniciAdi;
    yeni.Foto = model.Foto;
    yeni.Sifre = model.Sifre;
    yeni.UyeAdmin = model.UyeAdmin;

    db.Uye.Add(yeni);
    db.SaveChanges();
    sonuc.islem = true;
    sonuc.mesaj = "Üye Eklendi";
    return sonuc;
}

[HttpPut]
[Route("api/uyeduzenle")]
public SonucModel UyeDuzenle(UyeModel model)
{
    Uye kayit = db.Uye.Where(s => s.UyeId == model.UyeId).SingleOrDefault();

    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı";
        return sonuc;
    }
    kayit.AdSoyad = model.AdSoyad;
    kayit.Email = model.Email;
    kayit.KullaniciAdi = model.KullaniciAdi;
    kayit.Foto = model.Foto;
    kayit.Sifre = model.Sifre;
    kayit.UyeAdmin = model.UyeAdmin;

    db.SaveChanges();
    sonuc.islem = true;
    sonuc.mesaj = "Üye Düzenlendi";

    return sonuc;
}

[HttpDelete]
[Route("api/uyesil/{uyeId}")]
public SonucModel UyeSil(int uyeId)
{
    Uye kayit = db.Uye.Where(s => s.UyeId == uyeId).SingleOrDefault();
```



```
if (kayit == null)
{
    sonuc.islem = false;
    sonuc.mesaj = "Kayıt Bulunamadı";
    return sonuc;
}

if (db.Makale.Count(s => s.UyeId == uyeId) > 0)
{
    sonuc.islem = false;
    sonuc.mesaj = "Üzerinde Makale Kaydı Olan Üye Silinemez!";
    return sonuc;
}

if (db.Yorum.Count(s => s.UyeId == uyeId) > 0)
{
    sonuc.islem = false;
    sonuc.mesaj = "Üzerinde Yorum Kaydı Olan Üye Silinemez!";
    return sonuc;
}

db.Uye.Remove(kayit);
db.SaveChanges();
sonuc.islem = true;
sonuc.mesaj = "Üye Silindi";
return sonuc;
}
#endregion

#region Yorum

[HttpGet]
[Route("api/yorumliste")]
public List<YorumModel> YorumListe()
{
    List<YorumModel> liste = db.Yorum.Select(x => new YorumModel()
    {
        YorumId = x.YorumId,
        YorumIcerik = x.YorumIcerik,
        MakaleId = x.MakaleId,
        UyeId = x.UyeId,
        Tarih = x.Tarih,
        KullaniciAdi = x.Uye.KullaniciAdi,
        MakaleBaslik = x.Makale.Baslik,

    }).ToList();
    return liste;
}

[HttpGet]
[Route("api/yorumlistebyuyeid/{uyeId}")]
public List<YorumModel> YorumListeByUyeId(int uyeId)
{
    List<YorumModel> liste = db.Yorum.Where(s => s.UyeId == uyeId).Select(x => n
ew YorumModel()
    {
        YorumId = x.YorumId,
        YorumIcerik = x.YorumIcerik,
        MakaleId = x.MakaleId,
        UyeId = x.UyeId,
    })
}
```

```
Tarih = x.Tarih,
KullaniciAdi = x.Uye.KullaniciAdi,
MakaleBaslik = x.Makale.Baslik,

}).ToList();
return liste;
}
[HttpGet]
[Route("api/yorumlistebymakaleid/{makaleId}")]
public List<YorumModel> YorumListeBymakaleId(int makaleId)
{
    List<YorumModel> liste = db.Yorum.Where(s => s.MakaleId == makaleId).Select(
x => new YorumModel()
    {
        YorumId = x.YorumId,
        YorumIcerik = x.YorumIcerik,
        MakaleId = x.MakaleId,
        UyeId = x.UyeId,
        Tarih = x.Tarih,
        KullaniciAdi = x.Uye.KullaniciAdi,
        MakaleBaslik = x.Makale.Baslik,

    }).ToList();
    return liste;
}
[HttpGet]
[Route("api/yorumlistesoneklenenler/{s}")]
public List<YorumModel> YorumListeSonEklenenler(int s)
{
    List<YorumModel> liste = db.Yorum.OrderByDescending(o => o.MakaleId).Take(s)
.Select(x => new YorumModel()
    {
        YorumId = x.YorumId,
        YorumIcerik = x.YorumIcerik,
        MakaleId = x.MakaleId,
        UyeId = x.UyeId,
        Tarih = x.Tarih,
        KullaniciAdi = x.Uye.KullaniciAdi,
        MakaleBaslik = x.Makale.Baslik,

    }).ToList();
    return liste;
}

[HttpGet]
[Route("api/yorumbyid/{yorumId}")]

public YorumModel YorumById(int yorumId)
{
    YorumModel kayit = db.Yorum.Where(s => s.YorumId == yorumId).Select(x => new
YorumModel()
    {
        YorumId = x.YorumId,
        YorumIcerik = x.YorumIcerik,
        MakaleId = x.MakaleId,
        UyeId = x.UyeId,
        Tarih = x.Tarih,
        KullaniciAdi = x.Uye.KullaniciAdi,
```

```
        MakaleBaslik = x.Makale.Baslik,
    }).SingleOrDefault();

    return kayit;
}

[HttpPost]
[Route("api/yorumekle")]
public SonucModel YorumEkle(YorumModel model)
{
    if (db.Yorum.Count(s => s.UyeId == model.UyeId && s.MakaleId == model.Makale
Id && s.YorumIcerik == model.YorumIcerik) > 0)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Aynı Kişi, Aynı Makaleye Aynı Yorumu Yapamaz!";
        return sonuc;
    }

    Yorum yeni = new Yorum();
    yeni.YorumId = model.YorumId;
    yeni.YorumIcerik = model.YorumIcerik;
    yeni.MakaleId = model.MakaleId;
    yeni.UyeId = model.UyeId;
    yeni.Tarih = model.Tarih;
    db.Yorum.Add(yeni);
    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Yorum Eklendi";

    return sonuc;
}

[HttpPut]
[Route("api/yorumduzenle")]
public SonucModel YorumDuzenle(YorumModel model)
{
    Yorum kayit = db.Yorum.Where(s => s.YorumId == model.YorumId).SingleOrDefault();

    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }

    kayit.YorumId = model.YorumId;
    kayit.YorumIcerik = model.YorumIcerik;
    kayit.MakaleId = model.MakaleId;
    kayit.UyeId = model.UyeId;
    kayit.Tarih = model.Tarih;

    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Yorum Düzenlendi";

    return sonuc;
}
```

```
}

[HttpDelete]
[Route("api/yorumsil/{yorumId}")]
public SonucModel YorumSil(int yorumId)
{
    Yorum kayit = db.Yorum.Where(s => s.YorumId == yorumId).SingleOrDefault();

    if (kayit == null)
    {
        sonuc.islem = false;
        sonuc.mesaj = "Kayıt Bulunamadı!";
        return sonuc;
    }

    db.Yorum.Remove(kayit);
    db.SaveChanges();

    sonuc.islem = true;
    sonuc.mesaj = "Yorum Silindi";

    return sonuc;
}

#endregion
}
```